



**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ**



**ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ
СИСТЕМ И СЕТЕЙ**



КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

**СБОРНИК МАТЕРИАЛОВ 59-ОЙ НАУЧНОЙ
КОНФЕРЕНЦИИ АСПИРАНТОВ, МАГИСТРАНТОВ И
СТУДЕНТОВ БГУИР**



МИНСК 2023

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный
университет информатики и радиоэлектроники»
Факультет компьютерных систем и сетей

Материалы 59-й научной конференции аспирантов,
магистрантов и студентов БГУИР

по направлению

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

(Республика Беларусь, г. Минск, 17–21 апреля 2023 г.)

Научное электронное издание

Минск БГУИР 2023

© УО «Белорусский государственный
университет информатики
и радиоэлектроники», 2023

Комиссия конференции:

Нестеренков Сергей Николаевич — декан ФКСиС, кандидат технических наук, доцент, председатель комиссии по организации и проведению конференции «Компьютерные системы и сети»;

Русин Виталий Геннадьевич — заместитель декана ФКСиС, ответственный за подготовку, выпуск программы электронного сборника тезисов докладов конференции «Компьютерные системы и сети»;

Лапицкая Наталья Владимировна — кандидат технических наук, доцент, заведующая кафедрой ПОИТ;

Григорьев Александр Александрович — кандидат физико-математических наук, доцент, заведующий кафедрой физики;

Малыхина Галина Ивановна — кандидат философских наук, доцент, заведующая кафедрой философии;

Волорова Наталья Алексеевна — кандидат технических наук, доцент, заведующая кафедрой информатики;

Никульшин Борис Викторович — кандидат технических наук, доцент, заведующий кафедрой ЭВМ;

Качинский Михаил Вячеславович — кандидат технических наук, доцент кафедры ЭВС;

Баркова Елена Александровна — кандидат физико-математических наук, доцент, заведующая кафедрой ВМ.

В сборнике опубликованы материалы докладов, представленных на 59-й научной конференции аспирантов, магистрантов и студентов БГУИР по направлению «Электронные системы и технологии». Авторами рассматриваются вопросы имитационного моделирования, численных методов, новых информационных технологий в образовании, радиоэлектронике, телекоммуникациях, вычислительных сетях, а также возможности современных систем автоматизированного проектирования.

Материалы одобрены оргкомитетом и публикуются с учетом того, что авторская позиция и стилистические особенности публикаций полностью сохранены при соблюдении требований к оформлению.

Сборник предназначен для широкого круга специалистов в области электронных систем и технологий, а также будет полезен научным и инженерно-техническим работникам, преподавателям, аспирантам, магистрантам и студентам технических

**СЕКЦИЯ 1
«ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»**

АНАЛИЗ ЭФФЕКТИВНОСТИ ПРОГРАММНОГО МОДУЛЯ РАСПОЗНАВАНИЯ ЛИЦ НА БАЗЕ СИАМСКОЙ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ FACENET <i>Климбасов А.А.</i>	16
ДВОИЧНЫЙ СУММАТОР НА ПЛАТФОРМЕ ARDUINO <i>Бондарев К.Д., Зиновенко В.И.</i>	20
КИТАЙСКАЯ КОМНАТА И СИСТЕМЫ ШИФРОВАНИЯ <i>Протьюко М.А.</i>	25
КОМПЕНСАЦИЯ ИСКРИВЛЕНИЯ ПЛАТФОРМЫ С ЦЕЛЬЮ УЛУЧШЕНИЯ КАЧЕСТВА 3D-ПЕЧАТИ НА ANYCUBIC CHIRON АВТОМАТИЧЕСКИМ СПОСОБОМ С ПОМОЩЬЮ ДАТЧИКА VLTOUCH <i>Калютчик А.А.</i>	29
LED-ЧАСЫ С ИМИТАЦИЕЙ РАБОТЫ ИНДИКАТОРНЫХ ЛАМП <i>Бузук А.А.</i>	34
АЛГОРИТМЫ ПОВЕДЕНИЯ ИГРОВЫХ ПЕРСОНАЖЕЙ <i>Богдан Е.В.</i>	35
АППАРАТНОЕ УСКОРЕНИЕ НЕЙРОННЫХ СЕТЕЙ <i>Русакович А. В.</i>	37
РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С КАМЕРОЙ ВИДЕОНАБЛЮДЕНИЯ <i>Ахрамейко П.Д.</i>	38
ИССЛЕДОВАНИЕ ДРЕБЕЗГА КОНТАКТОВ С ЦЕЛЬЮ ГЕНЕРИРОВАНИЯ СЛУЧАЙНЫХ ЧИСЕЛ <i>Можейко Д.О.</i>	41
ПРОГРАММНО-АППАРАТНАЯ СИСТЕМА СТАБИЛИЗАЦИИ ПОЛЕТА БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА <i>Пржелясковская В.Д.</i>	44
КИТАЙСКАЯ КОМНАТА И СИСТЕМЫ ШИФРОВАНИЯ <i>Протьюко М.А.</i>	46
МЕТОД ДЛЯ ПРОВЕРКИ ЗАПРОСОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ К СЕМАНТИЧЕСКОЙ МОДЕЛИ <i>Кушнеревич П.М.</i>	48
МНОГОФАКТОРНЫЙ АНАЛИЗ ДАННЫХ ДЛЯ СИСТЕМЫ РАННЕГО ОБНАРУЖЕНИЯ ЛЕСНЫХ ПОЖАРОВ <i>Дронов Д.Д.</i>	50
МОДЕЛИ УГРОЗ ЛОКАЛЬНОЙ СЕТИ В УЧРЕЖДЕНИИ ОБРАЗОВАНИЯ <i>Силич С.С.</i>	53
ПРАКТИКА ПРИМЕНЕНИЯ MICROSOFT KINECT <i>Мельник Н. С.</i>	54

СИСТЕМА УПРАВЛЕНИЯ МНОГОЗАДАЧНЫМ РОБОТОМ <i>Шишков Ю. Ю.</i>	56
СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ АЛГОРИТМОВ ПОИСКА И СОРТИРОВОК НА МАССИВАХ <i>Малюш Д.О., Борисенко К.Н.</i>	58
УСТРОЙСТВО АВТОМАТИЗИРОВАННОГО РОЗЛИВА НАПИТКОВ <i>Самаль И.Д., Хатеневич А.А.</i>	61
ОЦЕНКА АЛЬТЕРНАТИВ ПО СОВОКУПНОСТИ ПОКАЗАТЕЛЕЙ В УСЛОВИЯХ РЕАЛИЗАЦИИ ПРОЕКТА МЕТОДОМ АНАЛИЗА ИЕРАРХИЙ <i>Русина Н.В., Русин Д.В.</i>	63

СЕКЦИЯ 2
«ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»

МЕТЕОСТАНЦИЯ-ЛОГГЕР НА БАЗЕ МИКРОКНТРОЛЛЕРА <i>Андросов К.С.</i>	68
СОПРОЦЕССОР ДВУМЕРНОЙ СВЕРКИ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ ДЛЯ ПЛИС XILINX ZYNQ <i>Андросов Е.С.</i>	71
РЕАЛИЗАЦИЯ АЛГОРИТМА ID3 НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON <i>Габриельчик П.В., Ермакович В.А.</i>	74
СИСТЕМА ОБРАБОТКИ НАВИГАЦИОННЫХ ДАННЫХ: АППАРАТНАЯ АРХИТЕКТУРА И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ <i>Гомолко В. А.</i>	79
ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ, ПОСТРОЕННЫХ НА БАЗЕ КОМБИНИРОВАННОГО ГЕНЕРАТОРА <i>Кайкы М.Н.</i>	82
РАСПОЗНАВАНИЕ ЭМОЦИЙ С ИСПОЛЬЗОВАНИЕМ КЕПСТРАЛЬНОГО ПРЕДСТАВЛЕНИЯ РЕЧЕВОГО СИГНАЛА И МЕТОДА ОПОРНЫХ ВЕКТОРОВ <i>Краснопрошин Д.В.</i>	90
ИСПОЛЬЗОВАНИЕ РЕЗЕРВУАРНОГО ПОДХОДА НА ОСНОВЕ ОСЦИЛЛЯТОРА ЧЖУА ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ <i>Кресс К.Н.</i>	93
ПОСТРОЕНИЕ ВИДЕОПОТОКА ДЛЯ СИСТЕМЫ РАСПОЗНАВАНИЯ ОБЪЕКТОВ <i>Мармузевич М.А.</i>	97
АНАЛИЗ ПОДХОДОВ ДЛЯ ПОСТРОЕНИЯ СИСТЕМ ОБНАРУЖЕНИЯ И РАСПОЗНАВАНИЯ ЛИЦ <i>Мусский Н.А.</i>	98
УСТРОЙСТВО ДЕТЕКТИРОВАНИЯ И ДЕКОДИРОВАНИЯ DTMF-СИГНАЛОВ <i>Осипов А.С.</i>	100
СИСТЕМА МЕДИЦИНСКОЙ ДИАГНОСТИКИ ГОЛОСОВЫХ СИГНАЛОВ НА ОСНОВЕ МЕТОДОВ ОБРАБОТКИ, СИНХРОНИЗИРОВАННЫХ С ЧАСТОТОЙ ОСНОВНОГО ТОНА <i>Пастернак В. В.</i>	105

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ОБНАРУЖЕНИЯ СЛАБЫХ СИГНАЛОВ КВАДРОКОПТЕРА НА БАЗЕ SDR КАНАЛА ПЕРЕДАЧИ ДАННЫХ НА ФОНЕ НЕПРЕДНАМЕРЕННЫХ ПОМЕХ <i>Стецко В.Ю.</i>	110
РЕАЛИЗАЦИЯ МЕТОДА ГЕНЕРИРОВАНИЯ СИНУСОИДАЛЬНОГО СИГНАЛА В АРИФМЕТИКЕ С ФИКСИРОВАННОЙ ЗАПЯТОЙ <i>Фатеева И.С.</i>	119
BUILDROOT ДЛЯ СИСТЕМ НА СКРИСТАЛЕ СЕМЕЙСТВА ALLWINNER F1CX00S <i>Ценцевицкий Д.А.</i>	122

**СЕКЦИЯ 3
«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

АВТОМАТИЗАЦИЯ РАБОТЫ ДИЗАЙНЕРА ИНТЕРЬЕРОВ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО СРЕДСТВА НА БАЗЕ СТЕКА ТЕХНОЛОГИЙ ASP.NET CORE И ANGULAR <i>Фролов А.О.</i>	128
АЛГОРИТМЫ ГЕНЕРАЦИИ ФРАКТАЛЬНЫХ ЛАНДШАФТОВ <i>Барилко М.А.</i>	130
АППАРАТНО-ПРОГРАММНОЕ СРЕДСТВО ТЕСТИРОВАНИЯ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ <i>Ахмед А.Н.</i>	133
БАЗОВЫЙ МОДУЛЬ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ МИКРООНТРОЛЛЕРА STM32 <i>Авсяник Е.С., Деменковец Д.В.</i>	136
ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ РАСПРЕДЕЛЕНИЯ ЗАДАЧ <i>Альберт К.Ю.</i>	139
ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ КОНТЕНТОМ САЙТА-ПОРТФОЛИО <i>Белоусов А.В.</i>	140
ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ПРОГРАММНОГО КОМПЛЕКСА ОРГАНИЗАЦИИ УДАЛЕННОГО ПРОЦЕССА ОБУЧЕНИЯ <i>Белых Д.В.</i>	141
ТИПОВЫЕ СХЕМЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ УЧРЕЖДЕНИЙ ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ <i>Бобрик А.Ю., Завалюк Д.Н.</i>	144
ПРОБЛЕМЫ ДЕШИФРОВАНИЯ В КРИПТОСИСТЕМЕ РАБИНА <i>Болтак С. В., Деренчук В.И.</i>	147
МЕТОД ПОЛУЧЕНИЯ ИЗОБРАЖЕНИЯ ДВИЖУЩЕГОСЯ ОБЪЕКТА В РЕАЛЬНОМ ВРЕМЕНИ НА ПЛАТФОРМЕ RASPBERRY PI CM 4 <i>Борисевич А.В., Мелешко А.С., Деменковец Д.В.</i>	148
ПРОГРАММНОЕ СРЕДСТВО ВИЗУАЛИЗАЦИИ В РЕАЛЬНОМ ВРЕМЕНИ ФИЗИЧЕСКИ КОРРЕКТНЫХ ТРЕХМЕРНЫХ СЦЕН НА БАЗЕ ГРАФИЧЕСКОГО API DIRECTX <i>Борсуков Е. А.</i>	151

ПРОГРАММНОЕ СРЕДСТВО КОНТРОЛЯ ЛОКАЦИИ И СОСТОЯНИЯ ОБЪЕКТА НА БАЗЕ КОНТРОЛЛЕРА STM32 <i>Быковский В.В.</i>	152
СИСТЕМА РАСПОЗНАВАНИЯ ТЕКСТА НА ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЯХ <i>Воронцова П.В.</i>	153
ВЕБ-ПРИЛОЖЕНИЕ «ОРГАНИЗАЦИЯ ОПРОСОВ» НА БАЗЕ JAVA И SPRING <i>Горегляд В.В.</i>	156
РЕГИСТРАТОР ВИБРАЦИОННЫХ СИГНАЛОВ <i>Григорьев Д.А.</i>	159
КЛАССИФИКАЦИЯ ДАННЫХ ПРЕЦИЗИОННОЙ МОДЕЛЬЮ НЕЙРОННОЙ СЕТИ <i>Деренчук В.И., Болтак С.В., Рябинкин Г.М.</i>	160
ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА ИНДИВИДУАЛЬНЫХ ЗАНЯТИЙ ПО ИНОСТРАННОМУ ЯЗЫКУ <i>Ермакович К. В.</i>	161
РЕАЛИЗАЦИЯ АЛГОРИТМА ОПРЕДЕЛЕНИЯ МЕРЫ РАЗЛИЧИЯ ДВОИЧНЫХ ТЕСТОВЫХ НАБОРОВ <i>Ильина А.Ю.</i>	163
ПРОГРАММНОЕ СРЕДСТВО «МЕНЕДЖЕР РАБОЧЕГО ВРЕМЕНИ» <i>Занько М.С.</i>	169
ЭТАПЫ И МЕТОДЫ РАСПОЗНАВАНИЯ ЖЕСТОВ БЕЛОРУССКОГО ЖЕСТОВОГО ЯЗЫКА <i>Ковалева-Евсюченя А.В.</i>	171
МОБИЛЬНЫЙ ДЕЦЕНТРАЛИЗОВАННЫЙ НЕКАСТОДИАЛЬНЫЙ КРИПТОВАЛЮТНЫЙ КОШЕЛЕК НА ПЛАТФОРМЕ IOS <i>Козко Р.С.</i>	174
ВЕБ-ПРИЛОЖЕНИЕ ОНЛАЙН-КИНОТЕАТР С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА JAVA И ФРЕЙМВОРКА SPRING BOOT <i>Колодко Н.А.</i>	176
ПРОГРАММНОЕ СРЕДСТВО УЧЁТА ПЕРСОНАЛА ОРГАНИЗАЦИИ НА ОСНОВЕ ТЕХНОЛОГИИ VOOCEL И POSTGRESQL <i>Кондрацкий А.А.</i>	177
ПРОГРАММНОЕ СРЕДСТВО РАСПОЗНАВАНИЯ ЯЗЫКА ЖЕСТОВ <i>Кравченко П.Д.</i>	178
ПРОГРАММНОЕ СРЕДСТВО ОТСЛЕЖИВАНИЯ ЦЕЛОСТНОСТИ КОНТЕЙНЕРОВ <i>Логвинов Д.А.</i>	179
МЕТОДЫ И АЛГОРИТМЫ КОНТРОЛЯ СООТВЕТСТВИЯ ВЕРСИЙ И ЦЕЛОСТНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ ПЕРЕМЕЩЕНИЙ <i>Марко А.Ф.</i>	180
ОНЛАЙН-СЕРВИС КОНТРОЛЯ ОСНОВНЫХ ПРОЦЕССОВ УПРАВЛЕНИЯ ПЕРСОНАЛОМ (HRM) ИЗ ЕДИНОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP – LARAVEL	

<i>Михальков М.Д.</i>	184
ПРОГРАММНОЕ СРЕДСТВО СБОРА И ВИЗУАЛИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ	
<i>Насанович Г.Б.</i>	187
СВЯЗНЫЕ НЕИСПРАВНОСТИ ВЗАИМНОГО ВЛИЯНИЯ	
<i>Деменковец Д.В., Петровская В.В.</i>	188
КРИПТОГРАФИЧЕСКОЕ ОБОСНОВАНИЕ СТОЙКОСТИ NFT НА БАЗЕ СМАРТ-КОНТРАКТОВ ETHEREUM	
<i>Плетинский И.В.</i>	190
ПРОЕКТИРОВАНИЕ ПАТТЕРНОВ СБОРКИ МОДЕЛЕЙ ОРИГАМИ И АЛГОРИТМ ОЦЕНКИ ИХ ПЛОСКОСКЛАДЫВАЕМОСТИ	
<i>Путято М.В.</i>	195
ПРОГРАММНОЕ СРЕДСТВО «НАВЫК ДЛЯ ГОЛОСОВОГО ПОМОЩНИКА АЛИСА. ИЗУЧЕНИЕ АНГЛИЙСКОГО ЯЗЫКА» С ИСПОЛЬЗОВАНИЕМ СЕРВИСА ЯНДЕКС. РАЗРАБОТЧИК	
<i>Радько А. А.</i>	198
ВЕБ-ОРИЕНТИРОВАННОЕ ПРОГРАММНОЕ СРЕДСТВО «СИСТЕМА СТРАХОВАНИЯ ЖИЗНИ И ЗДОРОВЬЯ»	
<i>Рафеев П. Ю.</i>	199
ВЕБ-ПРИЛОЖЕНИЕ “ЦИФРОВОЙ УСИЛИТЕЛЬ С ЭФФЕКТАМИ” НА БАЗЕ ЯЗЫКА JAVASCRIPT	
<i>Ролевич С.И.</i>	202
ПРОГРАММНОЕ СРЕДСТВО АНАЛИЗА ТЕКСТОВ ПУБЛИКАЦИЙ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ FLUTTER И JAVA SPRING BOOT	
<i>Синицкая К.Д.</i>	203
ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗИРОВАННОГО СБОРА ИНФОРМАЦИИ HTML-САЙТА ПОСРЕДСТВОМ ТЕХНОЛОГИИ RUBY	
<i>Страчинский Н.С.</i>	204
ПРОГРАММНОЕ СРЕДСТВО МОДЕЛИРОВАНИЯ СВЯЗАННЫХ НЕИСПРАВНОСТЕЙ НА ОСНОВЕ НЕИСПРАВНОСТЕЙ ВЗАИМНОГО ВЛИЯНИЯ	
<i>Тюшев Т.А., Деменковец Д.В.</i>	205
ПРИМЕНЕНИЕ ТЕСТА КАСИСКИ ПРИ ПОТОКОВОМ ШИФРОВАНИИ	
<i>Шлык П. А., Болтак С. В.</i>	210
ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ ВЗАИМОДЕЙСТВИЕМ С КЛИЕНТАМИ МЕДИЦИНСКОГО УЧРЕЖДЕНИЯ	
<i>Щемелев М.Ю.</i>	212
СЕКЦИЯ 4 «ИНФОРМАТИКА»	
НЕЙРОСЕТЕВОЙ ПОДХОД К ПРОГНОЗИРОВАНИЮ СПОРТИВНЫХ ТЕННИСНЫХ ДАННЫХ	
<i>Харкевич А.П.</i>	214

ОЦЕНКА СТОЙКОСТИ ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ <i>Касьян В.А., Новицкий З.Я.</i>	232
ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ В ХИРУРГИЧЕСКОМ ОБРАЗОВАНИИ <i>Канавальчик А.Д., Довнар А.Н., Владимцев В.Д.</i>	238
ИМИТАЦИЯ МОДЕЛИ ЗАРОЖДЕНИЯ ЖИЗНИ <i>Протьюко М.А.</i>	242
МЕДИЦИНСКОЕ ПРИЛОЖЕНИЕ ДЛЯ КЛАССИФИКАЦИИ ПЯТЕН НА КОЖЕ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ <i>Ермолович Д.С., Владимцев В.Д.</i>	246
ОГРАНИЧЕНИЯ РОСТА МОЩНОСТЕЙ НЕЙРОННЫХ СЕТЕЙ: ФИЗИЧЕСКИЕ И ЭНЕРГЕТИЧЕСКИЕ АСПЕКТЫ <i>Касьян В. А., Ахметов Р. Я., Сенько Н. С., Внук О.М., Владимцев В. Д.</i>	304
АНАЛИЗ ЭВОЛЮЦИИ NFT ТЕХНОЛОГИИ И ЕЁ ПЕРСПЕКТИВЫ <i>Азаров Е. А., Горгуун А. В., Шумский Д. С.</i>	311
ВЛИЯНИЕ GPT НА СПОРТИВНОЕ ПРОГРАММИРОВАНИЕ И ОБУЧЕНИЕ <i>Тимошевич К. С., Сорока Д. Ф., Владимцев В. Д.</i>	316
ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ НА ОСНОВЕ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ <i>Тимошевич К. С., Котова К. А., Кваченюк Я. Д., Николайчик А. С.</i>	321
ИСПОЛЬЗОВАНИЕ AI В ИНТЕЛЛЕКТУАЛЬНЫХ ИГРАХ <i>Клейн М.А., Косяков М.М., Владимцев В. Д.</i>	326
СТЕГАНОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ <i>Вилкина К. А., Клебеко Е. Ю., Носкович П. Н.</i>	330
ФУНКЦИИ АКТИВАЦИИ <i>Демещенко М.В., Марковец Р.С., Сугако Т.А., Владимцев В. Д.</i>	333
АРХИТЕКТУРА ИНТЕРАКТИВНОГО ПРИЛОЖЕНИЯ ДЛЯ ОБУЧЕНИЯ БЕЛОРУССКОМУ ЯЗЫКУ В УСЛОВИЯХ ИНФОРМАТИЗАЦИИ ОБЩЕСТВА <i>Макатров А.А., Ермолаев Н.В., Владимцев В. Д.</i>	341
ПРИМЕНЕНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ И MLOPS В СФЕРЕ МЕДИЦИНСКОГО ОБСЛУЖИВАНИЯ <i>Шкарубский М.А., Владимцев В. Д.</i>	352
АНАЛИЗ И ПРИМЕНЕНИЕ МЕТОДОВ ТЕПЛОВИЗИОННОЙ ОБРАБОТКИ ВИДЕОСИГНАЛОВ <i>Вашкевич Н.В., Осадчий О.Э., Владимцев В. Д.</i>	356
РАЗВЕРТЫВАНИЕ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ТЕХНОЛОГИЙ БЕССЕРВЕРНОЙ АРХИТЕКТУРЫ С ПРИМЕНЕНИЕМ ОБЛАЧНОЙ РАЗРАБОТКИ <i>Лаппо К.С., Владимцев В. Д.</i>	362

РАЗВИТИЕ И ОЦЕНКА НЕЙРОСЕТЕВЫХ МЕТОДОВ ВЕРИФИКАЦИИ СОБСТВЕННОРУЧНОЙ ПОДПИСИ <i>Мискевич П.Л., Петровец В.Н., Раловец А.А.</i>	365
ПРИМЕНЕНИЕ МЕТОДОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ И МАШИННОГО ОБУЧЕНИЯ ДЛЯ РАННЕЙ ДИАГНОСТИКИ ОТКЛОНЕНИЙ В РАБОТЕ ДВИГАТЕЛЯ <i>Раловец А.А., Мискевич П.Л., Деменковец Д.В.</i>	370
ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМА ШИФРОВАНИЯ IDEA В СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ <i>Сытько М.В., Жак М.В.</i>	373
МЕТОДЫ ОТОБРАЖЕНИЯ ЧЕТЫРЕХМЕРНОГО ПРОСТРАНСТВА НА ПРИМЕРЕ ВИДЕОИГР <i>Новиков В.А., Давыдовский Д.В.</i>	377
ИЗВЛЕЧЕНИЕ АКУСТИЧЕСКИХ ПРИЗНАКОВ ИЗ АУДИОСИГНАЛА В СИСТЕМАХ РАСПОЗНАВАНИЯ РЕЧИ <i>Крейс А.В.</i>	384
РАВНОВЕСИЕ НЭША <i>Сугако Т.А.</i>	388
СРАВНЕНИЕ МЕТОДА ОПОРНЫХ ВЕКТОРОВ С МЕТОДОМ ГЕНЕРАТИВНО-СОСЯЗАТЕЛЬНЫХ СЕТЕЙ В СТЕГАНОАНАЛИЗЕ <i>Барановский Г.В., Бекарев С.С., Гулис А.А., Шишов Е.П.</i>	391
СХЕМА ШНОРРА В КРИПТОГРАФИИ <i>Колесников П. В., Антихович М. В., Малец В. С.</i>	393
АНАЛИТИЧЕСКИЕ РЕШЕНИЯ ПО УГЛУБЛЕННОМУ АНАЛИЗУ ГРАФОВЫХ БД <i>Зорко П. А., Кулевич А. О.</i>	397
ИНТЕРНЕТ ВЕЩЕЙ. МЕТЕОСТАНЦИЯ НА БАЗЕ ARDUINO <i>Богомаз К.С., Юшкевич М.А.</i>	400
ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПОИСКА ОДИНАКОВЫХ ПОДГРАФОВ В ГРАФЕ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА <i>Бурко Л.А.</i>	401
ИССЛЕДОВАНИЕ ЗАДАЧ ОПТИМИЗАЦИИ РАБОТЫ ГРУЗОВЫХ ТЕРМИНАЛОВ <i>Голубович Ю.И.</i>	404
ПРИМЕНЕНИЕ ФИЗИЧЕСКОЙ КРИПТОГРАФИИ В УСТРОЙСТВАХ НА БАЗЕ FPGA <i>Карбовский Д.В.</i>	407
ИСПОЛЬЗОВАНИЕ BIG DATA В КОМПЬЮТЕРНЫХ ИГРАХ <i>Протасов А.П., Стаселько И.Д., Позняков Т.Д.</i>	410
ИСПОЛЬЗОВАНИЕ АЛГОРИТМА MFCC В ЗАДАЧЕ РАСПОЗНАВАНИЯ МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ <i>Петровец В.Н., Мискевич П.Л., Деменковец Д.В.</i>	412
КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВ ЗАМЕНЫ <i>Бигеза Я. О., Рыбак Д. В.</i>	414

ИССЛЕДОВАНИЕ СВОЙСТВ ДИЗЬЮНКТИВНЫХ ГРАФОВ И ИХ ПРИМЕНЕНИЕ В КАЧЕСТВЕ ИНСТРУМЕНТА ДЛЯ АНАЛИЗА СОЦИАЛЬНЫХ СЕТЕЙ <i>Давыдовский Д. В.</i>	416
---	-----

ПРОГНОЗИРОВАНИЕ ЦЕН АКЦИЙ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ <i>Клебанов Д.А.</i>	420
---	-----

**СЕКЦИЯ 5
«ФИЗИКА»**

ТЕХНОЛОГИЯ RFID. СПОСОБЫ ЗАЩИТЫ ДАННЫХ <i>Васютин А.Д., Бондарь Е.Н.</i>	423
---	-----

ОПТИМИЗАЦИЯ РАБОТЫ УЛЬТРАЗВУКОВОЙ ОЧИСТИТЕЛЬНОЙ ВАННЫ <i>Родевич Р.А., Шебеко В.П.</i>	428
---	-----

НОВЫЕ МАТЕРИАЛЫ В НАНОЭЛЕКТРОНИКЕ <i>Абушкевич А.А., Чапля М.Д., Агель А.А.</i>	431
--	-----

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ЭЛЕКТРОМАГНИТНЫХ ПОЛЕЙ <i>Бушенко В. А., Демидовец В. В.</i>	435
---	-----

ИССЛЕДОВАНИЕ КОСМИЧЕСКИХ ТЕЛ С ПОМОЩЬЮ ГАММА-ИЗЛУЧЕНИЯ <i>Войченко М.М., Василевский В.С.</i>	439
--	-----

ЭЛЕКТРОЛИЗ ВОДЫ ДЛЯ ПОЛУЧЕНИЯ ЭКОЛОГИЧЕСКИ ЧИСТОГО ТОПЛИВА <i>Гембар С.В., Шиян А.Г.</i>	441
---	-----

ОПТИМИЗАЦИЯ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ И АНАЛИЗА ДАННЫХ НА ПРИМЕРЕ ЭФФЕКТА ЗЕЕБЕКА <i>Глинский И.В.</i>	444
---	-----

ИССЛЕДОВАНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ СВЕРХПРОВОДИМОСТИ И ГРАФЕНА С МАГИЧЕСКИМ УГЛОМ НАКЛОНА <i>Лицкевич В.В., Гришель Д.Г., Дроздович В.И.</i>	446
--	-----

ФИЗИЧЕСКИЕ ПРИНЦИПЫ МАГНИТНОЙ ЗАПИСИ И СЧИТЫВАНИЯ ИНФОРМАЦИИ <i>Жоголь А.О.</i>	448
--	-----

СРАВНЕНИЕ ДВУМЕРНЫХ МАТЕРИАЛОВ ДЛЯ ПРИМЕНЕНИЯ В ТРАНЗИСТОРАХ <i>Кудин И. С.Громыко М. Р.</i>	451
---	-----

ФОТОПРИЁМНИК НА ОСНОВЕ ФОТОТРАНЗИСТОРА <i>Казеко Д.А., Лойко А.И.</i>	453
--	-----

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ФИКСАЦИИ ПЕРИОДА ГАРМОНИЧЕСКИХ КОЛЕБАНИЙ ФИЗИЧЕСКОГО МАЯТНИКА <i>Михнюк А.И., Пашковец М.В.</i>	456
---	-----

МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ ДАННЫХ В ФИЗИЧЕСКИХ ИССЛЕДОВАНИЯХ <i>Прокофьев Н.И.</i>	459
---	-----

ТУННЕЛЬНЫЙ ЭФФЕКТ ДЛЯ РАБОТЫ ТВЕРДОТЕЛЬНЫХ НАКОПИТЕЛЕЙ <i>Бахмат А.Д., Столбун Е.А.</i>	461
--	-----

ОПТИМИЗАЦИЯ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ И АНАЛИЗА ДАННЫХ НА ПРИМЕРЕ ЭФФЕКТА ЗЕЕБЕКА <i>Глинский И.В.</i>	464
---	-----

КВАНТОВОЕ ЯВЛЕНИЕ СВЕРХПРОВОДИМОСТИ <i>Панизник А.С., Ивановская Е.О.</i>	466
--	-----

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ИЗМЕРЕНИЯ КОЛЕБАНИЙ БАЛЛИСТИЧЕСКОГО МАЯТНИКА <i>Чубса А.А., Ковшер А.О., Усович И.В.</i>	468
--	-----

**СЕКЦИЯ 6
«ВЫСШАЯ МАТЕМАТИКА»**

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО АНАЛИЗА <i>Дроздов А.И.</i>	473
---	-----

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО АНАЛИЗА <i>Дроздов А.И.</i>	480
---	-----

АЛГОРИТМЫ ШИФРОВАНИЯ И ДЕШИФРОВАНИЯ С ПОМОЩЬЮ МЕТОДОВ ДИСКРЕТНОЙ МАТЕМАТИКИ <i>Новиков В.А., Жак М.В., Вашкевич Е.Г.</i>	487
---	-----

ВВЕДЕНИЕ В ЭЛЛИПТИЧЕСКУЮ КРИПТОГРАФИЮ <i>Патюпин М.С.</i>	494
--	-----

ФОРМАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ ЗАМКНУТЫХ СИСТЕМ <i>Протьюко М.А.</i>	504
---	-----

НЕЙРОСЕТЕВОЙ ПОДХОД К ПРОГНОЗИРОВАНИЮ СПОРТИВНЫХ ТЕННИСНЫХ ДАННЫХ <i>Харкевич А.П.</i>	507
---	-----

ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ НА ОСНОВЕ РАДИАЛЬНО-БАЗИСНЫХ ФУНКЦИЙ <i>Глинский И.В.</i>	525
---	-----

РАЗЛИЧНЫЕ ПОДХОДЫ К РЕШЕНИЮ БАЗЕЛЬСКОЙ ЗАДАЧИ <i>Войченко М.М.</i>	528
---	-----

САМОЕ ИРРАЦИОНАЛЬНОЕ ЧИСЛО <i>Головин Е.С.</i>	530
---	-----

ПРЕДАСТАВЛЕНИЕ БИНАРНОГО ОТНОШЕНИЯ С ПОМОЩЬЮ МАТРИЦЫ <i>Карлюк П.А.</i>	534
--	-----

МЕТОДЫ ПЕРЕМНОЖЕНИЯ МАТРИЦ. СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМА ШТРАССЕНА И ОБЫЧНОГО ПЕРЕМНОЖЕНИЯ МАТРИЦ <i>Кулик М.И.</i>	537
---	-----

БИНАРНАЯ ПРОБЛЕМА ГОЛЬДБАХА <i>Крутько А.А., Трубач К.И.</i>	541
---	-----

РАСПРЕДЕЛЕНИЕ ИГРОКОВ В ТУРНИРНОЙ СИСТЕМЕ <i>Лицкевич В.В.</i>	545
---	-----

ИССЛЕДОВАНИЕ ГИПОТЕЗЫ КОЛЛАТЦА: ОТ ПРОСТЫХ ЧИСЕЛ К СЛОЖНЫМ ПОСЛЕДОВАТЕЛЬНОСТЯМ <i>Убоженко М.А., Копыток М.Д.</i>	547
--	-----

**СЕКЦИЯ 7
«ФИЛОСОФИЯ, ЛОГИКА И ЖИЗНЬ»**

ТРАНСФОРМАЦИЯ СТАНДАРТОВ КРАСОТЫ В СОВРЕМЕННОСТИ <i>Алексеева К.А.</i>	551
НЕОФАШИСТКИЕ ИДЕИ В МОЛОДЕЖНОЙ СРЕДЕ: ОСНОВНЫЕ ФОРМЫ И ПРИЧИНЫ РАСПРОСТРАНЕНИЯ <i>Аксёненко М.А.</i>	555
ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ ФОРМИРОВАНИЯ ИНФОРМАЦИОННОГО ОБЩЕСТВА В РЕСПУБЛИКЕ БЕЛАРУСЬ <i>Бейнар А.В.</i>	559
СОВРЕМЕННЫЕ ГЛОБАЛИЗАЦИОННЫЕ ВЫЗОВЫ И РИСКИ <i>Венгеренко В.В.</i>	562
ФИЛОСОФИЯ И СОВРЕМЕННАЯ НАУЧНАЯ КАРТИНА МИРА <i>Глинистый Р.Р., Матошко А.О.</i>	566
ФЕНОМЕН ТОЛПЫ В КОНТЕКСТЕ ЦИФРОВЫХ КОММУНИКАЦИЙ <i>Зинковская Е. В.</i>	568
ИСТИНА И ПУСТОТА или КОНЦЕПЦИЯ ИСТИНЫ В ФИЛОСОФИИ ПОСТМОДЕРНИЗМА <i>Игнатюк Н.С.</i>	570
АНАЛИЗ НАПРАВЛЕНИЙ ФИЛОСОФИИ ТЕХНИКИ <i>Ильина А.Ю.</i>	572
ЭТИЧЕСКИЕ ОСОБЕННОСТИ ИНТЕРНЕТ-КОММУНИКАЦИИ <i>Карлюк П. А.</i>	574
ФЕНОМЕН ЭВТАНАЗИИ: СВОБОДА ВОЛИ ИЛИ АНТРОПОЛОГИЧЕСКИЙ КРИЗИС СОВРЕМЕННОГО ОБЩЕСТВА? <i>Радюкевич А.И., Кирзнер А.П.</i>	575
ИСКУССТВЕННЫЕ ЯЗЫКИ И СФЕРЫ ИХ ПРИМЕНЕНИЯ <i>Курьян А.И.</i>	578
ЧЕЛОВЕК В СИСТЕМЕ СОЦИАЛЬНЫХ КОММУНИКАЦИЙ. ЛИЧНОСТЬ И ОБЩЕСТВО <i>Нгуен В.Т.А.</i>	580
АРГУМЕНТАЦИЯ И ЕЕ РОЛЬ В НАУКЕ И ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ <i>Никанав М.Ю.</i>	583
РОЛЬ ФИЛОСОФИИ В ФОРМИРОВАНИИ ЛИЧНОСТИ И СОВРЕМЕННОГО НАУЧНОГО МЫШЛЕНИЯ <i>Никанав М.Ю.</i>	584
НАУКА КАК ВАЖНЕЙШАЯ ФОРМА ПОЗНАНИЯ. ФУНКЦИИ НАУКИ В СОВРЕМЕННОМ МИРЕ <i>Никитина В.А.</i>	585

ЦИФРОВОЕ ГИПЕРПОТРЕБЛЕНИЕ И ЕГО ВЛИЯНИЕ НА ПСИХИКУ СОВРЕМЕННОГО ЧЕЛОВЕКА <i>Петров В.В.</i>	587
ПРОБЛЕМЫ СОЦИАЛИЗАЦИИ ЛИЧНОСТИ В ЦИФРОВОМ МИРЕ. ЛИЧНОСТЬ И МАССОВАЯ КУЛЬТУРА. <i>Санец А.В.</i>	591
ПРЕДМЕТ И РОЛЬ ФИЛОСОФИИ В РАБОТАХ В.В. ОРЛОВА <i>Серяков Р.С.</i>	592
ОСОБЕННОСТИ И РАЗНООБРАЗИЕ СУБКУЛЬТУРЫ ГЕЙМЕРОВ В ИНФОРМАЦИОННОМ ОБЩЕСТВЕ <i>Слижёва А. В.</i>	595
БУДДИЗМ В СОВРЕМЕННОМ МИРЕ <i>Сончик Д.А.</i>	597
КИНЕМАТОГРАФ КАК ФАКТОР ПРИОБЩЕНИЯ К ИСТОРИКО-КУЛЬТУРНОМУ НАСЛЕДИЮ БЕЛАРУСИ <i>Трубач К.И.</i>	599
НЕЧЕТКАЯ ЛОГИКА В ИСКУССТВЕННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ <i>Цыркунов А.С.</i>	601
ФИЛОСОФСКИЕ ПРОБЛЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА <i>Чечко В.В.</i>	603

СЕКЦИЯ 1
«ЭЛЕКТРОННЫЕ
ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ»

УДК 004.032.26

АНАЛИЗ ЭФФЕКТИВНОСТИ ПРОГРАММНОГО МОДУЛЯ РАСПОЗНАВАНИЯ ЛИЦ НА БАЗЕ СИАМСКОЙ СВЁРТОЧНОЙ НЕЙРОННОЙ СЕТИ FACENET

Климбасов А.А., студент гр.950501

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Перцев Д.Ю. – канд. техн. наук

Аннотация. В данной работе проведен анализ эффективности программного модуля распознавания лиц на базе сиамской свёрточной нейронной сети FaceNet. Выявлены его достоинства и недостатки.

Ключевые слова. Глубокая нейронная сеть, идентификация лиц, FaceNet, самообучение нейронной сети.

Введение. Систематический рост мощностей вычислительных машин, наблюдаемый в последние десятилетия, открывает новые возможности в области анализа и обработки больших данных. Задачи, требующие автоматизации, обладают все менее структурированными данными, а построение математической модели процессов решения представляется все менее тривиальной задачей. Например, в области распознавания и идентификации лица человека данных факт дал толчок к развитию применения нейронных сетей. Благодаря возможности обучения, нейронные сети способны находить сложные зависимости между входными и выходными данными, выявление которых в виде математического отношения методами ручного анализа представляется, как минимум, сложно выполнимым.

Распознавание лиц – это сложный процесс, плохо поддающийся ручной алгоритмизации при условии достижения высокой точности. Распространенным методом решения данной задачи являются свёрточные нейронные сети глубокого обучения.

В случае стандартной классификации модель обучена определять ряд классов, каждый из которых является лицом, которое нейронная сеть может идентифицировать. При этом в процессе распознавания вычисляется распределение вероятностей по всем классам.

Минусом применения классических свёрточных нейронных сетей является необходимость переобучения первых слоёв сети, отвечающих за классификацию при добавлении нового лица. Переобучение модели требует времени и мощностей. Также повышается сложность реализации доступности сервисов, использующих подобные модели. Требование доступности сервиса ставит задачу одновременного обучения модели и её использования для обработки данных, что повышает сложность. Также такие архитектуры требуют большого количества данных для обучения для обеспечения конкурентного уровня точности.

Вышеописанные проблемы не распространяются на сиамскую архитектуру. Основная идея состоит в использовании двух идентичных моделей с одинаковыми весами, реализующих извлечение векторов признаков лиц из изображений в качестве встраиваемых структур, и функции сравнения векторов для вычисления меры схожести. Выходные вектора признаков встроены моделей поддаются сравнению простыми алгоритмами наподобие евклидова расстояния.

Главное преимущество модели с сиамской архитектурой – их способность к обучению на одиночном изображении (англ. One-Shot learning).

Конвейер предварительной обработки изображения. Для улучшения точности работы сети требуется предварительная обработка изображения. Такие факторы, как освещение, резкость изображения, угол поворота лица сильно влияют на точность предсказания. Как следствие требуется создать конвейер предобработки изображения для минимизации влияния вышеописанных факторов на работу модуля.

Корректировка яркости. Освещение – сильно влияющий на качество распознавания лица фактор. Тени на лице и не выровненная яркость могут вносить существенные неточности как при распознавании, так и в процессе добавления лица в базу данных, что является еще более уязвимой частью модуля идентификации. Для устранения дефектов освещения применяется гамма-коррекция с автоматическим определением гамма-коэффициента следующего вида:

$$\gamma = \log(\text{mid} \times 255) / \log(\text{mean}) \quad (1)$$

Корректировка контраста. Для корректного извлечения признаков лица требуется высокая контрастность изображения. Повышение контраста изображения происходит при помощи метода

адаптивной ограниченной нормализации гистограммы (англ. Contrast Limited Adaptive Histogram Equalization) [1].

Обнаружение лица. Лицо человека в общем случае не единственный объект на фотографии. Кроме того, возможно наличие нескольких лиц, в результате чего требуется проведение части процедуры предобработки и распознавания для каждого лица в отдельности. Обнаружение лица происходит на основе ориентиров лица, полученных при помощи модели `shape_predictor_5_face_landmarks` [2].

Выравнивание лица. Выравнивание лица происходит на основе евклидовых преобразований на основе ориентиров, полученных при помощи вышеупомянутой модели. Основная суть преобразования – изменить угол поворота лица относительно горизонтали.

При объединении блоков предобработки учитывается возможность улучшения показателей последующего блока за счёт предыдущего. Текущая последовательность конвейера: обнаружение лица, экстрагирование лица с достаточной для дальнейших преобразований областью, корректировка яркости, корректировка контраста, выравнивание лица, экстрагирование лица в соответствии с требованиями модели распознавания.

Архитектура сиамской нейронной сети изображена на рисунке 1. Архитектура данной нейронной сети состоит из блока извлечения и блока сравнения признаков. В качестве встраиваемой свёрточной сети для извлечения признаков используется реализация FaceNet [3] со встроенной моделью Inception [4].

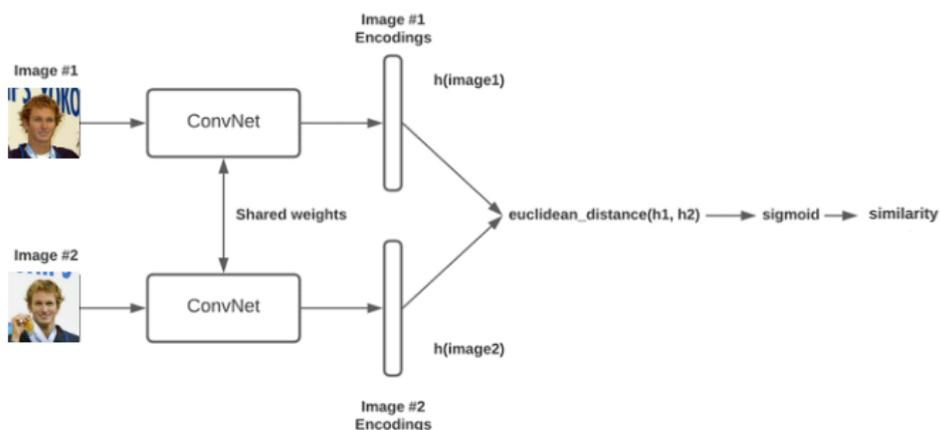


Рисунок 1 – Архитектура нейронной сети

В качестве входных данных в нейронную сеть поступает изображение размером 96 x 96 x 3.

Блок 1: Извлечение признаков (англ. Feature Extractor). В данном блоке к изображению применяется структура нейронной сети FaceNet с использованием вложенной сети Inception. Сеть FaceNet состоит из входного плотно связанного слоя размерностью 96 x 96 x 3. Далее следует блок сети Inception, описанный ниже. Выход Inception подаётся на слой нормализации, возвращающий вектор признаков. В процессе тренировки также следует функция потерь (англ. Triplet Loss) [4].

Архитектура FaceNet приведена на рисунке 2.

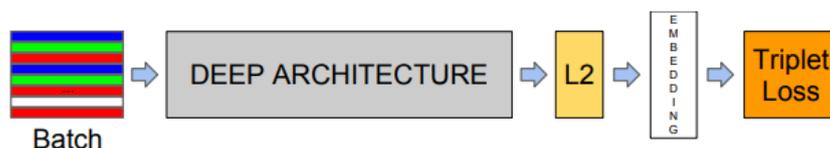


Рисунок 2 – Архитектура встроенной сети FaceNet

Сеть Inception состоит из 9-ти связанных последовательно повторяющихся блоков. Каждый блок представляет собой структуру, состоящую из 4 параллельных последовательностей обработки входных данных. Первая последовательность состоит из операции свёртки 1x1. Вторая – из операции свёртки 1x1 с последующей свёрткой 3x3. Третья – из операции свёртки 1x1 с последующей свёрткой 5x5. Четвёртая – из операции субдискретизации (англ. Max Pooling) 3x3 с последующей свёрткой 1x1. Выходные данные последовательностей конкатенируются в один выходной вектор, являющийся входным по отношению к вышестоящему блоку.

Структура блока изображена на рисунке 3.

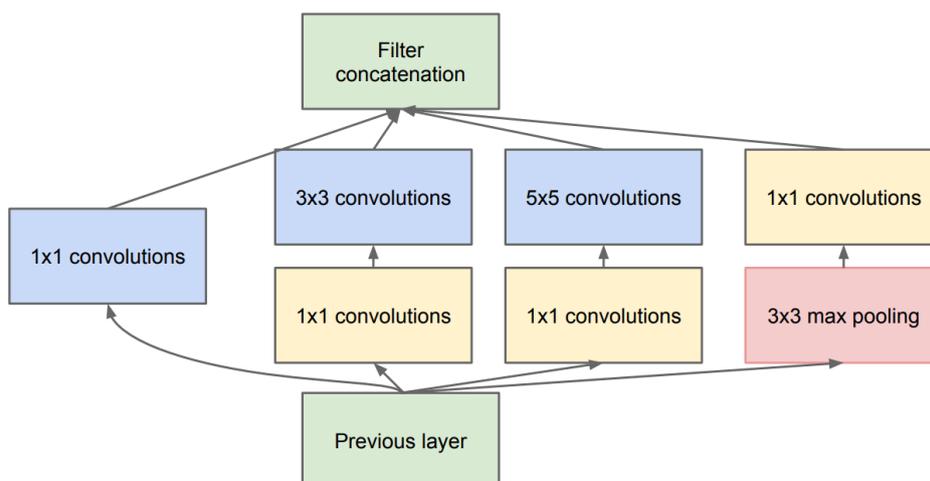


Рисунок 3 – Блок Inception с уменьшением размерности

Выходные данные представляют собой 128 байтовый вектор признаков.

Блок 2: Компаратор признаков. Выполняет вычисление евклидова расстояния между двумя векторами признаков. Представляет собой простую одноименную математическую функцию.

Тестирование нейронной сети.

Тестирование нейронной сети производилось на датасете Labeled Faces in the Wild [5] с использованием метрик точности (англ. precision) и полноты (англ. recall). В качестве порога срабатывания для выходного слоя нейронной сети использовались значения 0.8, 0,7, 0,68, 0,65, 0,5. Значения метрик сведены в таблицу 1.

Таблица 1 – Результаты измерения качества работы нейронной сети на основе выбранных метрик.

Метрика	Порог срабатывания				
	0,8	0,7	0,68	0,65	0,6
Precision	0,772	0,812	0,841	0,906	0,953
Recall	0,813	0,515	0,524	0,468	0,346

Анализ сравниваемых пар изображений, ошибочно распознанных сетью, показал, что сеть плохо справляется с изображениями с повёрнутыми относительно оси фокуса объектива камеры лицами. Предложено решение в виде добавления в конвейер предобработки выравнивания лиц на основе аффинных преобразований.

На основе полученных результатов можно сделать вывод, что нейронная сеть в совокупности с конвейером предобработки изображений с достаточной точностью выполняет распознавание в рамках области идентификации лиц.

Список использованных источников:

1. Contrast-limited adaptive histogram equalization [Electronic Resource] / Cinvestav. – Mode of access: <https://www.tamps.cinvestav.mx/~wgomez/material/AID/CLAHE.pdf>. – Date of access: 08.04.2023.
2. Shape Predictor 5 Face Landmarks model [Electronic Resource] / Dlib. – Mode of access: http://dlib.net/files/shape_predictor_5_face_landmarks.dat.bz2. – Date of access: 08.04.2021.
3. FaceNet: A Unified Embedding for Face Recognition and Clustering [Electronic Resource] / ArXiv. – Mode of access: <https://arxiv.org/pdf/1503.03832.pdf>. – Date of access: 08.04.2023.
4. Going Deeper with Convolutions [Electronic Resource] / THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL. – Mode of access: <https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf>. – Date of access: 08.04.2023.
5. Labeled Faces in the Wild / university of massachusetts. – Mode of access: <http://vis-www.cs.umass.edu/lfw>. – Date of access: 10.04.2023

EFFICIENCY ANALYSIS OF THE SOFTWARE MODULE FOR FACE RECOGNITION BASED ON THE SIAMESE CONVOLUTIONAL NEURAL NETWORK FACENET

Klimbasau A.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Pertsau D. – PhD in Computer Science

Annotation. In the paper, we analyzed the effectiveness of the face recognition software module based on the Siamese convolutional neural network FaceNet. Its advantages and disadvantages are revealed.

Keywords. Deep neural network, face identification, FaceNet, neural network self-learning.

УДК 004.315.2

ДВОИЧНЫЙ СУММАТОР НА ПЛАТФОРМЕ ARDUINO

Бондарев К.Д. и Зиновенко В.И., студенты гр.250702

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сидорович А.С. – старший преподаватель кафедры ЭВМ

Аннотация. В данной работе реализуется сумматор двоичных чисел в дополнительных кодах на платформе Arduino путем проектирования схемы для обработки входных и выходных данных и написания программы для их преобразования в конечное представление.

Ключевые слова. Параллельный двоичный сумматор, двоичная арифметика, логические «ИЛИ», «И», «XOR», обратный и дополнительный код, аппаратная платформа Arduino, симуляция, эмулятор, соединения, резистор, порты, жидкокристаллический дисплей, потенциометр, обработка нажатий кнопок, ввод и вывод.

В этом проекте используется двоичный параллельный сумматор дополнительных кодов, который обычно отображается в двоичном формате с использованием светодиодов, но вместо этого используются Arduino, чтобы принимать и отображать данные на жидкокристаллическом дисплее.

Двоичный сумматор дополнительных кодов – это сумматор, оперирующий представлениями двоичных чисел в дополнительном коде.

В сумматорах дополнительных кодов может производиться как сложение чисел с одинаковыми знаками, так и сложение чисел с разными знаками (вычитание). Сумма дополнительных кодов двоичных чисел представляет собой дополнительный код результата.

Существует несколько причин, почему выбор обратного или модифицированного кода может быть предпочтительнее, чем выбор обратного или модифицированного кода:

1. Простота операций. Операции сложения и вычитания в дополнительном коде требуют всего лишь простого суммирования или вычитания двух чисел в двоичном коде. В обратном коде для выполнения этих операций требуется дополнительный шаг инвертирования битов, что усложняет аппаратную реализацию. В случае модифицированных кодов требуется выполнить дополнительные операции, чтобы определить знак числа, что также усложняет процесс.

2. Уникальность представления. Каждое отрицательное число имеет единственное представление в дополнительном коде, что делает его более предсказуемым и удобным для работы в программном коде.

3. Меньший объем памяти. Представление чисел в дополнительном коде позволяет использовать меньшее количество битов для хранения чисел, по сравнению с обратным кодом и знаковым модулем. Например, для представления числа “-8” в дополнительном коде требуется всего 4 бита, тогда как в обратном коде или знаковом модуле требуется 5 битов.

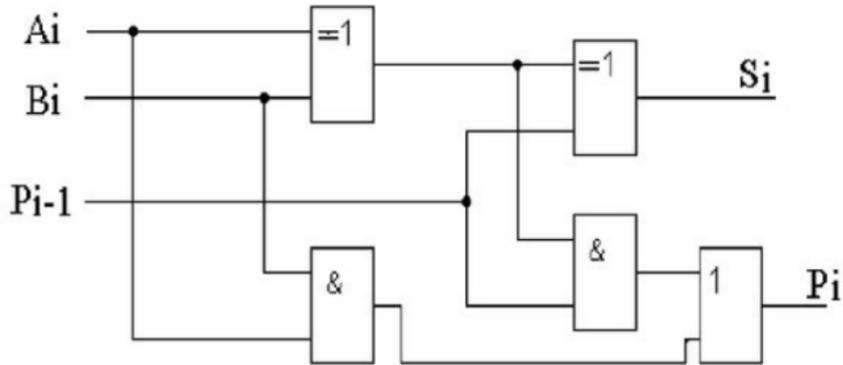
Таблица истинности полного одноразрядного сумматора двоичных кодов, где A_i , B_i – i -е разряды чисел A и B, P_{i-1} – перенос, формируемый в прошлом одноразрядном сумматоре, S_i – i -й разряд результата суммы, из которых формируется число результата, P_i – перенос в следующий одноразрядный сумматор, представлена в таблице 1.

Таблица 1 – Таблица истинности выходов полного одноразрядного сумматора двоичных кодов.

A_i	B_i	P_{i-1}	S_i	P_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1

1	1	1	1	1
---	---	---	---	---

Путем минимизации функции сложения была найдена минимальная дизъюнктивная функция и на ее основе спроектирован одноразрядный сумматор двоичных кодов. Правила сложения в одноразрядном сумматоре двоичных кодов двух чисел A_i и B_i отражены в схеме одноразрядного полного сумматора рисунке 1, выполненного на логических элементах.



$$S_i = A_i \oplus B_i \oplus P_{i-1};$$

$$P_i = A_i B_i + P_{i-1} A_i \oplus B_i.$$

Рисунок 1 – Структурная схема одноразрядного полного сумматора двоичных кодов

При параллельном способе сложения необходимо иметь отдельные одноразрядные сумматоры для каждого разряда чисел. Параллельный сумматор может быть составлен из одноразрядных сумматоров путем соединения выхода, на котором присутствует сигнал переноса данного разряда, со входом для сигнала переноса соседнего, более старшего разряда. За основу была взята данная схема параллельного сумматора двоичных кодов со сквозным переносом, представленная на рисунке 2.

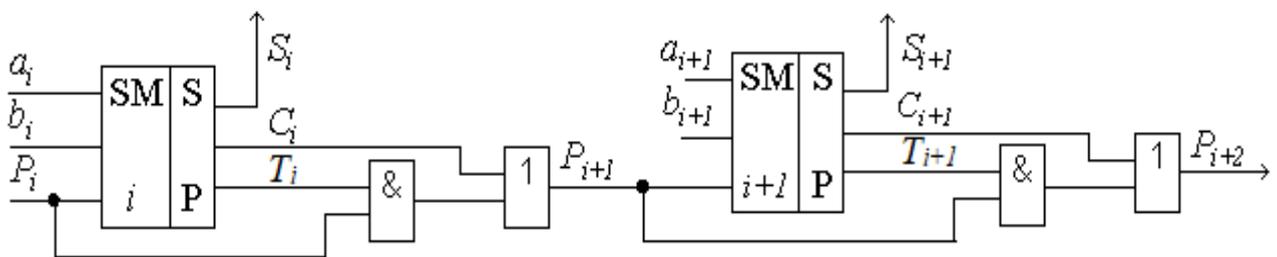


Рисунок 2 - Структурная схема параллельного сумматора со сквозным переносом

Для обеспечения функции вычитания, используется дополнительный код. Правило получения дополнительного кода отрицательных чисел можно сформулировать следующим образом: цифровая часть исходного числа инвертируется и совместно со знаком превращается в обратный код числа, к которому прибавляется код двоичного числа, соответствующий единице, в младшем разряде. Полученная сумма является числом в дополнительном коде к коду модуля этого числа. Дополнительный код положительного числа является его прямым кодом.

В работе используются 7-ми разрядные двоичные числа, самый старший разряд которых является знаковым. «1» или «0» являются «-» или «+» соответственно. Остальные разряды содержат значащую часть числа.

Для реализации аппаратной части Arduino используется:

- Arduino Uno R3
- 2 малые макетные платы
- Резисторы
- LCD 16 x 2 (жидко-кристаллический дисплей)
- 2 кнопки
- Потенциометр

Макет изображен на рисунке 3.

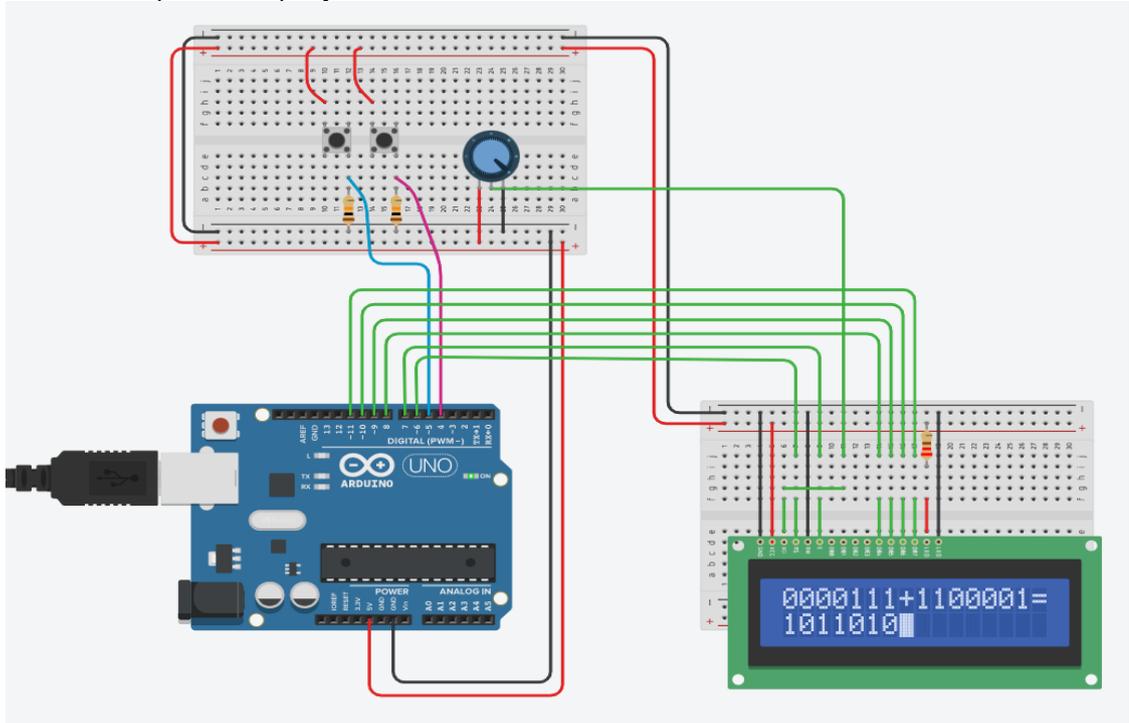


Рисунок 3 - Вид схемы двоичного сумматора на Arduino

Для разработки прототипов электронных устройств обычно используется безопасная макетная плата[1]. На верхней плате присутствуют две кнопки, отвечающие за ввод данных. К кнопкам ведут голубой и розовый провода, отвечающие за ввод «0» и «1» соответственно. Контакты 5V и GND подключены к макетным платам на плюс и минус соответственно, чтобы управлять электричеством на них. Также подключен ЖК-дисплей, яркость которого регулируется потенциометром. Потенциометр подключается к ЖК-дисплею по порту V0. Выбор ЖК-дисплея в качестве способа отображения информации обусловлен тем, что жидкокристаллические индикаторы, отображающие информацию в символах (текст и числа), — самые недорогие, простые и удобные в использовании[2].

Подключения портов представлены в таблице 2.

Таблица 2 – Таблица подключения портов.

Из Arduino uno r3	4	5	6	7	8	9	10	11
К чему	Кнопка "1"	Кнопка "0"	На ЖК-дисплее порт RS	На ЖК-дисплее порт E	На ЖК-дисплее порт DB4	На ЖК-дисплее порт DB5	На ЖК-дисплее порт DB6	На ЖК-дисплее порт DB7

Вся оставшаяся логическая структура сумматоров и обработка данных реализована в программной части Arduino при помощи логических функций XOR, OR, AND. Пример реализации одноразрядного полного сумматора на данных функциях изображен на рисунке 4.

```

110 char HSM(char a1, char a2){
111     char lastPrev = prev;
112     if(OR_1(AND_1(prev, XOR_1(a1, a2)), AND_1(a1, a2))=='1')
113         prev = '1';
114     else
115         prev = '0';
116     if(XOR_1(XOR_1(a1, a2), lastPrev)=='1')
117         return '1';
118     else return '0';
119 }

```

Рисунок 4 – Код одноразрядного полного сумматора двоичных кодов

Код базовых функций изображен на рисунке 5.

```

89 char XOR_1(char a1, char a2) {
90     if(a1 == a2)
91         return '0';
92     else
93         return '1';
94 }
95
96 char AND_1(char a1, char a2) {
97     if (a1 == '1' && a2 == '1')
98         return '1';
99     else
100         return '0';
101 }
102
103 char OR_1(char a1, char a2){
104     if(a1 == '1' || a2 == '1')
105         return '1';
106     else
107         return '0';
108 }

```

Рисунок 5 – Код логических функций XOR, OR, AND

Переменные a_1 и a_2 передают разряды чисел A_i и B_i соответственно, переменная $prev$ – значение переноса P_{i-1} , формируемого в прошлой итерации. При запуске программы значение переноса $prev$ стандартно задано '0'. Функция возвращает разряд S_i результата.

Также в коде реализованы функции для перевода чисел из прямого кода в дополнительный.

Таким образом в сумматор вводятся два семизрядных числа в прямом коде, которые при необходимости переводятся в дополнительный, складываются через цепь одноразрядных полных сумматоров, при необходимости результат переводится из дополнительного в прямой код и выводится на экран.

Иногда, при сложении чисел возникает переполнение разрядной сетки. Об этом свидетельствует перенос из знакового разряда при сложении отрицательных чисел или несовпадение знакового разряда результата со знаковыми разрядами складываемых положительных чисел. Такой случай обозначается на ЖК-экране единицей в правом нижнем углу.

В результате был получен макет семизрядного двоичного сумматора на Arduino. Работа помогает понять, каким образом проводятся арифметические операции в ЭВМ.

В дальнейшем можно сделать расширение до стандартной размерности чисел в ЭВМ. Это можно сделать на базе микрокомпьютера Raspberry Pi 3 Model B или STM32 Nucleo F401RE. В первом случае можно подключить многопоточность для сложения одновременно нескольких чисел. Во втором случае из-за большого количества портов и большей мощности удобно изменить способ отображения данных, что лучше для больших чисел. Однако для приведенной реализации достаточно и обычной платы на базе микроконтроллера Arduino Uno r3. Это обусловлено

следующими факторами: достаточные показатели производительности, низкая цена, поддержка онлайн эмуляторов, активное пользовательское сообщество, множество документации и информационных ресурсов в открытом доступе, а также возможность разработки на си-подобном языке программирования.

Список использованных источников:

1. *Arduino для изобретателей. Обучение электронике на 10 занимательных проектах: Пер. с англ. / Б. Хуанг, Д. Ранберг.* — СПб.: БХВ-Петербург, 2019. — 288 с.: ил.
2. Бокселл Джон. *Изучаем Arduino. 65 проектов своими руками.* 2-е изд. — СПб.: Питер, 2022. — 448 с.: ил.
3. Хан Т., Стивенсон Р. *Arduino: Подробное руководство.* - М.: ДМК Пресс, 2015. - 672 с.
4. Ашкрофт Д. *Arduino для начинающих.* - М.: ДМК Пресс, 2015. - 320 с.
5. *Арифметические и логические основы вычислительной техники :учеб. пособие / Ю. А. Луцки, И. В. Лукьянова.* — Минск : БГУИР, 2014. —174 с. : ил.
6. *Сервис для построения электронных схем [Электронный ресурс].* — Режим доступа: <https://www.tinkercad.com/circuits>

UDC 004.315.2

BINARY SUMMER ON THE ARDUINO PLATFORM

Bondarev K.D., Zinovenko V.I.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Sidorovich A.S. – senior lecturer of the computer department

Annotation. In this paper, the binary number adder is implemented in additional codes on the Arduino platform by designing a circuit for processing input and output data and writing a program to convert them into a final representation.

Keywords. Parallel Binary Adder, Binary Arithmetic, Logic OR, AND, XOR, Inverse and Complementary Code, Arduino Hardware Platform, Simulation, Emulator, Connections, Resistor, Pins, LCD Display, Potentiometer, Button Press Handling, input and output.

КИТАЙСКАЯ КОМНАТА И СИСТЕМЫ ШИФРОВАНИЯ

Протьюко М.А., студент гр.050502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Борисенко О.Ф. – канд. физ.-мат. наук

Аннотация. Рассматривается мысленный эксперимент и его влияние на производительность систем машинного обучения в области шифрования.

Ключевые слова. Генетический алгоритм, нейронные сети, эволюция, «китайская комната».

Введение

Рассмотрим мысленный эксперимент, описанный в [1]. Суть его проста. Вы – человек, не знающий китайского языка и находящийся в замкнутом пространстве. У вас есть инструкция, которую вы неукоснительно выполняете, и заключается она в следующем: когда вы видите некий символ, вы находите этот же символ, и инструкцию, позволяющую вам ответить на этот символ. Ваш собеседник за дверью комнаты наверняка может подумать, что вы знаете китайский язык. На деле вы лишь следуете предписанной инструкции.

Данный мыслительный эксперимент приводит к очень четкому выводу, о том, что же такое понимание. «свойство программ — их чисто формальный или синтаксический характер — фатально для взгляда, согласно которому ментальные и программные процессы тождественны друг другу. Объяснить это можно очень просто. Сознание есть нечто большее, чем формальные или синтаксические процессы. Наши внутренние ментальные состояния, по определению, обладают содержанием.» (глава 2 из [1]). Т.е., мы не можем сказать «машина понимает», пока единицы, с которыми она работает, не обладают неким семантическим смыслом.

То, что все операции определены лишь синтаксически и максимально абстрактно позволяет описать многие линейные задачи и выполнить их вне зависимости от самой программы, в которой они находятся, а также аппаратуры, в которой они выполняются.

Но что необходимо обеспечить в ситуации, когда нужно обращать внимание на семантику?

Рассмотрим следующую задачу [2]: в нашей системе есть изначальный алфавит. Каждому участнику системы разрешается создавать свою версию алфавита, посредством подстановок и перестановок символов (операции на множестве вычетов). Каждому участнику выдаются фрагменты изначального текста. Их задача сделать так, чтобы при обмене зашифрованным текстом соперник не смог его расшифровать. До участия в игре ни один из участников не знает ни одного шифра. Он знает лишь, что его соперник говорит с ним на одном языке.

Человек справится с этой задачей, но возможно ли на формальной логике не допускающей семантику решить данную задачу?

Какие решения мы можем использовать и как поймем, что решение найдено?

Целевая функция и криптостойкость

Для решения поставленной задачи воспользуемся генетическими алгоритмами (далее – г.а.).

В целях упрощения, рассмотрим системы симметричного шифрования.

Согласно [3], г.а. используются для решения задачи оптимизации. Т.е. поисков экстремумов некой функции. Опишем эту функцию для поставленной задачи по аналогии с [4].

$$f(a, b) = \bigcup_{j,i}^{c,k} (a_i * b_j) \bmod n, \quad i = \overline{1, k}, j = \overline{1, c}, \quad (1)$$

где $f(a, b)$ – некая функция симметричного шифрования (биекция), a и b – множество символов текста и ключа соответственно, k – количество символов текста, c – количество символов ключа, n – мощность алфавита символов, $*$ – любая обратимая операция на множестве вычетов n .

Видимых ограничений для использования этой функции в данной задаче нет.

Содержательное значение поиска экстремума данной функции таково: поскольку мы не ограничиваем выбор $*$ и порядка i и j , согласно свойству шифра, называемого криптостойкостью, нам необходимо найти множество таких значений $f(a, b)$, что не должно существовать полиномиального алгоритма, который, имея половину строки s (где s – строка объединений символов из функции $f(a, b)$) последовательности, сможет предсказать $k + 1$ бит с вероятностью большей 50%.

Т.е. в ситуации, когда мы несколько раз применим (1), мы должны получать дискретное равномерное распределение для каждого элемента из n , даже если этот элемент не встречается в a и b .

Определим на основании вышеописанных требований г.а.

«Хромосомой» в г.а. выступает алгоритм по шагам, где слот представлен неким действием (взять элемент из a или b , совершить * и т.д.).

Способы выбора генотипов для скрещивания и видоизменения в данной работе рассмотрены не будут.

Дадим определение фитнес-функции согласно [3].

$$\mu(s) = \frac{1}{n} \sum_{i=1}^n p(u_i) \quad (2)$$

где s – отбираемый генотип («хромосома» г.а.), $p(u_i)$ – вероятность встречи u_i символа, u_i – символ, полученный с помощью $f(a,b)$, где n – мощность множества a .

Расчет данной функции можно в дальнейшем упрощать с помощью свойств дискретного равномерного распределения.

В данном случае, чем ближе результат $\mu(s)$ к $1/n$, тем лучше. Т.е., данный генотип будет отобран в дальнейшие популяции, продолжая свое развитие.

$$\lim_{n \rightarrow \infty} \mu(s) = \frac{1}{n} \quad (3)$$

Т.е., выборка генотипов для следующей популяции будет основываться на близости к пределу $\mu(s)$.

Одной из возможных реализаций (2) в г.а. будет составление множеств s_1 и s_2 , полученных из $f(a,b)$ и $f(c,b)$ с мощностью множеств a и c равной n , а затем проверка (4):

$$(s_1 \cap s_2) \xrightarrow{n \rightarrow \infty} \emptyset \quad (4)$$

Выводы – можем ли мы решить поставленную задачу?

Да. Но в худшем случае нам придется рассмотреть все возможные комбинации из s (число размещений из w всевозможных слотов предусмотренных в г.а.).

Согласно [3], особенности г.а., которые не позволяют получить решение поставленной задачи при описанных условиях таковы:

- г.а. оперирует закодированным множеством параметров, а не с самими параметрами
 - в г.а. применяется вероятностное правило перехода, а не детерминистическое
- Из чего следует, что, используя г.а, мы получаем NP-задачу (перебор всех вариантов).

Полученный г.а. – самоадаптирующийся (согласно определению из [3]). Откуда следует, что он зависит от случайного выбора. Если рассмотреть все возможные решения s на отображении $\mu(s)$, может получиться ситуация с очень большим разбросом локальных экстремумов. Что значительно влияет на скорость поиска решения.

Также данная реализация никак не учитывает выбор параметров i и j из (1), когда эти индексы должны описываться также некими обратимыми функциями.

В данной реализации совершенно не учитывается семантический смысл выполняемого алгоритма. Она строго формализована, но практическое применение данному алгоритму получить так и не удалось [2].

Если возвращаться к аналогии с «китайской» комнатой, мы получили эксперимент, где человеку не была выдана инструкция по выбору подходящих иероглифов.

«Эволюция» и нейронные сети

Попытаемся учесть вышеописанные пункты используя нейронные сети.

Учитывая изначальные условия задачи, нам необходима самоорганизующаяся сеть – «Без учителя». В этом случае нейронная сеть формирует выходное пространство решений только на основе входных воздействий.

Об устройстве нейронной сети (далее н.с.) и её свойствах подробнее в [5].

Для использования н.с. следует ответить на следующие вопросы:

- реализуемый «алгоритм» (см. далее)
- максимальное число слоев
- Функция активации

На данные вопросы может ответить практическая реализация с сравнением получаемых характеристик и параметров.

Объединим г.а. и н.с. следующим образом:

Целевая функция останется как в (2), но генотип будет описывать не алгоритм по шагам, а нейронную сеть. Каждая единичная операция в данном случае будет представлять характеристику веса, или количество промежуточных слоев, или связь между конкретными нейронами.

Разобьем «хромосому» г.а. на три элемента: ген, кодирующий $f(a)$ – обратимую функцию, выдающую псевдослучайную последовательность i и j для (1) и ген, кодирующий операцию $*$, а также ген, который объединяет параметры в (1).

В данном случае, разбиение «хромосомы» на гены описывает «алгоритм», реализуемый нейронной сетью.

В данном случае нам уже необходимо учитывать три подобные друг другу независимые целевые функции для каждого гена.

Можем ли мы сказать, что, реализовав такую систему, мы получим решение данной задачи? Да. Будет ли оно оптимальным для заданных условий, к тому же, реализуемым (под реализацией имеется в виду количество поколений с учетом вероятности получения сходящихся последовательностей г.а. в случайной точке, иначе – сколько вычислений необходимо произвести, чтобы получить результат, и в скольких итерациях получить его невозможно)? Нет, поскольку использование параметра криптостойкости без учета семантики не гарантирует создание шифра (изначально криптостойкость – параметр, учитывающий использование прочих алгоритмов, помимо простого перебора, а также возможностей частотного анализа).

Определим необходимую семантику [1] через использование принципа «естественного отбора».

Под семантикой в данном случае будем понимать наличие некоего содержания.

Рассмотрим некое пространство, в котором будут существовать фенотипы популяции описанного г.а. У каждого фенотипа, далее называемого особью, будет определено время жизни. Особь будет продолжать существование, пока она способна выдать результат (1). Конкуренция между особями будем заключаться в поочередном применении параметра (2) к особи. У каждой особи есть возможность вычислить как (1), так и (2). Во время итерации на этапе отбора (см. [3]), между двумя особями будет проводиться соревнование на основе (2). Причем способ расчета вероятности оставляется на формируемую генотипом нейронную сеть.

Выводы:

Мы не можем в данном случае проверить избыточность данного разбиения, как и его оптимальность. Не можем также определить работоспособность топологии. Возможно ли оптимизировать этот процесс?

В данном случае, у нас имеется ситуация, когда была найдена лишь корреляция, а не причинно-следственные связи. Проблема китайской комнаты все также существенна для этого решения. Можно привести пример с дрессировкой собаки, которая включает свет, нажимая на кнопку. Можно ли сказать, что собака понимает предназначение кнопки? Проверить это очень просто – если избавиться от лампочки в данной цепочке, собака все так же будет жать на кнопку. Понятия «кнопка – свет- награда» и «кнопка - награда» для нее равноценны. Точнее, для собаки это выглядит так «действие – награда». Пока она не поймет сематический смысл «действия» («найти кнопку – включить свет – увидеть награду») она не научится включать и выключать свет. (подробнее об этом вопросе в [6])

Аналогично, пока в описанной системе не получится избавиться от формальных целевых функций, мы будем получать все ту же «китайскую комнату».

Но возможно ли создать такую вычисляемую систему?

Заключение

Рассмотренные в данной работе решения поставленной задачи позволяют в полной мере определить необходимость в наполненности семантическим смыслом. Используя генетические алгоритмы и нейронные сети возможно добиться весьма впечатляющих результатов, но только если они уже были повторены человеком. Т.е., данная система не способна выйти за рамки изначально предположенного решения задачи. По-другому ее просто невозможно сформулировать. Если использовать генетические алгоритмы, мы могли бы чисто случайным образом наткнуться на способ, о котором не подозревали. Но сможем ли мы распознать это решение, если его не понимаем?

Возможно, ответить на данный вопрос позволит идея клеточных автоматов. С их помощью, при определении нескольких правил возможно симулировать даже работу составляющих компьютера [7]. Уже существуют реализации, использующие нейронные сети и клеточные автоматы [8].

Если определить правила клеточных автоматов, как аксиомы некоего пространства, нейронные сети – как функции, а генетический алгоритм – как функцию отбора желаемых параметров, то возможно получить систему, выходящую за рамки предположенных решений. Т.е., если правила системы будут составляться на основании простейших шифров подстановки и перестановки, но с учетом тех свойств, которые присущи всем симметричным шифрам, используя достаточное количество усложнений, можно прийти к фундаментальным принципам, заложенных в них, таким образом создавая (или же узнавая) новые шифры или решения, о которых изначально не было известно.

Список использованных источников:

1. Searle J. *Minds, Brains and Science*. / Перевод на русский язык: А. Ф. Грязнов. — М., 1993. Глава 2. «Могут ли компьютеры мыслить?».
2. Протьюко, М.А., Борисенко, О.Ф. *Простейшие шифры и генетический алгоритм / репозиторий БГУИР*. 2023. - 24 с.
3. Панченко, Т.В. *Генетические алгоритмы: учеб.пособие / под ред. Ю. Ю. Тарасевича*. — Астрахань : Издательский дом «Астраханский университет», 2007. — 87 с
4. Уральский Н.Б., Сизов В.А., Капустин Н.К. *Оптимизация вычислительного процесса фитнес функции генетического алгоритма в распределённых системах обработки данных // Интернет-журнал «Науковедение» ISSN 2223-5167; Том 7, N 6 (2015)*.
5. Головкин В.А., Краснопрошин В.В. *Нейросетевые технологии обработки данных: учеб.пособие – Минск : БГУ, 2017.* – 263 с.
6. Протьюко М.А. *Формализация и исследование операций замкнутых систем // материалы 59 конференции студентов, магистрантов и аспирантов БГУИР, 2023.*
7. Nicolas Loizeau *Building a computer in Conway's game of life – Источник : <https://www.nicolasloizeau.com/gol-computer> Дата доступа: 7.04.2023.*
8. Gabriel Morariu, Hugo Lamarche, Elizabeth Pyvovarov and Karl-Philippe Bluteau *Creation of alternate Game of life realities and rules using Neural Networks - Источник : <https://medium.com/@flygongaby/creation-of-alternate-game-of-life-realities-and-rules-using-neural-networks-4db169f30adc>. Дата доступа: 7.04.2023.*

UDC 004.83

CHINESE ROOM AND ENCRYPTION SYSTEMS

Protsko M.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Borisenko O.F. – PhD in Physics and Mathematics

Annotation. Contains influence of one thought experiments in relation of machine learning systems in the field of encryption.

Keywords. Genetic algorithm, neural networks, evolution, "chinese room".

КОМПЕНСАЦИЯ ИСКРИВЛЕНИЯ ПЛАТФОРМЫ С ЦЕЛЬЮ УЛУЧШЕНИЯ КАЧЕСТВА 3D-ПЕЧАТИ НА ANYCUBIC CHIRON АВТОМАТИЧЕСКИМ СПОСОБОМ С ПОМОЩЬЮ ДАТЧИКА BLTOUCH

Калютчик А.А., студент гр.950503

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Перцев Д.Ю. – канд. техн. наук

Аннотация. В статье исследуется проблема искривления печатной поверхности принтера Anycubic Chiron и ее негативное влияние на качество 3D-печати. Предлагается установка датчика BLTouch и периодическая автоматическая регулировка уровня платформы, что позволяет компенсировать искривление платформы в процессе печати.

Ключевые слова. 3D-печать, качество печати, Anycubic Chiron, ABL, датчик BLTouch, искривление платформы для печати.

Студенческий инженерно-образовательный центр кафедры ЭВМ [1] оснащен тремя FDM/FFF 3D-принтерами: Flashforge Dreamer, Picaso Pro 250, Anycubic Chiron. Из всех перечисленных Anycubic Chiron (рисунок 1) обладает наибольшей областью печати (область построения) – 400x400x450 мм [2] – и предлагает разработанную компанией Anycubic платформу для печати: Ultrabase стекло с покрытием, обеспечивающим лучшую адгезию, совмещенное с алюминиевой нагревательной платформой [3].

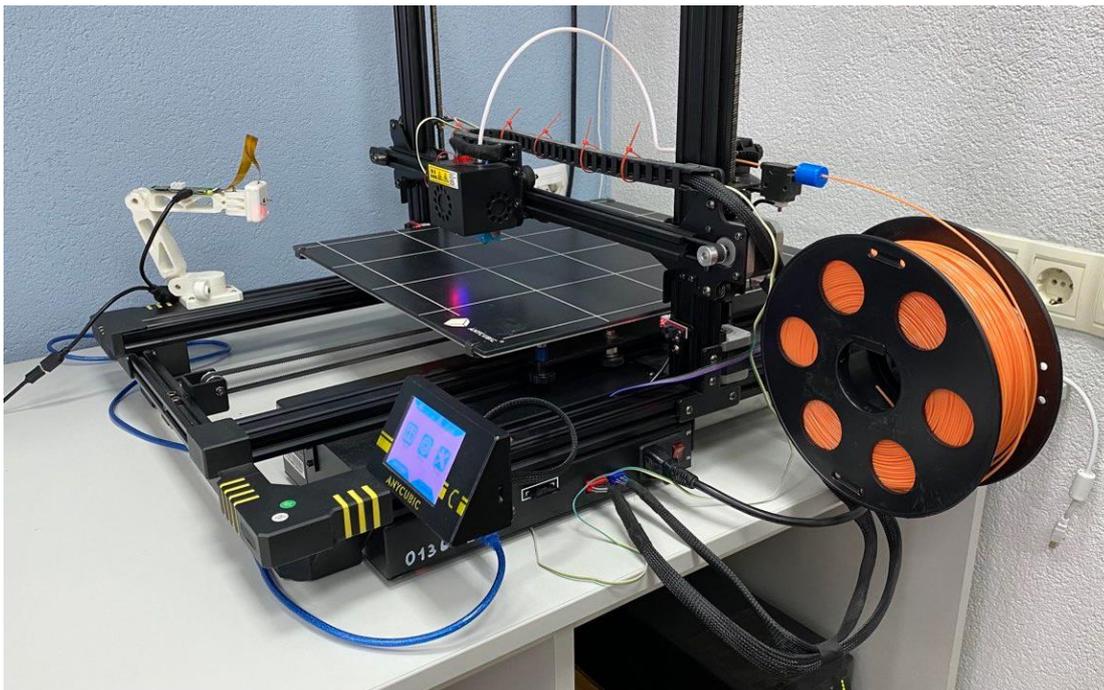


Рисунок 1 – FDM принтер Anycubic Chiron

При эксплуатации Anycubic Chiron было замечено ухудшение качества печати моделей с большой площадью основания, а также моделей, размещенных ближе к центру области печати. Первоначальное предположение о причинах ухудшения качества печати заключалось в неоднородности величин зазора между соплом и поверхностью печати, что является рядовой проблемой и решается периодической регулировкой (калибровкой) уровня платформы печати по факту ухудшения качества печати. В случае Anycubic Chiron платформа закрепляется в четырех точках с помощью подпружиненных винтов-барашков, т.е. является регулируемой [4]. Согласно руководству пользователя [5] регулировку уровня платформы печати можно осуществить как вручную (с помощью четырех винтов-барашков), так и полуавтоматически (Assisted Auto Leveling) [6]. С помощью ручной регулировки по 5 точкам (в местах крепления платформы печати, а также по

центру) улучшения качества печати достигнуть не удалось. При ручной регулировке визуально было обнаружено искривление непосредственно платформы печати.

К искривлению платформы привела следующая совокупность факторов:

- небольшая толщина платформы (4 мм);
- большой вес ввиду увеличенной, по сравнению с другими принтерами Anycubic, площади платформы;
- неизменившееся, по сравнению с другими принтерами Anycubic, количество точек опоры;
- штатные перепады температур при эксплуатации.

Заводская прошивка [7] также предлагает полуавтоматическую регулировку по 25 точкам с использованием съемного модуля автовыравнивания (Auto Leveling Module) [8], входящего в комплект поставки. 3D-печать с установленным модулем невозможна, так как модуль крепится к печатающей головке, а непосредственно датчик располагается между соплом и платформой. Следует отметить, что датчик и сопло расположены на разных уровнях. После автоматических замеров, необходимо снять модуль и вручную завершить регулировку, т.к. согласно руководству, сопло по умолчанию смещено вверх на несколько миллиметров, чтобы избежать контакта с печатной платформой. Руководство пользователя предлагает завершать регулировку путем ручного изменения величин для каждой из 25 точек в процессе тестовой печати, что сопряжено с риском повреждения целостности поверхности платформы из-за вероятного и нежелательного контакта с соплом. С помощью полуавтоматической регулировки улучшения качества печати достичь не удалось. Следует отметить, что такой способ является времязатратным и нерациональным для инженера, обслуживающего данный 3D-принтер, и, как результат, не подходит для регулярных перекалибровок, связанных с необходимостью нагрева до разных целевых температур.

Возникает необходимость в функционале автоматического выравнивания печатной платформы. Следует отметить, что данный функционал довольно распространен в современных FDM/FFF 3D-принтерах и предусмотрен производителем. Среди энтузиастов 3D-печати известны разнообразные по принципу действия датчики для автоматической калибровки, например, BLTouch от Antclabs [9]. Датчик BLTouch – это датчик автоматического выравнивания, который использует механический зонд для измерения расстояния между печатным слоем и соплом экструдера. Эта информация используется для регулировки высоты оси Z во время печати, что помогает обеспечить ровность первого слоя печати и его надлежащее прилипание к слою. Было решено заказать BLTouch-совместимый аналог (рисунок 2) [10].

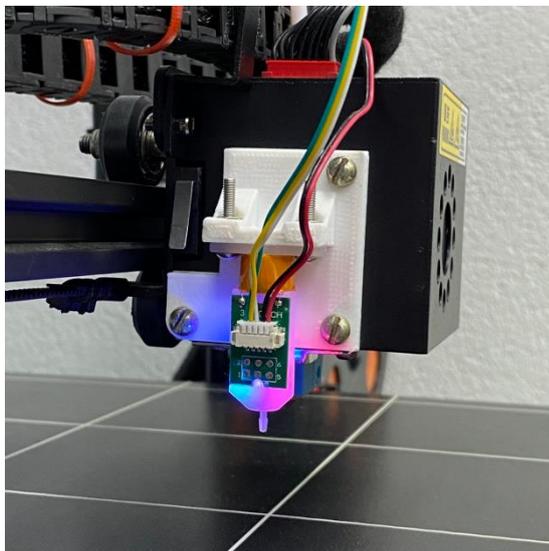


Рисунок 2 – BLTouch-совместимый датчик, установленный на Anycubic Chiron

Заводская плата управления Trigorrilla [11], лежащая в основе Anycubic Chiron, основана на популярной аппаратной платформе Arduino Mega 2560 и плате расширения Ramps 1.4, которые широко используются в сообществе 3D-печати. В свою очередь Ramps 1.4 поддерживает BLTouch-совместимый датчик, что делает данную модификацию 3D-принтера возможной.

Заводская прошивка Anycubic Chiron [7] не предполагает расширение функционала и программно не поддерживает BLTouch-совместимый датчик. Среди энтузиастов 3D-печати известна

прошивка, с открытым исходным кодом, изначально разработанная для 3D-принтеров FDM проекта RepRap с использованием платформы Arduino, под названием Marlin [12][13]. Marlin предлагает ряд функций и возможностей по кастомизации, которые недоступны в заводской прошивке (например, функционал автоматического выравнивания и программную поддержку BLTouch-совместимого датчика). Установка прошивки Marlin на Anycubic Chiron была сопряжена с изменениями в конфигурационных файлах в соответствии с аппаратным обеспечением и настройками Chiron [14]. Алгоритм первичной калибровки и его обоснование представлены в [15].

С помощью плагина [16] для OctoPrint [17] были получены визуализации искривления платформы по 24 точкам относительно стартовой точки (рисунки 3-5).

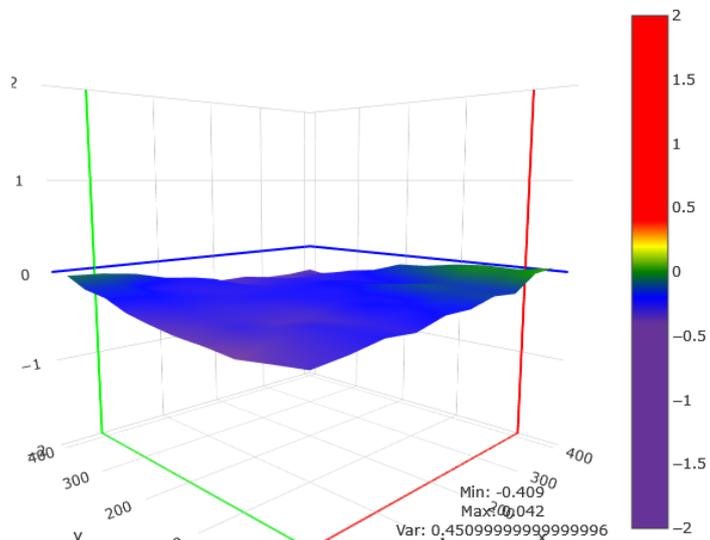


Рисунок 3 – Визуализация искривления платформы при 22 °C

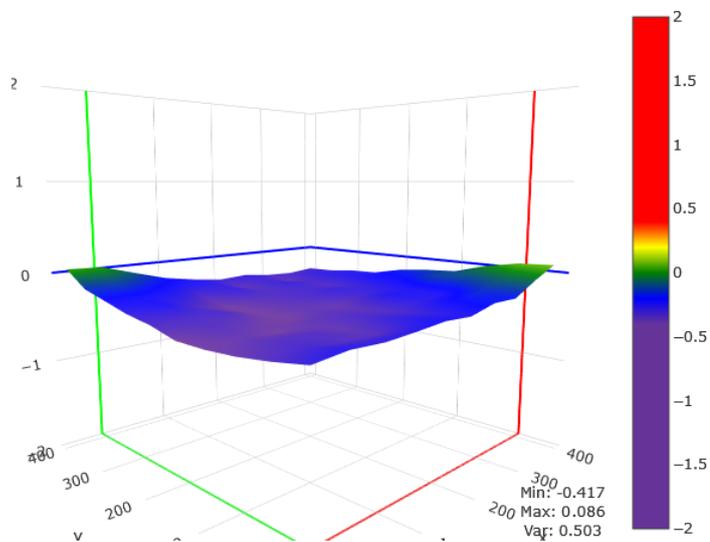


Рисунок 4 – Визуализация искривления платформы при 70 °C

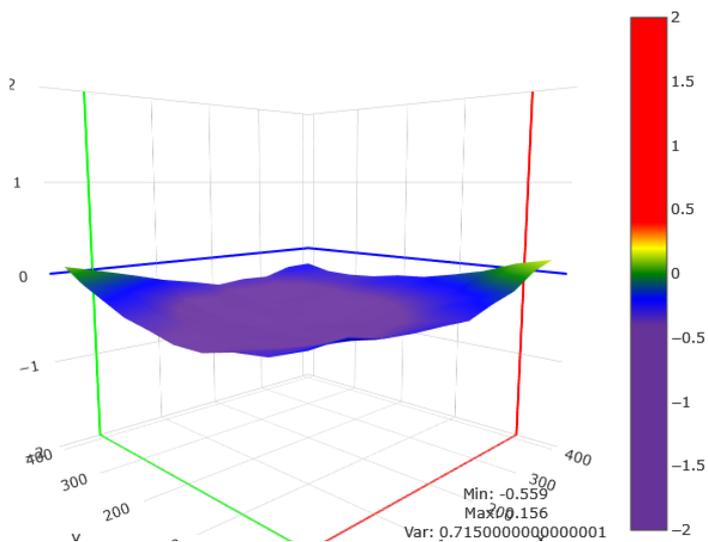


Рисунок 5 – Визуализация искривления платформы при 100 °С

На рисунке 3 показана неровность поверхности платформы для печати при 22 градусах Цельсия. На рисунках 4 и 5 показаны визуализации искривления платформы при достижении целевых температур 70 и 100 градусов Цельсия – температура платформы для печати PLA-пластиком и ABS-пластиком соответственно. Следует отметить, что на качество 3D-печати влияет множество факторов, один из них – плоскостность поверхности построения [18].

В результате выполнения работы было установлено, что ухудшение качества 3D-печати связано с искривлением печатной платформы. Предложена установка датчика BLTouch и периодическая автоматическая калибровка, что позволяет компенсировать искривление платформы при печати. Эксплуатация аппаратно и программно модифицированного Anycubic Chiron показала улучшение качества распечатанных моделей.

Данная работа выполнялась на базе студенческого инженерно-образовательного центра кафедры ЭВМ БГУИР.

Список использованных источников:

1. Как готовят айтишников в БГУИРе - Onliner [Электронный ресурс]. – Режим доступа: <https://tech.onliner.by/2019/06/24/bguir>. – Дата доступа: 06.04.2023.
2. Anycubic Chiron - Exciting Features and Larger Build Size 3d Printer [Электронный ресурс]. – Режим доступа: <https://www.anycubic.com/products/anycubic-chiron-3d-printer>. – Дата доступа: 06.04.2023.
3. Ultrabase Glass Plate with Heated Bed for Chiron – ANYCUBIC 3D Printing [Электронный ресурс]. – Режим доступа: <https://www.anycubic.com/collections/for-chiron/products/ultrabase-hotbed-platform-plate-for-chiron>. – Дата доступа: 06.04.2023.
4. Виды столов для 3D принтера - правильная калибровка стола [Электронный ресурс]. – Режим доступа: <https://cvetmir3d.ru/blog/poleznoe/vybiraem-stol-dlya-3d-printera-kalibrovka-i-nastroyka/>. – Дата доступа: 06.04.2023.
5. ANYCUBIC Chiron User Manual [Электронный ресурс]. – Режим доступа: <https://drive.google.com/open?id=1ArxYckBoXfHWiq06bcsk7FtgU1yg7bhO>. – Дата доступа: 06.04.2023.
6. Anycubic Chiron - Exciting Features and Larger Build Size 3d Printer [Электронный ресурс]. – Режим доступа: <https://www.anycubic.com/products/anycubic-chiron-3d-printer>. – Дата доступа: 06.04.2023.
7. Firmware of Anycubic 3D printer Chiron [Электронный ресурс]. – Режим доступа: https://github.com/ANYCUBIC-3D/ANYCUBIC_CHIRON_V1.3.0. – Дата доступа: 06.04.2023.
8. Auto Leveling Module for Chiron | ANYCUBIC 3D Printing [Электронный ресурс]. – Режим доступа: <https://www.anycubic.com/products/auto-leveling-module-for-chiron>. – Дата доступа: 06.04.2023.
9. BLTouch V2 | antclabs [Электронный ресурс]. – Режим доступа: <https://www.antclabs.com/bltouch>. – Дата доступа: 06.04.2023.
10. RAMPS 3D Touch Sensor Auto Bed Leveling Sensor Auto Touch Smart Z-probe Bed Leveling for Ender 3/5 3D Printer Parts [Электронный ресурс]. – Режим доступа: https://aliexpress.ru/item/1005003826327332.html?srcSns=sns_More&businessType=ProductDetail&spreadType=socialShare&tt=MG&utm_medium=sharing&sku_id=12000027285549113. – Дата доступа: 06.04.2023.
11. TriGorilla Motherboard for Mega/Kossel/Chiron/4Max Pro – ANYCUBIC 3D Printing [Электронный ресурс]. – Режим доступа: <https://www.anycubic.com/products/trigorilla-motherboard-for-mega-kossel>. – Дата доступа: 06.04.2023.
12. Marlin Firmware [Электронный ресурс]. – Режим доступа: <https://marlinfw.org/>. – Дата доступа: 06.04.2023.
13. RepRap [Электронный ресурс]. – Режим доступа: <https://reprap.org/wiki/RepRap>. – Дата доступа: 06.04.2023.
14. Marlin 2.0.x Version for Anycubic i3 MEGA M/S/P/X/CHIRON and 4MAX with Anycubic TFT or the "new" DGUS Clone TFT - Now also with BLTouch! [Электронный ресурс]. – Режим доступа: <https://github.com/knutwurst/Marlin-2-0-x-Anycubic-i3-MEGA-S>. – Дата доступа: 06.04.2023.
15. BLTouch Calibration Worksheet [Электронный ресурс]. – Режим доступа: <https://github.com/knutwurst/Marlin-2-0-x-Anycubic-i3-MEGA-S/files/9547522/BLTouchCalibrationWorksheet.pdf>. – Дата доступа: 06.04.2023.
16. Bed Level Visualizer - OctoPrint Plugin Repository [Электронный ресурс]. – Режим доступа: <https://plugins.octoprint.org/plugins/bedlevelvisualizer/>. – Дата доступа: 06.04.2023.
17. OctoPrint [Электронный ресурс]. – Режим доступа: <https://octoprint.org/>. – Дата доступа: 06.04.2023.

UDC 004.9

COMPENSATING FOR BED WARPING TO IMPROVE 3D PRINTING QUALITY ON ANYCUBIC CHIRON AUTOMATICALLY WITH BLTOUCH SENSOR

Kaliutchyk A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Pertsau D. – PhD in Computer Science

Annotation. The article examines the problem of warping the printing surface of the Anycubic Chiron printer and its negative impact on the quality of 3D printing. It is proposed to install the BLTouch sensor and periodically automatically adjust the level of the platform, which allows compensating for the warping of the platform during printing.

Keywords. 3D printing, print quality, Anycubic Chiron, ABL, BLTouch, bed warping

LED-ЧАСЫ С ИМИТАЦИЕЙ РАБОТЫ ИНДИКАТОРНЫХ ЛАМП

Бузук А.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Стракович А.И. – ассистент кафедры ЭВМ

Приведены основные результаты схемотехнического и технического моделирования, а также принцип работы Led-часов.

Данное устройство – это развитие часов на газоразрядных лампах[1], в котором вместо ламп используются подсвечивающиеся пластины из оргстекла с гравировкой цифр. Устройство отображает текущее время в формате ЧЧ:ММ:СС, а также имеет функции будильника, секундомера и графического анализатора спектра аудио сигнала.

Основным управляющим устройством может выступать практически любой микроконтроллер с GPIO. Наиболее простым решением является использование платформы Arduino с микроконтроллером ATmega328[2].

Для учета времени в устройстве можно использовать любой модуль часов реального времени совместимый с используемым контроллером. В данном устройстве используется модуль RTC DS1302[3].

Для реализации функции графического анализатора спектра аудио сигнала контроллер получает данные с микрофона. Обработка полученных данных производится при помощи быстрого преобразования Хартли. Данный режим включается при смене положения устройства с вертикального на горизонтальное.

Важной частью устройства являются подсвечивающиеся пластины из оргстекла, которые расположены плотно друг к другу, из-за чего возможна засветка соседних пластин. Для решения данной проблемы были сделаны 2 специальные детали корпуса. Первая деталь плотно прилегает к плате, для чего отверстия в детали полностью повторяют расположение и размер светодиодов, и необходима для предотвращения распространения света между светодиодами. Вторая деталь повторяет отверстия нижней детали, но с меньшей шириной, и необходима для контроля рассеивания света от светодиодов и устранения подсвечивания соседних пластин из оргстекла.

Пластины размещаются блоками по 10 штук для отображения цифр от 0 до 9. Всего в проекте используется 60 пластин для отображения цифр, и 2 пластины для отображения разделительных точек. Подсветка пластин реализована при помощи адресных RGB светодиодов WS2812B. Для подсветки одной пластины используется 1 светодиод.

Корпус устройства выполнен из фанеры толщиной 4 мм и собирается при помощи соединения ШИП-ПАЗ на клей. Для монтажа верхней и нижней части корпуса предусмотрены соединительные стойки. Соединение всех деталей корпуса устройства, а также расположение специальных деталей представлены на рисунке 1.

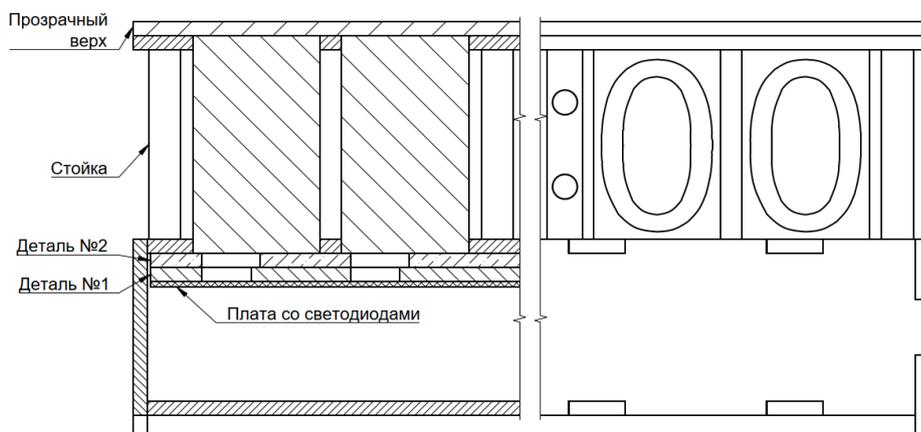


Рисунок 1 – Внешний вид корпуса устройства

Список использованных источников:

1. Часы ручной работы на основе газоразрядных индикаторов [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://past-indicator.ru>.
2. Документация Arduino [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://docs.arduino.cc/>
3. Mh real time clock modules 2 [Электронный ресурс]. – Электронные данные. – Режим доступа : <https://domoticzfaq.ru/hw084-podklyucheniye-k-arduino/>

АЛГОРИТМЫ ПОВЕДЕНИЯ ИГРОВЫХ ПЕРСОНАЖЕЙ

Богдан Е.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Фролов И.И. – канд. тех. наук, доцент

Дается обобщенная характеристика основных видов и подходов к созданию игрового искусственного интеллекта. Рассматривается проблема использования различных подходов к написанию алгоритмов поведения игровых персонажей. Также излагаются взгляды на использование нейронных сетей и машинного обучения для создания персонажей.

На сегодняшний день компьютерные игры перестали быть чем-то необычным. Их использование для досуга стало абсолютной нормой для людей. И это привело к тому, что разработка игр стала прибыльной и появилось огромное количество компаний для их разработки. Каждая из них обычно специализируется на разработке игр определенного жанра, которых также стало много. Вместе с тем, из-за количества создаваемых игр и их разнообразия, пользователи, приобретающие данный продукт, стали более избирательными. Их уже не удивляют простые игры без особых интересных механик, без проработанного игрового мира и прочих вещей.

Одним из аспектов, которые интересуют игроков, является проработанный алгоритм поведения персонажей. Алгоритмы поведения в игре принято также называть игровым искусственным интеллектом [1]. При этом его качество зависит не только от логики поведения, но и от соответствия игровому миру, систем анимаций. В некоторых случаях особенности поведения групп персонажей являются основной механикой игрового искусственного интеллекта. Стоит также упомянуть, что чаще всего делается упор на несколько составляющих, уменьшая качество других.

Как можно понять, искусственный интеллект не может быть унифицирован. Его разработкой часто занимается отдельная группа людей. В нее могут входить не только программисты, но и аниматоры, специалисты по звукозаписи и др. Если говорить про подходы к реализации сложного искусственного интеллекта [2], то их существует огромное множество. Наиболее распространенными являются:

- Деревья поведений
- Графы состояний
- Системы планирования

Конечно, все они так или иначе используют систему с граничным числом состояний и похожи друг на друга. Однако различия при детальном разборе выявляются почти сразу. При создании систем с игровым искусственным интеллектом могут использоваться сразу несколько подходов. Например, при разделении полномочий управления персонажем. Систему так или иначе необходимо разделять на блоки, которые могут использовать разные подходы при реализации. Примером является использование графов состояний для описания работы анимаций, но при этом логика поведения может использовать иной подход — деревья поведений или системы планирования.

Использование более сложных систем чаще всего не является оптимальным. Например, нейронные сети при разработке такого искусственного интеллекта почти не используются и на данный момент не известно ни одной успешной игры, где логика персонажей была бы завязана на их использовании. Осложняется это на этапе обучения по причине отсутствия какого-либо учителя или отсутствия времени для обучения в принципе.

Машинное обучение не используется по схожей причине. Оно так или иначе должно использовать некий опыт, базу данных для принятия решения о текущем действии. Что на этапе разработки почти невозможно. Полностью логику персонажей также сложно описать используя лишь методы машинного обучения.

Однако применение нейронных сетей или машинного обучения для части задач, выполняемых персонажем, остается возможным в особых ситуациях. Но на практике также используется довольно редко. Обычно это небольшие проекты симуляции для обучения или схожие по задаче игровые приложения.

Проблема выбора конкретного подхода для реализации игрового искусственного интеллекта полностью ложиться на разработчика. Неправильный выбор может повлечь за собой риски многократного увеличения времени разработки программного продукта и как следствие, возможное уменьшение дохода

Список использованных источников:

1. Роли алгоритмов поведения игровых персонажей [Электронный ресурс] – Режим доступа: https://www.irjmets.com/uploadedfiles/paper/issue_7_july_2022/27897/final/fin_irjmets1658458318.pdf

2. Подходы к созданию игрового искусственного интеллекта [Электронный ресурс] – Режим доступа: <https://www.gamedev.net/tutorials/programming/artificial-intelligence/the-total-beginners-guide-to-game-ai-r4942/>

АППАРАТНОЕ УСКОРЕНИЕ НЕЙРОННЫХ СЕТЕЙ

Русакович А. В.

*Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь*

Перцев Д.Ю., к.т.н., доцент

В данном материале рассмотрены основные методы ускорения обучения и выполнения нейронных сетей, а также история их развития.

В последнее десятилетие все более актуальной становится проблема обучения и выполнения нейронных сетей, особенно в условиях ограниченных аппаратных ресурсов.

С появлением первых нейронных сетей, таких как многослойный перцептрон [1], проблема ограниченности ресурсов была сдерживающим фактором для увеличения сложности модели и, следовательно, их точности и производительности. В 1980-е годы для обучения и выполнения первых нейронных сетей чаще всего применялся центральный процессор (CPU), подходящий для общих вычислений. Однако в пользу общности жертвовалось производительностью специфических вычислений, например, векторных. Ситуация поменялась с появлением графических процессоров (GPU), аппаратная архитектура которых нацелена на работу с 3D объектами и одновременной работой с множеством данных. Известным примером является AlexNet [2], являющийся одной из первых сверточных нейронных сетей, пригодной к применению на GPU и позволившей выиграть соревнование по распознаванию изображений ImageNet. Это дало толчок к развитию как самих GPU, так и к поиску других аппаратных ускорителей.

Успех сверточных нейронных сетей, накопление большого объема данных создавало запрос на улучшение аппаратных возможностей. Большинство нейронных сетей оперируют сверточными слоями, полносвязными слоями, а также различными функциями активации между ними. Поскольку и сверточные слои, и полносвязные слои можно представить как операцию матричного или тензорного умножения, одно из направлений ускорения была работа именно над оптимизацией этой операции. Так, в 2018 году NVIDIA презентовала новое поколение графических ускорителей серии RTX 2xxx, особенностью которой являлось наличие так называемых тензорных ядер [3], выполняющих операцию сплавленного сложения и умножения в матричном виде.

Поскольку основной специализацией графического процессора является графика, а не вычисления, большего ускорения можно достичь за счет дальнейшей специализации. Это привело к созданию и использованию тензорных вычислительных модулей (TPU) [4], используемых для обучения, и нейронных процессоров (NPU) для выполнения. Оба типа устройств отличаются характерным высоким соотношением производительности к единице потребляемой энергии, но при этом в своем использовании ограничены только обучением или выполнением модели нейронной сети.

Одно TPU ядро состоит из множества матричных вычислительных блоков, число которых зависит от версии, а также одного векторного и скалярного блоков. Ядра оперируют числами с пониженной точностью, в частности 8 или 16 бит, что позволяет упростить оборудование. Вместе с использованием квантования параметров и вычислений это позволяет значительно ускорить вычисления с небольшой потерей точности. Также, TPU ядра могут масштабироваться и объединяться в кластер для повышения эффективности.

В то же время NPU используются для выполнения нейронных сетей на маленьких устройствах, таких как смартфоны или Edge устройства. Для NPU также характерно использование вычислений с пониженной точностью в 8, 16 бит или даже под-байтовой точностью, а также использование SIMD/векторных блоков.

Таким образом наблюдается явное движение в сторону использования специализированных устройств на базе аппаратных решений для ускорения. В будущем можно ожидать дальнейшую специализацию решений только для определенного класса задач или архитектур. Например, устройств с аппаратным ускорением свертки или обработки звука.

Список использованных источников:

1. Хайкин, С. Нейронные сети. Полный курс: учебное пособие. / С. Хайкин — Дialeктика, 2019. — 1104 p.
2. Классификация ImageNet с помощью глубоких сверточных нейронных сетей [Электронный ресурс] — Режим доступа: https://papers.nips.cc/paper_files/paper/2012-Abstract.html — Дата доступа: 22.03.2023.
3. Тензорные ядра NVIDIA [Электронный ресурс] — Режим доступа: <https://www.nvidia.com/en-us/data-center/tensor-cores> — Дата доступа: 20.03.2023.
4. Тензорные вычислительные модули, используемые для обучения [Электронный ресурс] — Режим доступа: <https://cloud.google.com/tpu/docs/system-architecture-tpu-vm> — Дата доступа: 21.03.2023.

РЕАЛИЗАЦИЯ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С КАМЕРОЙ ВИДЕОНАБЛЮДЕНИЯ

Ахрамейко П.Д., студент гр.950501

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Луцик Ю.А. – канд. технических наук

Тезисы посвящены описанию разработки программно-аппаратного приложения для реализации устройства видеонаблюдения, состоящего из одноплатного компьютера Raspberry Pi Zero W, модуля камеры Raspberry Pi Camera Rev 1.3, а также специально написанного веб-сервиса и программы для управления камерой.

Одной из основных потребностей человека является и являлась потребность в безопасности. Данная потребность выражается не только в желании защитить себя, но и защитить своих близких и свое имущество. Помимо этого, можно защищать также и свои интересы, свою точку зрения в конфликтах.

В разные времена существовали различные способы достижения безопасности, но в данный момент одним из самых распространённых способов является видеонаблюдение.

Благодаря быстрому темпу развития технологий любому человеку доступно устройство видеонаблюдения. Существует большое количество разнообразных устройств из разных ценовых категорий с различными характеристиками.

Для некоторых задач может понадобиться дополнительная обработка изображения, полученного с системы видеонаблюдения. Это может выполняться разными способами и для различных целей. Целью может являться выделение каких-то определённых черт у изображения, удаление шумов, выделение и в дальнейшем распознавание объекта. Достигается это с помощью алгоритмов обработки изображения. В некоторых сложных случаях для этого применяются нейронные сети. Данные возможности могут понадобиться так-же, например, при установке камеры на какое-либо беспилотное устройство, что возможно благодаря небольшим габаритам устройства.

Целью проекта является реализация веб-сервиса для работы с камерой видеонаблюдения. Для взаимодействия с камерой посредством сервиса будет создано приложение, обладающее минимальным необходимым функционалом. Камера будет способна передавать изображение в реальном времени, делать снимки, сохранять их в базу данных с дальнейшей возможностью их анализа. Местом хранения снимков была выбрана база данных, поскольку это позволит хранить большие объёмы информации, чем на мобильном устройстве или SD-карте. Пользователь сможет сам извлекать изображения из базы данных для их анализа и дальнейшей обработки.

Подобная система может быть полезна пользователю, который хочет иметь доступ к камере, находясь на значительном расстоянии от неё, с помощью предусмотренного интерфейса.

Главным преимуществом данного проекта будет являться также и небольшой размер камеры, что позволит установить её на какое-либо дистанционно управляемое устройство. Данный факт будет влиять на выбор аппаратных средств.

Задачи разработки: 1) выбор подходящих аппаратных средств; 2) реализация веб-сервиса для управления камерой; 3) организация дистанционного доступа к полученным с камеры изображениям; 4) реализация просмотра и сохранения полученных изображений; 5) предоставление возможности фильтрации изображений; 6) добавление базы данных для хранения изображений; 7) разработка мобильного приложения для получения доступа к камере.

Первой задачей является выбор аппаратных средств. Для создания видеокamеры были выбраны два модуля: одноплатный компьютер и камера. В проекте используются: одноплатный компьютер Raspberry Pi Zero W [1] (см. рисунок 1а), предоставляющий необходимые вычислительные ресурсы для выполнения поставленных задач, а также камера Raspberry Pi Camera Rev 1.3 (см. рисунок 1б), которая обеспечивает нужное качество получаемых изображений.



а)



б)

Рисунок 1 – Модули камеры. а) Raspberry Pi Zero W, б) Raspberry Pi Camera Rev 1.3

После выбора аппаратных средств необходимо выбрать технологии программирования, которые будут использоваться при разработке мобильного приложения и сервера. Вся система разрабатывается с использованием клиент-серверной архитектуры. Её суть заключается в том, что мобильное приложение общается с сервером посредством запросов. Непосредственное выполнение необходимых задач происходит на стороне сервера, а результаты доступны приложению.

Сервер, как было сказано выше, представляет собой устройство – вычислительное ядро системы, которое выполняет все необходимые в проекте задачи: выполняет обработку данных и их представление. При выборе языка программирования для разработки сервера предпочтение было отдано языку разработки Python, а также фреймворку Flask. Flask — это микрофреймворк для создания простого и быстрого проекта на языке программирования Python с возможностью масштабирования до сложных приложений. Понятие «микрофреймворк» означает, что в комплекте нет набора инструментов и библиотек, программист может сам установить их в зависимости от задач.

В приложении используется удалённая база данных, которая предоставляется Firebase. База данных будет использоваться для хранения изображений и последующего доступа к ним с помощью мобильного приложения.

Клиентом в данном случае будет выступать мобильное устройство пользователя с приложением, которое будет являться клиентской частью проекта. Мобильное приложение будет посылать запросы на сервер для получения всей необходимой информации и данных. Для разработки мобильного приложения используется язык программирования Kotlin, который сейчас является рекомендованным для мобильной разработки, а также различные фреймворки для работы с серверной частью, разработки пользовательского интерфейса и иных задач. Примером подобных фреймворков может послужить Retrofit, который используется для обеспечения взаимодействия с сервером посредством запросов.

С использованием перечисленных выше технологий были разработаны веб-сервис и клиентское приложение для осуществления управления камерой. В процессе разработки были предусмотрены следующие возможности: 1) возможность сохранять изображения на мобильное устройство; 2) возможность задать сохраняемому изображению название; 3) возможность фильтрации изображения.

Пример работы мобильного приложения (одного из его окон) приведен на рисунке 2.

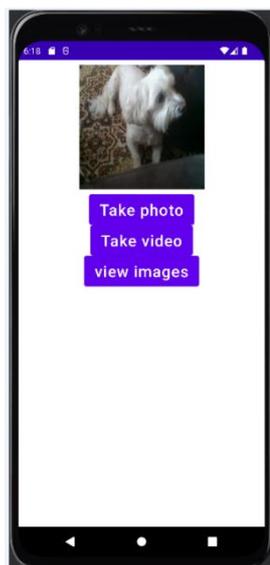


Рисунок 2 – мобильное приложение для управления камерой видеонаблюдения

Главной особенностью объекта разработки является возможность его установки на различные устройства, которые управляются удалённо. Это возможно благодаря его небольшим

размерам. Помимо этого, преимуществом является и то, что к нему можно подключаться удалённо и получать изображение, что также удобно при работе с устройствами с удалённым доступом.

В дальнейшем имеется возможность совершенствования уже существующего приложения (клиентской и серверной его частей), добавляя различный функционал. К такому функционалу можно отнести авторизацию и дополнительные опции обработки изображений.

Список использованных источников:

1. Vasilis Tzivaras. *Raspberry Pi Zero W Wireless Projects*. Packt Publishing, 2017. – 240 с.

ИССЛЕДОВАНИЕ ДРЕБЕЗГА КОНТАКТОВ С ЦЕЛЬЮ ГЕНЕРИРОВАНИЯ СЛУЧАЙНЫХ ЧИСЕЛ

Можейко Д.О.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Иванюк А.А. – доктор технических наук, доцент

В ходе работы были проанализированы кнопки и переключатели. Была спроектирована схема экспериментальной установки на языке описания аппаратуры VHDL, с помощью которой при нажатии на заданную кнопку либо переключатель, на семисегментные индикаторы выводится количество передних N_R (rising) и задних N_F (falling) фронтов сигнала. Была собрана статистика дребезжания кнопок и переключателей для 100 нажатий в обычном режиме. По результатам проведенных экспериментов были рассчитаны значения матожидания $\mu(N_R)$ и $\mu(N_F)$, вычислены стандартные отклонения $\delta(N_R)$ и $\delta(N_F)$, а также составлены графики числа спадов (N_F) и подъемов (N_R).

Генерация случайных чисел — процесс, который с помощью устройства генерирует последовательность чисел или символов, которая может быть предсказана разумным образом только на основании случайности. Генерация случайных чисел имеет большое значение в различных областях науки. Современные научные и инженерные области требуют большое количество симуляций для изучения различных физических, химических и биологических процессов. Для таких симуляций требуются случайные числа, чтобы имитировать различные условия и состояния. Именно поэтому важна генерация действительно случайных последовательностей.

Генераторы случайных чисел (ГСЧ) подразделяются на физические генераторы случайных чисел, которые вырабатывают случайные числа в зависимости от текущего значения какого-либо атрибута физического объекта либо среды, который практически невозможно смоделировать при текущем уровне знаний, и генераторы псевдослучайных чисел.

Генераторы псевдослучайных чисел основаны на определенных алгоритмических правилах, которые вырабатывают случайные числа после задачи исходного условия. Они не являются истинными, так как числа, которые они генерируют, только выглядят случайными, но на самом деле являются детерминированными и могут быть воспроизведены, если известна математическая модель, на основании которой работает генератор.

Генератор действительно случайных чисел (ГДСЧ) порождает последовательность случайных чисел на основе измеряемых, хаотически изменяющихся параметров протекающего физического процесса. Генераторы случайных чисел, использующие физические случайные процессы, позволяющие получать случайные числа с заданным распределением, являются относительно сложными и дорогими. Но существуют и более доступные источники случайности, например такие как: использование шума микрофона, использование времени нажатия на кнопки клавиатуры или мышки, использование сети или интернета, а также анализ дребезга контактов [3]. Самый легкий и доступный генератор действительно случайных чисел основывается на анализе дребезга контактов. Дребезг – явление, возникающее в электромеханических коммутационных устройствах и аппаратах (кнопках, реле, герконах, переключателях, контакторах, магнитных пускателях и др.) при замыкании и размыкании электрических контактов. После замыкания за счет упругости материалов и деталей контактной системы происходят многократные неконтролируемые замыкания и размыкания контактов — некоторое время контакты отскакивают друг от друга при соударениях, размыкая и замыкая электрическую цепь. Другими словами, кнопки и переключатели находятся в состоянии неопределенности, а переключатель не в состоянии выбрать одно из дискретных состояний (0 либо 1), как показано на рисунке 1. Дребезг зачастую является нежелательным явлением в технических устройствах. В связи с этим существуют различные способы его подавления (как аппаратные, так и программные) [2].

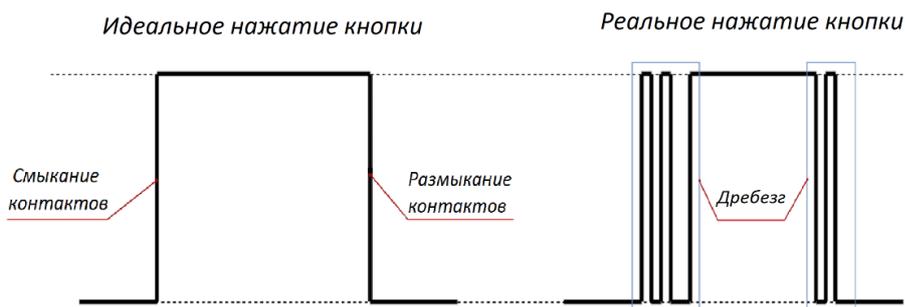


Рисунок 1 – Нажатие кнопки

Частота и количество касаний контактов зависит от множества факторов, например таких, как свойства компонентов коммутирующего узла, уровень напряжения, упругость пружины, сила нажатия на кнопку, температура, степень дрожания пальцев при нажатии, время удержания и т.д. Все эти факторы вносят свою долю непредсказуемости в генерацию случайных последовательностей. Это помогает генератору случайных чисел, основанному на анализе дребезга контактов, достичь цели действительной случайности.

Исследование дребезга контактов проводилось на учебной плате быстрого прототипирования Artix-7 FPGA (Digilent Nexys 4) [1]. В ходе работы были проанализированы имеющиеся на плате кнопки и переключатели. Была спроектирована схема экспериментальной установки на языке описания аппаратуры VHDL, с помощью которой при нажатии на заданную кнопку либо переключатель, на семисегментные индикаторы выводится количество передних N_R (rising) и задних N_F (falling) фронтов сигнала. Была собрана статистика дребезжания кнопок и переключателей для 100 нажатий в обычном режиме. По результатам проведенных экспериментов были рассчитаны значения матожидания $\mu(N_R)$ и $\mu(N_F)$, вычислены стандартные отклонения $\delta(N_R)$ и $\delta(N_F)$, которые представлены в таблице 1, а также составлены графики числа спадов (N_F) и подъемов (N_R), представленные на рисунках 2 и 3.

Таблица 1 – Статистика дребезга

	Кнопка 1		Переключатель 1	
	N_R	N_F	N_R	N_F
μ	2,26	2,17	33,71	34,71
δ	2,47664	2,31837	54,7638	55,8143

Собранная статистика показывает, что каждая кнопка (переключатель) ведет себя совершенно по-разному, однако кнопки ведут себя более стабильнее, чем переключатели, так как у переключателей намного больше амплитуда нажатия, что в свою очередь вызывает сильное

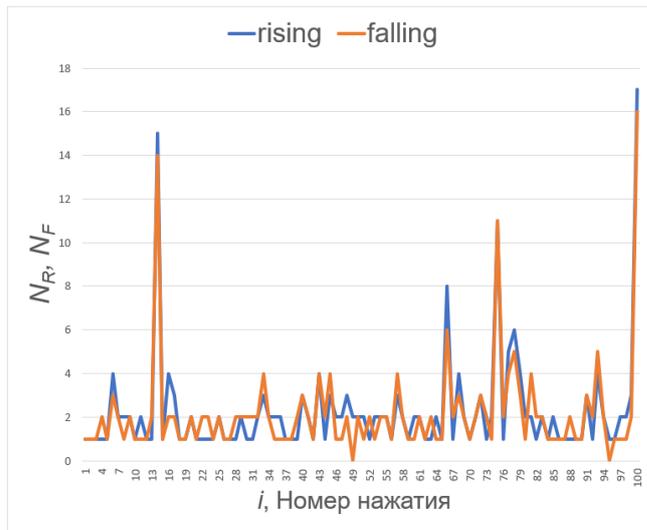


Рисунок 2 – Соотношение спадов и подъемов сигнала (Кнопка 1)

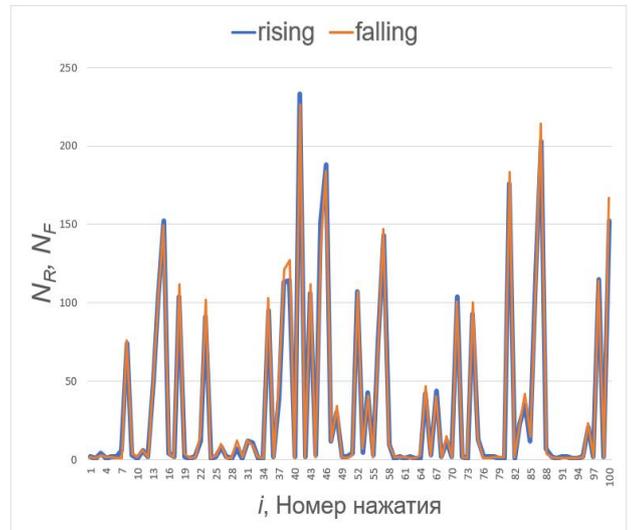


Рисунок 3 – Соотношение спадов и подъемов сигнала (Переключатель 1)

дребезжание.

В дальнейших исследованиях неповторимость дребезга будет использована для создания генератора случайных чисел. Данный способ генерирования случайных чисел очень удобен (за исключением скорости генерирования чисел), поскольку появляется возможность генерирования чисел абсолютно везде, где это необходимо, так как все что для этого необходимо это кнопка и пользователь, который будет нажимать эту кнопку. Также в последующих работах будет

усовершенствован и автоматизирован способ сбора результатов нажатия кнопки (количество отскоков контактов). Так как в данной работе вся статистика собиралась вручную, следовательно, автоматизация данного процесса значительно ускорит анализ и генерацию случайных чисел.

Создание ГДСЧ является довольно важной темой, так как существует много областей применения, где случайные и невоспроизводимые числа имеют важное значение, что говорит о актуальности данного исследования.

Список использованных источников:

1. Digital reference: <https://digilent.com/reference/programmable-logic/zybo-z7/start>
2. Jack G. Ganssle. *A Guide to Debouncing*.
3. Stipčević, Mario, and Çetin Kaya Koç, 2014. *True random number generators* .

ПРОГРАММНО-АППАРАТНАЯ СИСТЕМА СТАБИЛИЗАЦИИ ПОЛЕТА БЕСПИЛОТНОГО ЛЕТАТЕЛЬНОГО АППАРАТА

Пржелясковская В.Д., студент группы 050501

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Луцик Ю.А. – доцент, канд. технических наук

Объектом исследования являются беспилотные летательные аппараты (БПЛА). Предметом исследования является технология стабилизации полета квадрокоптера. Цель исследования: разработать квадрокоптер с внедренной системой стабилизации полета, которая обеспечит устойчивый полет и эффективное управление полетом в различных условиях, таких как изменение направления и скорости ветра, механические воздействия на беспилотный летательный аппарат, изменение температуры и давления в окружающей среде. Для достижения поставленной задачи была выполнена доработка контроллера полета для реализации возможности ориентации квадрокоптера в пространстве и разработан вариант программного обеспечения для управления стабилизацией летательного аппарата.

Беспилотные летательные аппараты набирают все большую популярность в различных областях, таких как разведка, доставка, аэрофотосъемка и многие другие. Квадрокоптеры являются одними из самых распространенных типов беспилотных летательных аппаратов благодаря своей маневренности и компактности. Однако, для эффективного использования квадрокоптеров в различных условиях, необходимо обеспечить их стабильный полет. В данном исследовании рассмотрена проблема стабилизации полета квадрокоптера при помощи программно-аппаратной системы, основанной на использовании таких датчиков, как гироскоп, акселерометр и барометр.

Гироскоп – устройство, способное реагировать на изменение углов ориентации тела, на котором оно установлено, относительно инерциальной системы отсчета.

Оси вращения гироскопа с тремя степенями свободы обеспечиваются двумя рамками карданова подвеса. Центральный элемент гироскопа вращается по своей вертикальной оси, при этом он фиксируется в рамке, которая способна поворачиваться только по горизонтальной оси. Она закрепляется в еще одной рамке, которая может оборачиваться вокруг третьей оси. Такая конструкция прибора позволяют его центральному элементу всегда находиться в вертикальном положении, вне зависимости от того, как будет поворачиваться корпус гироскопа [1].

Три оси, вокруг которых может вращаться гироскоп, имеют определенные названия: крен, тангаж и рысканье. Ось крена проходит через гироскоп перпендикулярно продольной оси квадрокоптера. Ось тангажа проходит через гироскоп параллельно продольной оси квадрокоптера. Ось рысканья проходит через гироскоп перпендикулярно горизонтальной плоскости.

Несбалансированность моторов может вызвать вращение вокруг своей оси при движении квадрокоптера в определенном направлении, что может привести к нарушению устойчивости и падению. Гироскоп измеряет изменение угла поворота, что позволяет определить, как сильно и в каком направлении квадрокоптер начал данное вращение. Эти данные затем передаются микроконтроллеру, который использует их для корректировки движения моторов и стабилизации полета.

Акселерометр – прибор для измерения ускорения тела в трех направлениях (вперед/назад, влево/вправо и вверх/вниз), который работает как датчик изменения положения устройства в пространстве [2].

Акселерометр в квадрокоптерах позволяет определить текущую скорость, а также положение БПЛА в пространстве. Его показания применяют для стабилизации полета и управления им. Например, когда пилот наклонил джойстик пульта управления вперед, акселерометр измеряет ускорение в этом направлении, и микроконтроллер использует эту информацию для корректировки движения моторов и удержания квадрокоптера в горизонтальном положении.

Кроме того, акселерометры также используются для измерения вертикальной скорости взлета и посадки квадрокоптера, что позволяет контроллеру полета автоматически регулировать мощность моторов и обеспечивать плавную посадку.

Барометр – прибор, используемый в БПЛА для измерения высоты полета. Он фиксирует изменения атмосферного давления, которые микроконтроллер использует для автоматического регулирования мощности моторов и поддержания нужной высоты полета.

Цель данной работы заключается в разработке квадрокоптера, оснащенного системой стабилизации полета. Для эффективного тестирования данной системы на 3D принтере был разработан вспомогательный стенд, представленный на рисунке 1 (а), позволяющий зафиксировать квадрокоптер и безопасно наблюдать за результатами балансировки БПЛА, а также следить за его реакцией на определенные сигналы от пульта управления.

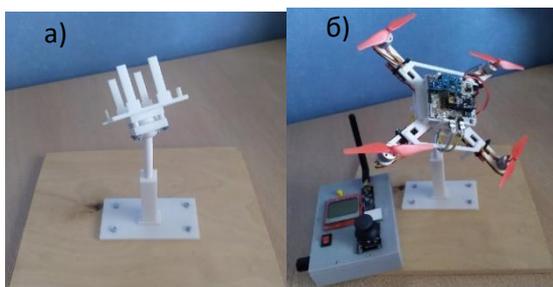


Рисунок 1 – а) Вспомогательный стенд для фиксации квадрокоптера в процессе стабилизации;
б) Квадрокоптер, установленный на вспомогательный стенд, а также пульт дистанционного управления

Разрабатываемый квадрокоптер в базовой комплектации оснащен управляющим микроконтроллером, моторами, контроллерами скорости, гироскопом, акселерометром, барометром и модулем связи для коммуникации с пультом управления [3]. Также был разработан пульт дистанционного управления, который имеет потенциометр для регулировки скорости моторов, джойстик для управления квадрокоптером в пространстве и ЖКИ-экран, на котором могут отображаться текущие характеристики полетов (скорости моторов, высота и т.д.).

Квадрокоптер, представленный на рисунке 1 (б) построен на легкой 7-дюймовой раме, которая была изготовлена на 3D принтере. Он оснащен четырьмя бесколлекторными моторами, способными развивать скорость до 4300 kV. Управление полетом и всеми датчиками квадрокоптера осуществляется с помощью микроконтроллера ATmega 328p из семейства AVR.

В качестве гироскопа и акселерометра был выбран модуль MPU-6050, который включает в себя данные датчики. Для измерения угла поворота используется технология MEMS и эффект Кориолиса. Выходные данные гироскопа измеряются в градусах в секунду, поэтому для получения углового положения необходимо интегрировать угловую скорость. С другой стороны, акселерометр MPU6050 производит измерения ускорения, которое может быть использовано для определения гравитационного ускорения по трем осям. Дополнительно, при использовании тригонометрических вычислений можно вычислить угол, под которым расположен датчик. При объединении данных акселерометра и гироскопа, возможно получить высокоточную информацию об ориентации датчика в пространстве. Модуль MPU6050 также называют устройством слежения за движением по шести осям или устройством с 6 степенями из-за его 6 выходов, или 3 выхода акселерометра и 3 выхода гироскопа.

Датчик барометра BMP180 возвращает значение атмосферного давления в единицах гектопаскаля (гПа), где 1 гПа равен 100 Паскалям. Уровень моря имеет стандартное давление около 1013 гПа, и каждый метр высоты уменьшает давление примерно на 0,11 гПа. Следовательно, высота над уровнем моря в метрах может быть вычислена путем вычитания 1013 из значения, возвращаемого датчиком BMP180, и деления оставшейся разницы на 0,11 [1].

Радиомодуль NRF24L01 отвечает за создание беспроводного соединения между квадрокоптером и пультом дистанционного управления.

В будущем планируется внедрить некоторые улучшения, включающие установку защиты на винты и аппаратную часть устройства. Кроме того, планируется добавление GPS-модуля для определения местоположения квадрокоптера. Это позволит осуществлять автономный полет и программирование маршрутов, а также GPS может использоваться для возвращения дрона в точку старта и предотвращения его потери в случае потери связи с пультом управления. Добавление камеры FPV (First Person View) позволит получать видео изображение с квадрокоптера в реальном времени и передавать его на экран управляющего устройства. Благодаря этому пилот сможет видеть окружающую среду с точки зрения камеры на борту дрона, что сделает управление более удобным и точным.

Возможно создание программных улучшений, таких как мобильное приложение для управления дроном с помощью смартфона или программа для отображения работы системы стабилизации БПЛА.

Таким образом будет получено устройство, пригодное для совершения безопасных и точных полетов.

Список использованных источников:

1. Родионов, Ю. А. Микроэлектронные датчики и сенсорные устройства: учеб. пособие / Ю. А. Родионов. – Минск: БГУИР, 2019. – 300 с.
2. Матвеев В.В. Основы построения бесплатформенных инерциальных навигационных систем / Под ред. В.Я. Распопова. – СПб. ГНЦ РФ ОАО “Концерн “ЦНИИ “Электроприбор”, 2009. — С. 118–157.
3. Building a Quadcopter with Arduino / Vasilis Tzivaras – Packt Publishing, 2016. – 124 p.

КИТАЙСКАЯ КОМНАТА И СИСТЕМЫ ШИФРОВАНИЯ

Протьюко М.А., студент гр.050502

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Борисенко О.Ф. – канд. физ.-мат. наук

Рассматривается мысленный эксперимент и его влияние на производительность систем машинного обучения в области шифрования.

Рассматривая мысленный эксперимент, называемый «китайской комнатой» [1], возможно прийти к выводам, способным исправить проблему решения трудно формализуемых задач, подробно рассмотренных в [2]. Трудность формализации данных задач (в точности задачи обучения алгоритму шифрования без использования выборки шифров в качестве «учителя») заключается в отсутствии понятия семантического смысла в системе, которая должна этим понятием оперировать.

Рассмотрим следующую задачу: в нашей системе есть изначальный алфавит. Каждому участнику системы разрешается создавать свою версию алфавита, посредством подстановок и перестановок символов (операции на множестве вычетов). Каждому участнику выдаются фрагменты изначального текста. Их задача сделать так, чтобы при обмене зашифрованным текстом соперник не смог его расшифровать. До участия в игре ни один из участников не знает ни одного шифра. Он знает лишь, что его соперник говорит с ним на одном языке.

Для поиска решения поставленной задачи были использованы принципы, изложенные в [3],[4],[5].

Посредством формализации данной задачи было разработано несколько вариантов систем машинного обучения, основанных на генетических алгоритмах и нейронных сетях.

Кратко опишем полученные системы.

Функция, для которой ведется поиск локальных экстремумов:

$$f(a, b) = \bigcup_{j,i}^{c,k} (a_i * b_j) \bmod n, \quad i = \overline{1, k}, j = \overline{1, c}, \quad (1)$$

где $f(a,b)$ – некая функция симметричного шифрования (биекция), a и b – множество символов текста и ключа соответственно, k – количество символов текста, c – количество символов ключа, n – мощность алфавита символов, $*$ – любая обратимая операция на множестве вычетов n .

Целевая функция:

$$\mu(s) = \frac{1}{n} \sum_{i=1}^n p(u_i) \quad (2)$$

где s – отбираемый генотип («хромосома» г.а.), $p(u_i)$ – вероятность встречи u_i символа, u_i – символ, полученный с помощью $f(a,b)$, где n – мощность множества a .

Из чего далее следует, что для отбора в популяцию генотип должен стремиться к (3):

$$\lim_{n \rightarrow \infty} \mu(s) = \frac{1}{n} \quad (3)$$

Рассмотрим некое пространство, в котором будут существовать фенотипы популяции описанного г.а. У каждого фенотипа, далее называемого особью, будет определено время жизни. Особь будет продолжать существование, пока она способна выдать результат (1). Конкуренция между особями будем заключаться в поочередном применении параметра (2) к особи. У каждой особи есть возможность вычислить как (1), так и (2). Во время итерации на этапе отбора (см. [3]), между двумя особями будет проводиться соревнование на основе (2). Причем способ расчета вероятности оставляется на формируемую генотипом нейронную сеть.

В итоге, можно сделать вывод, что описанные системы не способны выйти за рамки изначально предположенного решения задачи, что является весьма существенной проблемой в недостаточно изученных областях и/или областях, требующих множество промежуточных вычислений.

Потенциальным решением этой проблемы является подход, объединяющий нейронные сети, генетические алгоритмы и клеточные автоматы, подобно [6] и [7].

Список использованных источников:

1. Searle J. *Minds, Brains and Science*. / Перевод на русский язык: А. Ф. Грязнов. — М., 1993. Глава 2. «Могут ли компьютеры мыслить?».

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. *Протьюко, М.А., Борисенко, О.Ф. Простейшие шифры и генетический алгоритм / репозиторий БГУИР. 2023. – 24 с.*
3. *Панченко, Т.В. Генетические алгоритмы: учеб.пособие / под ред. Ю. Ю. Тарасевича. — Астрахань: Издательский дом «Астраханский университет», 2007. — 87 с*
4. *Уральский Н.Б., Сизов В.А., Капустин Н.К. Оптимизация вычислительного процесса фитнес функции генетического алгоритма в распределённых системах обработки данных // Интернет-журнал «Науковедение» ISSN 2223-5167; Том 7, N 6 (2015).*
6. *Головко В.А, Краснопрошин В.В. Нейросетевые технологии обработки данных: учеб.пособие – Минск : БГУ, 2017.– 263 с.*
7. *Nicolas Loizeau Building a computer in Conway's game of life – Источник : <https://www.nicolasloizeau.com/gol-computer> Дата доступа:7.04.2023.*
8. *Gabriel Morariu, Hugo Lamarche, Elizabeth Pyvovarov and Karl-Philippe Bluteau Creation of alternate Game of life realities and rules using Neural Networks - Источник : <https://medium.com/@flygongaby/creation-of-alternate-game-of-life-realities-and-rules-using-neural-networks-4db169f30adc>. Дата доступа: 7.04.2023.*

МЕТОД ДЛЯ ПРОВЕРКИ ЗАПРОСОВ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ К СЕМАНТИЧЕСКОЙ МОДЕЛИ

Кушнеревич П.М.

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Герман Ю.О. – канд. тех. наук, доцент

Работа посвящена разработке метода для проверки запросов на естественном языке к семантической модели. Предложенный метод включает лексический, синтаксический и семантический анализ текста для формирования запросов. Описаны основные шаги алгоритма для обработки естественно-языковых запросов, включая предварительную обработку, лексический и синтаксический анализ, семантическую интерпретацию и исполнение запроса. Приведены примеры использования метода в различных приложениях, в том числе в электронной коммерции и чат-ботах. Работа представляет интерес для специалистов в области искусственного интеллекта в части обработки естественного языка.

В современном мире существует необходимость взаимодействия человека с компьютерными системами на естественном языке. Для этого требуется разработка формального аппарата для проверки корректности запросов к семантической модели.

В рамках данной работы предложен метод проверки запросов на естественном языке для взаимодействия с семантическими моделями на основе искусственного интеллекта. Метод включает лексический, синтаксический и семантический анализ текста для формирования запросов.

Для реализации данного подхода предложен следующий алгоритм.

Шаг 1. Предварительная обработка запроса. На данном этапе происходит удаление стоп-слов, то есть слов, которые не несут смысловой нагрузки, таких как "и", "или", "но" и т.д. Кроме того, выполняется стемминг – процесс нахождения основы слова, чтобы можно было свести различные формы слова к единой форме. Например, для запроса "Какие магазины в Москве продают яблоки?" будут удалены стоп-слова "какие", "в" и "продают", а слово "магазины" будет приведено к основе "магазин". Для выполнения этого шага используются инструменты обработки естественного языка, такие как NLTK (Natural Language Toolkit) и SpaCy.

Шаг 2. Лексический и синтаксический анализ. Лексический анализ заключается в разбиении запроса на лексемы – минимальные смысловые единицы, такие как слова и знаки препинания [1].

Синтаксический анализ заключается в определении структуры запроса и его синтаксических свойств. На данном этапе происходит построение дерева синтаксического разбора, которое показывает зависимости между словами в запросе и их ролью в предложении. Для выполнения этого шага используются инструменты обработки естественного языка, такие как Stanford Parser и SyntaxNet.

Шаг 3. Семантическая интерпретация. На этом этапе происходит определение смысла запроса, то есть его семантики. Для этого используется семантическая модель, которая содержит знания о мире и его объектах [2]. Для выполнения этого шага используются инструменты обработки естественного языка, такие как ConceptNet и WordNet.

Шаг 4. Исполнение запроса. На данном этапе выполняется поиск соответствующих объектов в базе знаний и возвращается результат пользователю.

Примеры использования данного подхода демонстрируют его широкие возможности и гибкость в различных сферах. Одной из них является электронная коммерция.

В рамках данной сферы метод может быть использован для проверки запросов на естественном языке, которые позволяют потенциальным клиентам находить необходимые товары и услуги с большей эффективностью.

Например, пользователь может задать запрос на естественном языке, такой как "найти черные кроссовки размером 42", и система сможет автоматически обработать этот запрос, произвести поиск в базе данных и вывести пользователю результаты, соответствующие его запросу.

Другим направлением использования данного метода являются чат-боты. Они могут помочь пользователям быстро и удобно получать необходимую информацию.

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

Например, чат-бот для покупок может помочь пользователю найти товар, основываясь на запросе на естественном языке, таком как "хочу купить куртку", и предложить пользователю соответствующие варианты.

В целом предложенный формальный аппарат имеет широкий потенциал применения в различных областях, связанных с взаимодействием человека с компьютерной системой.

Список использованных источников:

1. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., & Polosukhin, I. *Attention is All You Need* // *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA: Curran Associates, 2017. P. 112-113.
2. Hirschberg, J. & Manning, C.D. *Advances in Natural Language Processing* // *Science*. 2015. Vol. 349, Number 6245. P. 261-266.

МНОГОФАКТОРНЫЙ АНАЛИЗ ДАННЫХ ДЛЯ СИСТЕМЫ РАННЕГО ОБНАРУЖЕНИЯ ЛЕСНЫХ ПОЖАРОВ

Дронов Д.Д.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Перцев Д.Ю. – канд. тех. наук, доцент каф. ЭВМ

Многофакторный анализ данных — это статистический метод, используемый для анализа взаимодействия множества факторов на одну или несколько зависимых переменных. Система раннего обнаружения лесных пожаров собирает данные из многих источников для оценки риска возникновения пожара на определённой территории. Используя в качестве факторов температуру, концентрацию углекислого газа, водорода и углеводородных газов, осадки, направление и скорость ветра, с помощью многофакторного анализа возможно найти закономерности и предсказывать условия с наивысшим риском возникновения пожара. Эта информация отображается на карте с помощью геоинформационной системы и позволяет сохранить природу и жизни людей.

Лесные пожары — это серьезная угроза для окружающей среды и человеческой жизни. Они могут привести к уничтожению животных и растительных видов, земельных угодий и инфраструктуры. Необходимо принимать меры для предотвращения пожаров и контроля их распространения, а также обеспечивать эффективное и быстрое реагирование на возможные пожары для минимизации их негативных последствий [1].

Система раннего обнаружения лесных пожаров представляет собой сеть устройств, объединённых в сеть ячеистой топологии [2] по протоколам IoT. Каждое устройство содержит датчик температуры, углекислого газа, водорода и углеводородных газов [3][4].

Входные данные включают дату и время, идентификатор устройства, геопозицию и показания с датчиков (пример данных представлен в таблице 1). Дополнительные факторы могут включать информацию о количестве осадков, силе и направлении ветра от метеорологических служб. Все эти данные обрабатываются с помощью многофакторного анализа данных.

Таблица 1 - Пример данных, получаемых от устройств

№	Идентификатор устройства	Дата	Время	GPS-широта	GPS-долгота	Температура	CO2	H2	CxHx
1	1	05.06.2022	15:00	29	52	28	0,0143	0,2906	0,9413
2	6	05.06.2022	15:01	34	47	28	0,0075	0,2603	1,4221
3	9	05.06.2022	15:02	37	44	28	0,0074	0,2139	0,9248
4	4	05.06.2022	15:03	32	49	28	0,0076	0,2579	1,9364
5	8	05.06.2022	15:04	36	45	28	0,0128	0,2607	0,1835
6	2	05.06.2022	15:05	30	51	28	0,0164	0,2237	0,0473
7	7	05.06.2022	15:06	35	46	28	0,0092	0,2095	0,3126
8	5	05.06.2022	15:07	33	48	28	0,0010	0,2297	0,6582
9	7	05.06.2022	15:08	35	46	28	0,0180	0,2136	1,8317
10	3	05.06.2022	15:09	31	50	28	0,0051	0,2329	1,4122

Для данной системы анализ данных начинается с предварительной обработки входных данных. Входные данные очищаются от выбросов, пропущенных значений и ошибок измерения. Далее производится нормализация данных, что позволяет сравнить различные переменные на основе их значений.

После этого производится построение модели многофакторного анализа данных. Данная модель включает в себя факторы, которые могут оказывать влияние на исходную переменную - наличие пожара в лесу.

Одним из методов многофакторного анализа данных является метод главных компонент (РСА). Он позволяет уменьшить размерность данных, выявить скрытые зависимости между факторами и сократить количество используемых переменных.

Вторым методом анализа данных, который может быть применен для системы раннего обнаружения лесных пожаров, является регрессионный анализ. Регрессионный анализ позволяет определить связь между зависимой переменной и набором независимых переменных, и использовать эту связь для прогнозирования значений зависимой переменной на основе значений независимых переменных.

В случае системы раннего обнаружения лесных пожаров, зависимой переменной может быть вероятность возникновения пожара, а независимыми переменными - температура, содержание углекислого газа, концентрация водорода и других газов, а также осадки, направление и скорость ветра. Для построения регрессионной модели необходимо использовать исторические данные о значениях этих переменных в периоды, когда происходили пожары.

Для визуализации собранных и обработанных данных используется геоинформационная система [5]. В зависимости от настройки параметров, возможно отображение датчиков и информации с них, либо раскрашивание карты в соответствии с риском возникновения пожара. Для указания области возникновения пожара используется несколько методов, изображенных на рисунке 1.

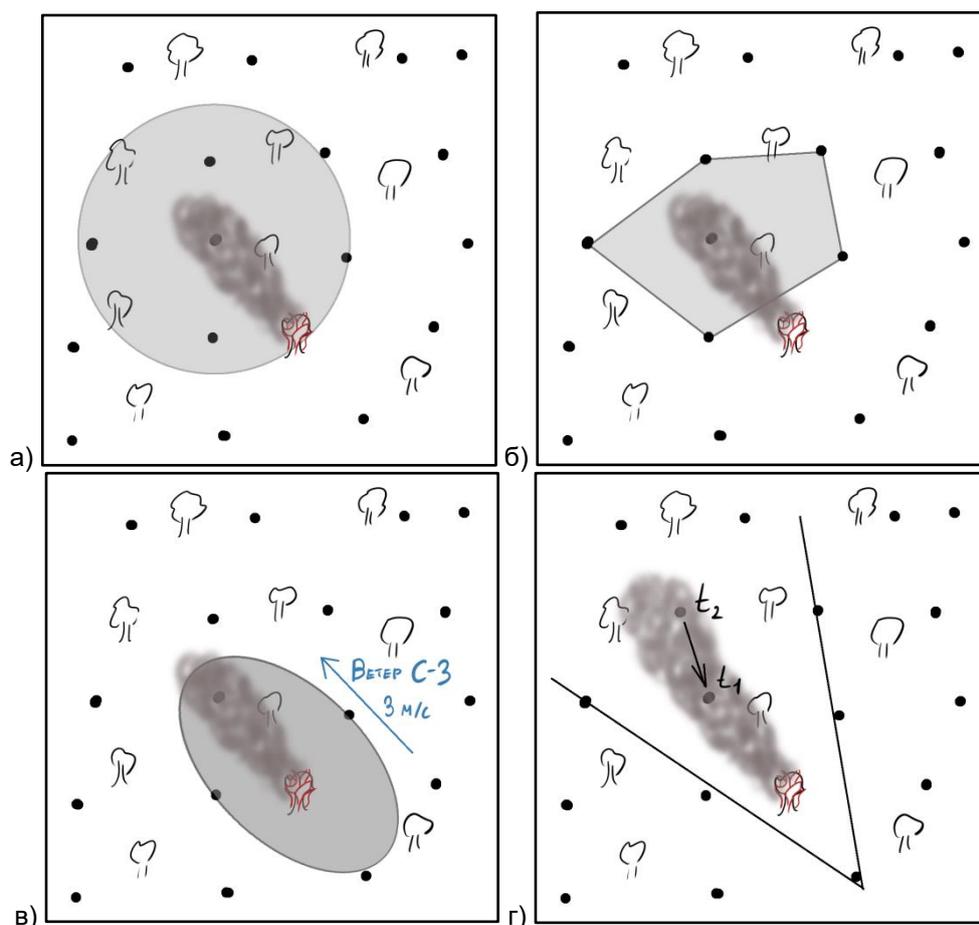


Рисунок 1 - Определение области возникновения пожара (а - одно устройство, б - полигональное, в - с учётом метеоданных, г - на основе нескольких датчиков)

С помощью авторегрессионной модели ARIMAX [6] возможно построить модель риска возникновения лесного пожара, которая учитывает сезонные изменения исследуемых факторов.

После построения модели можно использовать ее для предсказания вероятности возникновения пожара на основе текущих значений переменных. Для этого необходимо ввести значения независимых переменных в модель и получить прогноз для зависимой переменной. Если прогнозируемая вероятность возникновения пожара выше заданного порогового значения, система может автоматически сигнализировать об этом и предпринимать необходимые меры.

Список использованных источников:

1. Пожары меняют климат, даже когда мы об этом забываем [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://greenpeace.ru/blogs/2019/09/23/pozhary-menjajut-klimat-dazhe-kogda-my-ob-jetom-zabyvaem/> Дата доступа: 15.03.2023
2. Wireless Mesh Networks in IoT Networks / Yu Liu, Xiangdong Qiu [et al.] // 2017 International Workshop on Electromagnetics: Applications and Student Innovation Competition, 2017. – С.55-57.

Конференция «Компьютерные системы и сети»

3. *Wildland fire emissions, carbon, and climate: Emission factors / Shawn Urbanski // Forest Ecology and Management, 2014. – С.51-60.*
4. *Early forest fire detection and verification using optical smoke, gas and microwave sensors / Wolfgang Krüll [et al.] // Procedia Engineering, 2012. – С.584-594.*
5. *Зиновьев, А.Ю. Визуализация многомерных данных / А.Ю. Зиновьев. – М. : Изд-во КГТУ, 2000. – 180 с.*
6. *Тихонов, Э.Е. Методы прогнозирования в условиях рынка: учебное пособие / Э.Е. Тихонов – Невинномысск, 2006. – 221 с.*

МОДЕЛИ УГРОЗ ЛОКАЛЬНОЙ СЕТИ В УЧРЕЖДЕНИИ ОБРАЗОВАНИЯ

Силич С.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шевчук О.Г. – канд. тех. наук, доцент

Рассмотрены виды угроз, возникшие в компьютерных локальных сетях, их отличительные особенности. Предложены общие рекомендации по применению как программных, так и аппаратных способов устранения угроз.

Современные информационные системы имеют сложную структуру. Они содержат пользовательские приложения, работающие во взаимодействии с различными операционными системами, установленными на компьютерах, объединенных в локальную сеть, часто связанную тем или иным образом с сегментом глобальной сети. Обеспечение безопасности требует проведения целого комплекса мероприятий в соответствии с разработанной на предприятии политикой информационной безопасности [1].

Существует два возможных направления политики информационной безопасности:

1. Ограничительная политика, пользователь имеет право использовать любые ресурсы, кроме тех, доступ к которым ограничен или закрыт.

2. Нормативная политика, пользователь имеет право использовать только те ресурсы, которые ему явным образом выделены.

Информационная безопасность обеспечена в случае, если для любых информационных ресурсов в системе поддерживается определенный уровень конфиденциальности (невозможности несанкционированного получения какой-либо информации), целостности (невозможности несанкционированной либо случайной ее модификации) и доступности (возможности за разумное время получить требуемую информацию). При этом учитывается не только вероятность нарушения какого-либо из аспектов безопасности в результате умышленных либо неумышленных действий пользователей, но и вероятность выхода из строя каких-либо узлов информационной системы [2].

Многоуровневая защита информационной сети:

1. Внешний уровень определяет взаимодействие информационной системы организации с глобальными ресурсами и системами других организаций.

2. Сетевой уровень связан с доступом к информационным ресурсам внутри локальной сети организации. Безопасность информации на этом уровне обеспечивается средствами проверки подлинности пользователей и разграничением доступа к ресурсам локальной сети (аутентификация и авторизация).

3. Системный уровень связан, прежде всего, с управлением доступом к ресурсам ОС. На этом уровне происходит непосредственное взаимодействие с пользователями, запускаются приложения, и определяются «правила игры» между информационной системой и пользователем (задается либо изменяется конфигурация системы).

4. Уровень приложений связан с использованием прикладных ресурсов информационной системы. Поскольку именно приложения на содержательном уровне работают с пользовательскими данными, для них, нужны собственные механизмы обеспечения информационной безопасности.

При организации защиты от интернет-угроз важным является понятие периметра – укрепленной границы сети. Периметр может состоять из различных подсистем, как представленных различными аппаратными и программными средствами, так и объединенными в единый программно-аппаратный комплекс. Такими подсистемами обычно являются:

- маршрутизаторы (routers);
- межсетевые экраны (брандмауэры, firewalls);
- прокси-серверы;
- системы обнаружения вторжений (IDS);
- средства создания виртуальных частных сетей (VPN);
- антивирусные средства;
- экранированные подсети

Таким образом, основным типом угроз является атака. Для её минимизации рекомендуется: применять прокси-сервера как единой точки выхода в сеть Интернета; на сервере с прокси сервером установку межсетевого экрана (программного или аппаратного); доступ к ресурсам сети через логин и пароль с ограничением на число попыток.

Список использованных источников:

1. Технологии Репликации [Электронный ресурс] – Режим доступа: <https://techrepl.ru/uslugi/ddos.html>

ПРАКТИКА ПРИМЕНЕНИЯ MICROSOFT KINECT

Мельник Н. С.

Белорусский государственный университет информатики и радиоэлектроники
Минск, Республика Беларусь

Перцев Д.Ю., к.т.н., доцент

В данном материале описаны принципы работы сенсора Microsoft Kinect, рассмотрен опыт разработки и реализации проектов с использованием сенсора в различных сферах человеческой деятельности.

Microsoft Kinect был представлен в 2010 году компанией Microsoft. Устройство было разработано в качестве нового способа взаимодействия пользователя с игровой консолью и привело к революции в игровой индустрии. Для распознавания движений игрока Kinect не нужны дополнительные датчики на самом человеке, что стало конкурентным преимуществом по сравнению с альтернативными разработками.

Последующие поколения устройств привели к разделению линейки на Kinect для ОС Windows и Kinect для Xbox. Предоставленные разработчикам инструменты позволили реализовать новые программные продукты для применения в различных сферах человеческой жизнедеятельности. Хорошими примерами могут послужить проекты по организации занятий в виртуальном пространстве [1] или применения в учебном процессе физической культуры в университете [2].

Строение и принципы работы Microsoft Kinect

Microsoft Kinect является системой, состоящей из следующих компонентов: видеокамеры, инфракрасного излучателя и приемника, ряда микрофонов, а также ножки для настройки наклона [3].

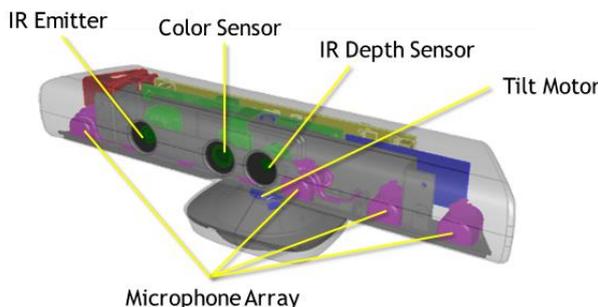


Рисунок 1 – Устройство Microsoft Kinect v1.0

Инфракрасный излучатель применяется для формирования инфракрасных лучей, которые отражаются от предметов и попадают назад в приемник, где одновременно с приемом отраженных лучей формируется матрица расстояний. Таким образом, выстраивается кадр с максимальным размером 640x480 (30 fps). Для поддержки функционала различных видеочатов и дополненной реальности Kinect имеет встроенную видеокамеру, которая поддерживает захват видео с разрешением 1280x960 (12 fps), углом обзора 43° по вертикали и 57° по горизонтали. Есть возможность выбора цветовой модели RGB или YUV.

Следует отметить, что в отличие от своих конкурентов, Kinect поддерживает управление голосовыми командами. Для этого устройство имеет 4 встроенных микрофона, с возможностью определения месторасположения источника звука.

Важным аспектом при разработке программного обеспечения с использованием Kinect является выбор диапазона расстояний. У устройства есть два основных режима: режим близких диапазонов и режим дальних расстояний. Схема расстояний для устройства представлена на рисунке 2.

Необходимым требованием для корректной работы Kinect является определенное световое окружение. Сам производитель рекомендует убрать или приглушить яркие источники света, направленные на игрока. Это связано с использованием инфракрасного излучения. Данная особенность ограничивает варианты реализации многих сценариев, предусматривающих использование устройства вне помещений.

Как упоминалось ранее, существуют две линейки устройств. Главными отличиями являются поддерживаемые диапазоны расстояний и устройства, с которыми работает Kinect. Kinect для Xbox тестировался и разрабатывался именно для этой консоли, что может ограничить реализацию возможного функционала при использовании устройства с системой Windows.

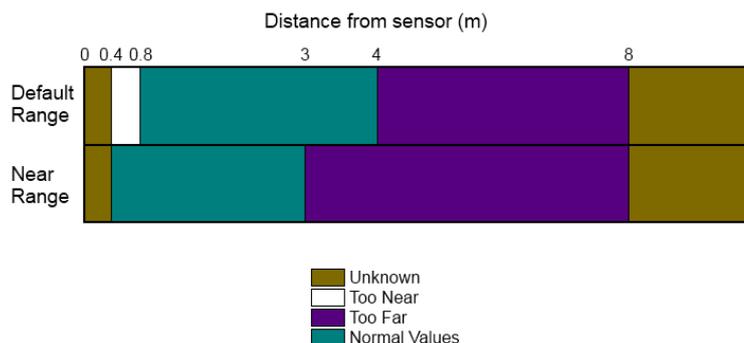


Рисунок 2 – Схема диапазонов Kinect

Для разработки программных продуктов производителем разработан бесплатный Kinect SDK и студия разработки Kinect. Сочетание данных инструментов позволяет разработчикам использовать возможности отслеживания положения человеческого тела, трекинга лица и распознавание речи.

Применение Kinect в ВУЗах

На базе Поволжского государственного технического университета разработана виртуальная платформа проведения занятий vAcademia [1]. Основной задачей был перенос движения преподавателя в виртуальный мир посредством повтора движений аватаром. В ходе разработки платформы авторы представили новые методы обработки неестественных движений и положений тела. Таким образом, был представлен новый подход к организации учебного процесса.

Еще одним способом применения Kinect может служить организация занятий по физической культуре [2]. Студентам из специальной медицинской группы по физкультуре была предоставлена возможность выполнять упражнения в играх на консоли Xbox. Проведенный эксперимент помог привлечь студентов к занятиям физкультурой, снизить их психологический барьер, связанный с физическими ограничениями, а также проявить творческие способности студентов.

Другие сценарии использования Kinect

Интересным для рассмотрения является проект «Who am I?», позволяющий собрать статистические данные о посетителях выставок [4]. Пользователям предлагается ответить на несколько вопросов о возрасте, национальности и половой принадлежности. Для ввода ответов пользователь использует различные жесты и движения, обрабатываемые при помощи Kinect.

На многих форумах можно найти различные обсуждения о применении Kinect в качестве стереокамеры для построения карт глубины. Данный способ применения обрел особую популярность в робототехнике. Сенсоры позволяют реализовать такой функционал роботов, как автоматический объезд препятствий, следование за человеком или распознавание лиц [5].

Вывод

За годы существования устройства на рынке были разработаны разнообразные решения с использованием Kinect для различных сфер деятельности человека таких, как образование, организация выставок и робототехника. Несмотря на развитие анализа видеопотока с помощью искусственного интеллекта, проекты с использованием Kinect до сих пор остаются актуальными. Многие разработчики в настоящее время используют сочетание ИИ и Kinect SDK для создания ПО, что открывает дополнительные возможности реализации новых идей. Большой набор инструментов и дешевизна сенсора способствуют расширению круга разработчиков ПО для использования с Microsoft Kinect.

Список использованных источников:

1. Морозов М. Н.; Сморгалов А. Ю.; Богданов И. О. Использование Microsoft Kinect и вычислительных возможностей потоковых процессов для проведения занятий в виртуальном мире vAcademia – Образовательные технологии и общество. 2013 г.
2. Витун В.Г.; Витун Е.В.; Киязова Е.А. Использование Smart-физкультуры в учебном процессе студентов специальной медицинской группы – Известия ТулГУ. 2018 г. С 8-13.
3. Хабр [Электронный ресурс]. – Kinect for Windows SDK. – Режим доступа: <https://habr.com/ru/articles/150955/>
4. Tobias Beducker Who am I?: Full-body interaction [Электронный ресурс]. - Режим доступа: <http://michaelkipp.de/interaction/projects/content/9.201718w/12.WhoAml/report.pdf>
5. Hackster.io [Электронный ресурс]. – Human-Following Robot with Kinect. – Режим доступа: <https://www.hackster.io/turtle-rover/human-following-robot-with-kinect-efb3cd>

СИСТЕМА УПРАВЛЕНИЯ МНОГОЗАДАЧНЫМ РОБОТОМ

Шишков Ю. Ю.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Стракович А. И. – ассистент кафедры ЭВМ

Показан подход к созданию системы управления многозадачным роботом. Рассмотрены различные особенности и варианты разработки, а также процесс создания системы управления на платформе ROS поэтапно.

Система управления многозадачным роботом является одним из наиболее важных аспектов в робототехнике. Она отвечает за координацию и контроль всех функций робота, включая механику, электронику, программное обеспечение, сенсоры и другие компоненты.

Одна из наиболее распространенных систем управления многозадачным роботом — это Robot Operating System (ROS, операционная система для роботов), которая предоставляет набор инструментов и библиотек, которые позволяют программистам и инженерам управлять и координировать работу роботов с помощью высокоуровневых команд, а также позволяет решать сложные задачи в робототехнике, такие как навигация, обработка изображений и данных, управление манипуляторами и многие другие.

Данная система имеет графовую архитектуру, где обработка данных происходит в узлах (nodes), которые могут принимать и передавать сообщения (messages) между собой, что позволяет разработчикам создавать гибкие и модульные системы управления роботами. ROS состоит из двух частей. Первая, это ядро — `roscore`, которое отвечает за работу системы и взаимодействие всех пакетов. Вторая часть — это пользовательские пакеты (packages) или наборы этих пакетов, организованные в стек.

Создание робота на ROS можно разделить на несколько этапов, первый из которых это — выбор аппаратной платформы. Данная система поддерживает взаимодействие с множеством аппаратных платформ, таких как Raspberry Pi, Arduino, NVIDIA Jetson и другие. Выбор платформы зависит от требований проекта и бюджета. После выбора платформы необходимо установить ROS.

Затем создается рабочее пространство, в котором будут храниться все файлы проекта. В рабочем пространстве создаются пакеты, а в пакетах — узлы. Пакет — это основной блок программного обеспечения в ROS, в котором содержится код для управления роботом, сенсорами, алгоритмами и другими компонентами. Узел — это программа, которая выполняет определенную функцию, например, управление двигателями робота или обработка данных с сенсоров. Для создания узлов можно написать код на любом, поддерживаемом ROS, языке программирования: C++, Python или Java.

Следующим шагом создаются и настраиваются сообщения и сервисы — способы передачи данных между узлами, для чего определяются типы данных, которые будут передаваться между узлами.

После создания всех компонентов проекта необходимо составить файл на языке разметки XML или YAML, который определяет, какие узлы и параметры нужно задействовать для работы робота.

На конечном этапе создания робота необходимо протестировать его работу. Для тестирования можно использовать инструменты ROS, такие как `rostopic`, `rosservice`, `rqt` и другие. После тестирования проводится настройка робота, например, настройка параметров узлов, калибровка сенсоров и других компонентов.

ROS изначально создавалась для использования в управлении сложными роботами, стоимость которых может достигать тысяч долларов. Поэтому прежде, чем затрачивать деньги на дорогостоящие физические компоненты, рекомендуется начать с моделирования и симуляции робота в виртуальной среде, а затем уже переходить к созданию физической модели.

В нашем случае, в связи с тем, что робот был сделан раньше, то используется обратный подход, а именно ROS настраивается под физическую модель. В качестве бортового компьютера, в роботе используется Raspberry Pi 4 Model B с 4 ГБ ОЗУ. Робот оснащен 4 колесами с моторами, драйвером для их управления, инфракрасными датчиками препятствий, микрофоном и динамиками.

Разработка программного обеспечения (ПО) для робота требует мощной визуализации, которая может оказаться очень ресурсоемкой. Например, работа с трехмерными объектами и визуализация на Raspberry Pi могут замедлить разработку ПО. Кроме того, некоторые задачи, которые ставятся перед роботом, такие как навигация в помещении, очень сложные и требуют большой вычислительной мощности. Выполнение этих алгоритмов на Raspberry Pi может быть очень медленным или даже невозможным. При использовании мощного ноутбука вместо Raspberry Pi, можно вести весь проект на одной машине. В данном же случае ROS установлен на две машины

– Raspberry Pi и персональный компьютер (ПК). Система на роботе занимается простыми вычислениями и поддерживает работу драйверов низкого уровня, а настольный компьютер с ROS выполняет сложные вычисления навигации и визуализации.

Поскольку ROS используется на двух компьютерах, необходимо на каждом из них в созданном рабочем пространстве поддерживать одинаковые и актуальные файлы проекта. Хотя такой подход может показаться неудобным, он является необходимым при использовании двух машин вместо одной. Для обеспечения синхронизации рабочих пространств на настольном компьютере и Raspberry Pi можно использовать различные утилиты, такие как git или rsync.

После установки ROS на обе машины можно приступать к написанию низкоуровневых драйверов, но сперва необходимо настроить сеть ROS, в которой может быть запущено только одно ядро roscore. Именно машина с запущенным ядром отвечает за работу всей системы. В сети ROS она называется master, а остальные машины становятся slave. В данном проекте в качестве master выбран настольный компьютер. Для всех ROS-компьютеров в сети нужно указать, какая именно машина является master.

После настройки сети ROS создано рабочее пространство. Затем был написан первый пакет проекта, который отвечает за драйвер моторов, были написаны два узла — talker и listener.

Узел talker запускается только на ПК, он отвечает за передачу сообщений роботу, здесь считываются команды управления с клавиатуры ПК. Этот узел является узлом-публицистом, которая публикует сообщения на шину chatter.

Узел listener запускается на Raspberry Pi, он отвечает за обработку команд пришедших роботу. Данный узел является узлом-подписчиком, который читает сообщения с шины chatter.

Протестировав драйвер моторов, был написан узел ds_talker, который отвечает за управление роботом с помощью геймпада DualSense PS5, который подключается по Bluetooth или USB к настольному компьютеру.

Структура ROS-графа проекта после написания всех узлов представлена на рисунке 1.

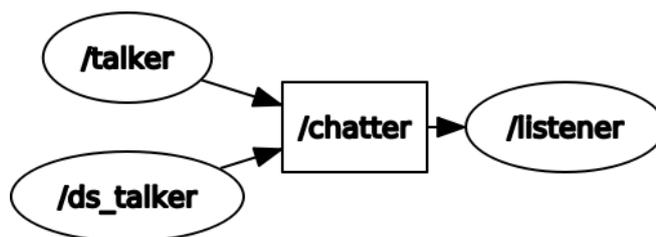


Рисунок 1 – ROS-граф проекта

Собранный робот представлен на рисунке 2. После успешного тестирования написанных узлов управления роботом, можно переходить к дальнейшему развитию проекта, а именно улучшению навигации при помощи датчиков препятствия.

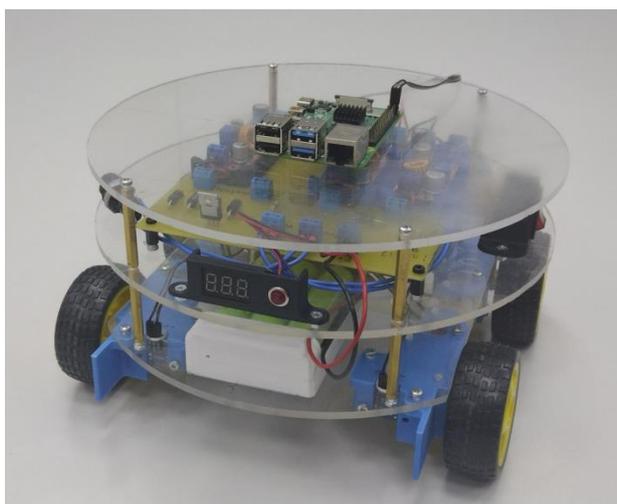


Рисунок 2 – Многозадачный робот

Список использованных источников:

1. ROS [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://wiki.ros.org/> - Дата доступа: 01.04.2023

СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ АЛГОРИТМОВ ПОИСКА И СОРТИРОВОК НА МАССИВАХ

Малюш Д.О., студент гр. 250505, Борисенко К.Н., студент гр. 250501

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сидорович А.С. – старший преподаватель

В научной работе обсуждается выбор оптимальных алгоритмов поиска и сортировки данных в массивах. Это предполагает сравнительный анализ алгоритмов.

Существует множество способов поиска необходимой позиции в массиве и алгоритмов сортировки массива. Для исследования сравнительных характеристик алгоритмов поиска и сортировок будет рассматриваться массив, состоящий из 100 символов.

Алгоритм поиска в массиве

Линейный поиск. Линейный поиск — это простой алгоритм поиска, который работает путем последовательного просмотра каждого элемента в массиве и сравнения его с искомым значением. Если элемент найден, алгоритм возвращает его индекс[5].

Индексно-последовательный поиск. Индексно-последовательный поиск — это алгоритм поиска, который использует индекс для поиска элемента в списке или массиве. Затем алгоритм использует индекс для быстрого нахождения диапазона элементов, которые могут содержать искомое значение. После этого алгоритм выполняет последовательный поиск в этом диапазоне для нахождения искомого элемента. Массив изначально должен быть отсортирован.

Бинарный (двоичный) поиск. Бинарный поиск также известен как логарифмический поиск. Он работает путем деления массива на две части и сравнения искомого значения с элементом в середине[1]. Если искомое значение меньше значения в середине массива, то поиск продолжается в левой половине массива. В другом случае поиск продолжается в правой половине массива. Этот процесс повторяется до тех пор, пока не будет найден искомый элемент или пока не останется ни одного элемента для поиска.

Поиск прыжками. Поиск прыжками - это алгоритм поиска, требует предварительной сортировки набора данных[2].

Алгоритм работает путем прыжков через фиксированное количество элементов в массиве и сравнения каждого прыжка с элементом, который мы должны найти. Если текущий прыжок больше искомого элемента, то поиск продолжается в предыдущем интервале прыжка с помощью линейного поиска.

Сравнение скорости работы перечисленных алгоритмов поиска представлено на рисунке 1.

Сравнение памяти, необходимой для работы перечисленных алгоритмов поиска представлено на рисунке 2.

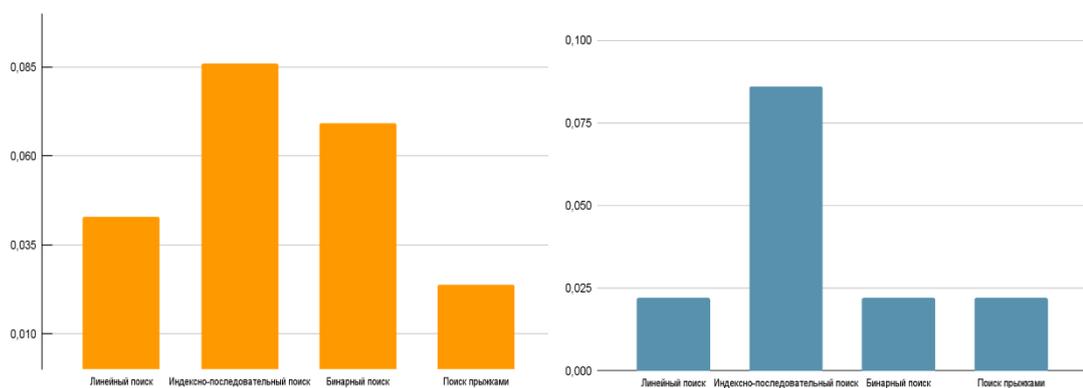


Рисунок 1 – Диаграмма, отображающая различие в скорости работы рассматриваемых алгоритмов поиска

Рисунок 2 – Диаграмма, отображающая различие в количестве памяти, требуемой для работы рассматриваемых алгоритмов поиска

Сортировка массива

Сортировка пузырьком. Сортировка пузырьком — это простой алгоритм сортировки. Он работает путем многократного сравнения и обмена соседних элементов, если они находятся в неправильном порядке[3]. Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется перестановка элементов.

Быстрая сортировка (сортировка Хоара): Быстрая сортировка — это алгоритм сортировки. Состоит из трех шагов: выбор опорного элемента из массива, разбиение массива таким образом, чтобы элементы меньше опорного помещались перед ним, а большие или равные - после, и рекурсивное повторение этих действий для левой и правой части массива[3].

Сортировка методом вставок. Сортировка вставками — это алгоритм сортировки, который работает путем перемещения текущего элемента входной последовательности на нужную позицию в уже отсортированном массиве. Отсортированный массив строится по одному элементу за раз[4]. Внутри функции используется цикл, который начинается с первого значения массива и продолжается до конца массива. На каждой итерации цикла выбирается новый элемент из массива и сравнивается с элементами в уже отсортированном списке. Если порядок между сравниваемыми элементами нарушен, то меняем их местами.

Сортировка методом Шелла. Сортировка Шелла — алгоритм сортировки, являющийся усовершенствованным вариантом сортировки вставками. Идея метода Шелла состоит в сравнении элементов, стоящих не только рядом, но и на определенном расстоянии друг от друга[3].

При сортировке Шелла сначала сравниваются и сортируются между собой значения, стоящие один от другого на некотором расстоянии. После этого процедура повторяется для некоторых меньших значений.

Сравнение скорости работы перечисленных алгоритмов сортировки представлено на рисунке 3.

Сравнение памяти, необходимой для работы перечисленных алгоритмов сортировки представлено на рисунке 4.

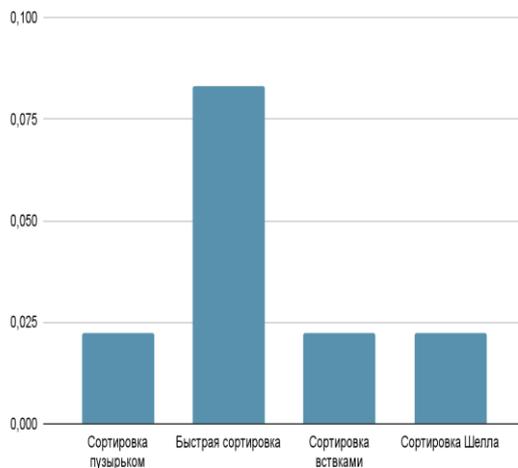


Рисунок 3 – Диаграмма, отображающая различие в скорости работы рассматриваемых алгоритмов сортировок.

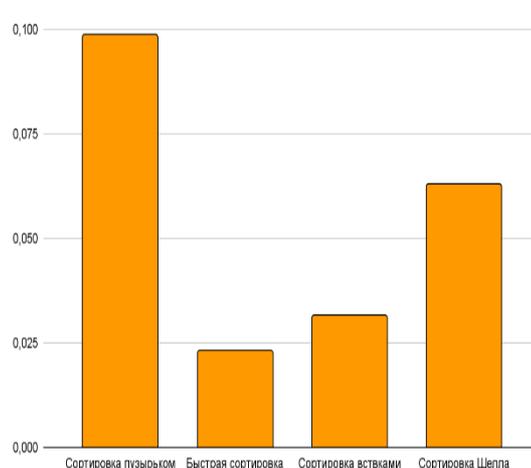


Рисунок 4 – Диаграмма, отображающая различие в количестве памяти, требуемой для работы рассматриваемых алгоритмов сортировки

Каждый способ сортировки и алгоритм поиска будет уместен в той или иной ситуации. Для максимально быстрой сортировки лучше использовать сортировку Хоара, ведь этот алгоритм имеет наибольшую скорость, а самым эффективным алгоритмом поиска по всем параметрам оказался именно алгоритм поиска прыжками. Зная данную информацию, можно выбирать наиболее подходящие алгоритмы поиска или сортировки, для того чтобы оптимизировать и ускорить работу программы.

Список использованных источников:

1. Левитин А. В. Глава 4. Метод декомпозиции: Бинарный поиск // *Алгоритмы. Введение в разработку и анализ* — М.: Вильямс, 2006. — С. 180—183. — 576 с.
2. Скиена С. *Алгоритмы. Руководство по разработке*. 2-е изд.: Пер. с англ. — СПб.: БХВ-Петербург. 2011. — 720 с.: ил.
3. Липачёв Е. К. // *Технология программирования. Методы сортировки данных // учебное пособие / Е.К. Липачёв. – Казань: Казан. ун-т, 2017. – 58 с.*
4. Кнут Д. Э. 5.2 Внутренняя сортировка // *Искусство программирования. Том 3. Сортировка и поиск = The Art of Computer Programming. Volume 3. Sorting and Searching / под ред. В. Т. Тертышного (гл. 5) и И. В. Красикова (гл. 6). — 2-е изд. — Москва: Вильямс, 2007. — Т. 3. — 832 с..*
5. Кнут Д. *Искусство программирования, том 3. Сортировка и поиск = The Art of Computer Programming, vol.3. Sorting and Searching. — 2-е изд. — М.: «Вильямс», 2007. — С. 824*

УСТРОЙСТВО АВТОМАТИЗИРОВАННОГО РОЗЛИВА НАПИТКОВ

Самаль И.Д. Хатеневич А.А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Стракович А.И. – ассистент каф. ЭВМ

Приведены основные технические решения и особенности создания устройства автоматизированного розлива напитков.

Устройство автоматизированного розлива напитков или автоматический бармен – устройство, предназначенное для приготовления напитков с минимальным вмешательством человека. Для создания напитка человеку необходимо выбрать его из предложенного списка, после чего автоматический бармен нальёт необходимые ингредиенты, согласно объёму, указанному в рецепте.

Основной частью устройства является платформа, предназначенная для передвижения стакана. Вдоль корпуса установлена направляющая, по которой при помощи шагового двигателя 17HS4401 и ременной передачи перемещается платформа. Над самой платформой, по линии ее движения размещены дозаторы и помпы, предназначенные для наполнения стакана нужными ингредиентами. Помпы производят наполнение путем перекачки жидкости из бутылки в стакан, а дозатор выдает фиксированный объем напитка при нажатии на него. Дозаторы размещаются на расстоянии 10 см друг от друга, а в промежутки между ними выведены силиконовые трубки от помп, которые в свою очередь монтируются в задней части корпуса.

Воздействие на дозаторы можно осуществлять следующими способами: при помощи линейного привода, расположенного на самой платформе и осуществляющего нажатие на необходимый дозатор, или при помощи сервоприводов, каждый из которых закреплён за отдельным дозатором и осуществляет нажатие на него. Так как использование выдвигного механизма утяжеляет платформу и во время нажатия создает дополнительную нагрузку на неё, то в данном проекте использован второй вариант.

Обработку данных и управление модулями осуществляет плата ESP32[1]. Для управления сервоприводами используется широтно-импульсная модуляция или ШИМ сигнал[2], который можно подавать через выходы ESP32, но, с целью упрощения схемы и снижения занятости выходов платы, в устройстве присутствует расширитель портов ШИМ PCA9685. Данный расширитель управляется по интерфейсу I2C и предоставляет 16 портов вывода ШИМ сигнала, каждый из которых настраивается по отдельности.

Для шагового двигателя необходимо отдельное питание 12 В. Используемый двигатель представляет собой классический двухфазный шаговый электродвигатель, в котором для вращения ротора последовательно активируются обмотки статора. Для управления им необходим специальный драйвер A4988, который способен передать напряжение с блока питания на обмотки статора двигателя в правильном порядке. Сам драйвер питается напряжением 5 В и управляется двумя цифровыми сигналами с платы.

Для управления питанием помп, которым для работы также необходимы 12 В, можно использовать полевые транзисторы или реле. Данные элементы имеют одну цель: передать напряжение с блока питания на контакты помпы строго по сигналу с платы. Воздействовать на эти элементы можно при помощи цифрового сигнала. В проекте уже используется расширитель портов ШИМ, поэтому его можно использовать для управления помпами, для чего нужно выдавать сигнал со скважностью равной 100%.

Для отображения доступных рецептов, состояния устройства и другой информации используется LCD дисплей. Сами рецепты сохранены в файл на SD карте. Такой подход обеспечивает не только хранение большого количества рецептов, но и их простое редактирование через подключение SD карты к компьютеру. Соответствие ингредиента с определённой помпой или дозатором хранится в файле конфигураций, который также сохранён на SD карте. Устройство имеет два режима работы: приготовления и редактирования, выбор которых происходит через меню посредством кнопок. При приготовлении напитка обращение к файлу конфигураций происходит для подтверждения наличия всех необходимых ингредиентов и получения информации о номерах подсоединённых к ним помп и дозаторов. В процессе приготовления напитка объём ингредиента уменьшается, что приводит к изменению данных в файле. В режиме редактирования пользователь осуществляет замену ёмкостей с напитками в держателях, после чего изменения сохраняются в файл.

Список использованных источников:

1. ESP32 Wi-Fi & Bluetooth MCU [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.espressif.com/en/products/socs/esp32>

2. Широтно-импульсная модуляция [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://prosvetodiod.ru/informatsiya-ob-osveshhenii/cto-takoe-shim-shirotno-impulsnaya-modulyatsiya>

ОЦЕНКА АЛЬТЕРНАТИВ ПО СОВОКУПНОСТИ ПОКАЗАТЕЛЕЙ В УСЛОВИЯХ РЕАЛИЗАЦИИ ПРОЕКТА МЕТОДОМ АНАЛИЗА ИЕРАРХИЙ

Русина Н.В., Русин Д.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Аннотация. В работе представлен метод принятия решений, который можно использовать для анализа сложных проблем – метод анализа иерархий (метод Саати), метод предполагает разбиение задачи на более мелкие составные части и попарное сравнение их друг с другом для определения их относительной важности.

Ключевые слова: системный анализ, матрица парных сравнений, принятие решений.

Целью задачи является аренда офиса для IT-компании. Необходимо выбрать лучшее предложение с точки зрения выделенных критериев. Собственники компании предварительно отобрали три возможных альтернативных варианта: A_1 , A_2 , A_3 . Было выявлено три основных критерия, влияющих на принятие решения о выборе: логистическое удобство (K_1), стоимость аренды (K_2), наличие развитой инфраструктуры (K_3).

Решение начинается с построения иерархического представления задачи, включающего все элементы, учитываемые при ее решении (в данном случае – альтернативы и показатели). На первом уровне в иерархическом представлении задач всегда указывается один элемент – выбор (цель), на втором уровне – показатели, по которым делается выбор, на третьем – альтернативы. Выполняется попарное сравнение всех элементов задачи. Сравнение состоит в указании экспертных оценок превосходства (или, наоборот, отставания) элементов задачи относительно друг друга. Сначала сравниваются показатели по их важности. Затем сравниваются альтернативы по каждому показателю. Для этого заполняются матрицы парных сравнений, затем они обрабатываются и проверяются на непротиворечивость по правилам метода Саати.

1. Реализация метода в условиях конкретной задачи

Сначала сравним показатели по важности. Матрица заполняется в соответствии с мнением экспертов о важности показателей. Исходные данные задачи отображены в Таблице 1, Таблице 2, Таблице 3, Таблице 4.

Матрица парных сравнений показателей по важности:

Таблица 1

$i \backslash j$	K_1	K_2	K_3
K_1	1	$\frac{1}{2}$	7
K_2	2	1	8
K_3	$\frac{1}{7}$	$\frac{1}{8}$	1

Средние геометрические строк матрицы: $C_1 = 1,52$; $C_2 = 2,52$; $C_3 = 0,26$. Сумма средних геометрических $C = 4,3$. Определим локальные приоритеты (в нашем случае - оценки важности показателей): $L'_1 = \frac{C_1}{c} = \frac{1,52}{4,3} = 0,35$; $L'_2 = \frac{C_2}{c} = \frac{2,52}{4,3} = 0,59$; $L'_3 = \frac{C_3}{c} = \frac{0,26}{4,3} = 0,06$.

Чем больше локальный приоритет, тем важнее показатель. В нашем случае показатель «стоимость аренды» носит приоритетный характер.

На основании матриц парных сравнений вычисляются оценки важности показателей, оценки предпочтительности альтернатив по каждому из показателей, обобщенные оценки предпочтительности альтернатив. Матрица парных сравнений (таблица 2) альтернатив по первому критерию (логистическое удобство), матрица парных сравнений (таблица 3) альтернатив по второму критерию (стоимость аренды), матрица парных сравнений альтернатив по третьему критерию (наличие развитой инфраструктуры):

Таблица 2

Таблица 3

Таблица 4

i \ j	A ₁	A ₂	A ₃		i, j	A ₁	A ₂	A ₃		i, j	A ₁	A ₂	A ₃
A ₁	1	7	3		A ₁	1	$\frac{1}{7}$	$\frac{1}{5}$		A ₁	1	5	5
A ₂	$\frac{1}{7}$	1	3		A ₂	7	1	3		A ₂	$\frac{1}{5}$	1	$\frac{1}{5}$
A ₃	$\frac{1}{3}$	$\frac{1}{3}$	1		A ₃	5	$\frac{1}{3}$	1		A ₃	$\frac{1}{5}$	5	1

2. Обработка матрицы парных сравнений

2.1. Логистическое удобство K₁ (таблица 2):

Средние геометрические строк матрицы (цена альтернатив): C₁ = 2,76; C₂ = 0,75; C₃ = 0,48.

Сумма цен альтернатив: C = 2,76 + 0,75 + 0,48 = 3,99.

Локальные приоритеты (в нашем случае – оценка важностей показателей или веса альтернатив):

$$L_1 = \frac{c_1}{c} = \frac{2,76}{3,99} = 0,69; L_2 = \frac{c_2}{c} = \frac{0,75}{3,99} = 0,19; L_3 = \frac{c_3}{c} = \frac{0,48}{3,99} = 0,12.$$

Предпочтительней является альтернатива с максимальным весом (чем больше локальный приоритет, тем важнее показатель). В нашем случае наиболее удобным является расположение офиса A₁ (по показателю K₁), затем A₂ и A₃.

Таблица 5

i \ j	A ₁	A ₂	A ₃	C _i	L _i
A ₁	1	7	3	2,76	0,69
A ₂	$\frac{1}{7}$	1	3	0,75	0,19
A ₃	$\frac{1}{3}$	$\frac{1}{3}$	1	0,48	0,12
Сумма	1,48	8,33	7	3,99	1

Определим λ_{max} :

- суммируем каждый столбец матрицы парных сравнений;
- сумму первого столбца умножаем на первую компоненту локального вектора приоритетов, сумму второго столбца на вторую компоненту и т. д.;
- полученные произведения суммируем.

$$\lambda_{max} = 1,48 * 0,69 + 8,33 * 0,19 + 7 * 0,12 = 1,0212 + 1,5827 + 0,84 = 3,4439$$

Далее найдём ИС (индекс согласованности):

$$ИС = \frac{\lambda_{max} - n}{n - 1} \quad (1);$$

$$ИС = \frac{3,4439 - 3}{3 - 1} = \frac{0,4439}{2} = 0,22195$$

Итак, ОС (отношение согласованности):

$$ОС = \frac{ИС}{СС} \quad (2);$$

В нашем случае $СС = 0,58$, согласно размеру матрицы, $ОС = \frac{0,22195}{0,58} = 0,382$. Если отношение согласованности превышает 0,1, то требуется уточнение матрицы парных сравнений. В нашем случае $ОС = 0,382$, это означает, что суждения эксперта, на основе которых построена матрица, рассогласованы и, как правило, в таком случае эксперту предлагают заполнить матрицу снова и более внимательно использовать шкалу парных сравнений.

2.2. Стоимость аренды K_2 (таблица 3):

Цена альтернатив: $C_1 = 0,3$; $C_2 = 2,76$; $C_3 = 1,19$. Сумма цен альтернатив $C = 4,25$.

Веса альтернатив: $L_1 = 0,07$; $L_2 = 0,65$; $L_3 = 0,28$.

Наиболее важным является стоимость аренды офиса (по показателю K_2) A_2 , затем A_3 и A_1 .

Таблица 6

i \ j	A_1	A_2	A_3	C_i	L_i
A_1	1	$\frac{1}{7}$	$\frac{1}{5}$	0,3	0,07
A_2	7	1	3	2,76	0,65
A_3	5	$\frac{1}{3}$	1	1,19	0,28
Сумма	13	1,44	4,2	4,25	1

$$\lambda_{max} = 13*0,07 + 1,44*0,65 + 4,2*0,28 = 0,91 + 0,936 + 1,176 = 3,022 \approx 3,02.$$

Найдём ИС (индекс согласованности) по формуле (1): $ИС = 0,01$.

$СС = 0,58$, $ОС = 0,017$ по формуле (2).

В данном случае $ОС < 1$. Это означает, что суждения эксперта принимаются и уточнение матрицы парных сравнений не требуется.

2.3. Наличие развитой инфраструктуры K_3 (таблица 4):

Цена альтернатив: $C_1 = 2,92$; $C_2 = 0,34$; $C_3 = 1$. Сумма цен альтернатив $C = 4,26$.

Оценка важностей показателей: $L_1 = 0,69$; $L_2 = 0,08$; $L_3 = 0,23$.

В этом случае наиболее важным является аренда офиса (по показателю K_3) A_1 , затем A_3 и A_2 .

Таблица 7

i \ j	A_1	A_2	A_3	C_i	L_i
A_1	1	5	5	2,92	0,69
A_2	$\frac{1}{5}$	1	$\frac{1}{5}$	0,34	0,08
A_3	$\frac{1}{5}$	5	1	1	0,23
Сумма	1,4	11	6,2	4,26	1

$$\lambda_{max} = 1,4*0,69 + 11*0,08 + 6,2*0,23 = 0,966+0,88 +1,426 =3,272$$

По формуле (1) ИС = 0,136, по формуле (2) ОС =0,23.

В этом случае суждения эксперта, на основе которых построена матрица, рассогласованы и эксперту предлагают заполнить матрицу снова и более внимательно использовать шкалу парных сравнений.

3. На основании полученных оценок определим глобальный приоритет альтернатив следующим образом:

$$G_1 = L'_1 * L_1 + L'_2 * L_2 + L'_3 * L_3 = 0,36 \text{ (данные } L_1, L_2, L_3 \text{ берём из п. 2.1.)}$$

$$G_2 = L'_1 * L_1 + L'_2 * L_2 + L'_3 * L_3 = 0,4242 \text{ (данные } L_1, L_2, L_3 \text{ берём из п. 2.2.)}$$

$$G_3 = L'_1 * L_1 + L'_2 * L_2 + L'_3 * L_3 = 0,2675 \text{ (данные } L_1, L_2, L_3 \text{ берём из п. 2.3.)}$$

Итак, чем больше приоритет, тем лучше альтернатива. В нашем случае наиболее приемлемой является аренда офиса A_2 .

Заключение

Метод анализа иерархий в решении задач позволяет выявить основные критерии, влияющие на принятие решения. Если для конкретной матрицы окажется, что $ОС > 0,1$, то можно утверждать, что суждения эксперта, на основе которых заполнена исследуемая матрица, не согласованы и ему надлежит заполнить матрицу заново, более внимательно использовать при этом шкалу парных сравнений. В случае, если $ОС < 0,1$, суждения эксперта принимаются.

Список литературы

1. Никольшин Б. В. и др. Системный анализ и принятие решений в проектной и управленческой деятельности. -Минск БГУИР 2021
2. Саати Т. Принятие решений. Метод анализа иерархий. - М.: Радио и связь, 2003.

ASSESSMENT OF ALTERNATIVES ACCORDING TO THE SET OF INDICATORS IN THE CONDITIONS OF THE PROJECT IMPLEMENTATION BY THE METHOD OF HIERARCHY ANALYSIS

N. V. RUSINA, D. V. RUSIN

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Abstract. This paper presents a decision-making method that can be used to analyze complex problems, the Hierarchy Analysis Method (Saaty Method), which involves breaking down a problem into smaller constituent parts and comparing them to each other to determine their relative importance.

Keywords: system analysis, matrix of pairwise comparisons, decision making.

СЕКЦИЯ 2
«ВСТРАИВАЕМЫЕ ВЫЧИСЛИТЕЛЬНЫЕ
СИСТЕМЫ РЕАЛЬНОГО ВРЕМЕНИ»

МЕТЕОСТАНЦИЯ-ЛОГГЕР НА БАЗЕ МИКРОКОНТРОЛЛЕРА

Андросов К.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Порхун М.И. – магистр тех. наук.

В работе представлена метеостанция-логгер на базе микроконтроллера ATmega328. Устройство выполняет сбор и фиксацию на SD-карту информацию о температуре, влажности и давлении воздуха, а также содержании в воздухе углекислого газа. Представлена структура и краткое описание метеостанции. Проведена проверка корректности работы устройства путём макетирования.

Здоровая окружающая среда является решающим фактором здоровья и развития человека. Согласно данным Министерства природных ресурсов и охраны окружающей среды РБ к экологически обусловленным заболеваниям относятся: респираторные инфекции, туберкулез, неонатальные болезни, рак, психические и неврологические расстройства. Для управления природными ресурсами необходимо постоянно осуществлять мониторинг метеоусловий и состояния окружающей среды. Важным элементом решения данной задачи является техническое обеспечения процесса сбора климатических и экологических данных [1]. Для выполнения поставленных задач применяются различные средства, в том числе устройства на базе микроконтроллеров.

Для отслеживания и мониторинга метеоусловий и составления на их основе математических моделей с предсказаниями используются стационарные метеостанции. На данный момент на территории Беларуси функционирует 51 подобное устройство, 14 из которых предоставляют информацию в режиме реального времени [2]. При всех своих преимуществах данные устройства имеют и свои недостатки, к которым можно отнести:

- Энергозависимость;
- Громоздкость;
- Стоимость;
- Необходимость наличия персонала для обслуживания.

Исходя из выше указанных недостатков главной целью работы являлось создание полнофункционального макета метеостанции, способного работать в автономном режиме и фиксировать требуемые параметры без участия обслуживающего персонала.

Устройство разработано на базе микроконтроллера ATmega328 [3]. Ключевыми параметрами при выборе этого микроконтроллера стали его стоимость и распространённость, а также хорошие показатели вычислительных мощностей относительно конкурентов своего ценового сегмента. Метеостанция-логгер производит сбор и фиксацию следующих данных:

- Температура воздуха;
 - Влажность воздуха;
 - Давление воздуха;
 - Содержание углекислого газа в воздухе.
- Структура устройства представлена на рисунке 1.

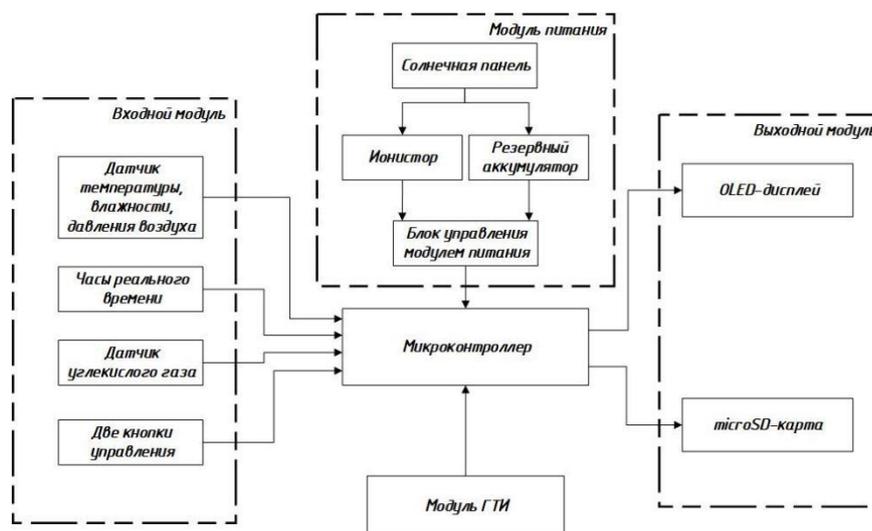


Рисунок 1 – Структурная схема метеостанции-логгера

Сбор данных осуществляется с помощью периферийных датчиков. Для сбора сведений о температуре, давлении и влажности воздуха используется датчик ВМЕ-280 [4]. Связь датчика с микроконтроллером осуществляется через интерфейс I2C. Сбор данных о содержании углекислого газа в воздухе осуществляет датчик МН-219В [5]. Обмен данными с микроконтроллером и управление датчиком осуществляется с помощью интерфейса UART. Помимо датчиков в состав устройства входят часы реального времени (RTC) [4]. С их помощью пользователь может задать текущие дату и время. RTC также предназначены для обеспечения возможности пользователю настроить точный временной интервал для выполнения измерений. Информация с датчиков о состоянии окружающей среды, а также текущая дата и время отображаются на OLED-дисплее SD1306 [6]. Важной особенностью данного дисплея является его относительно низкое энергопотребление при сохранении высокой четкости и контрастности изображения.

Управление метеостанцией-логгером осуществляется с помощью пары кнопочных переключателей «ON», «MODE». При однократном нажатии на клавишу «ON» производится вывод на дисплей текущих параметров с датчиков, а также текущих времени и даты. При одновременном нажатии на клавиши «ON» и «MODE» метеостанция-логгер переходит в режим настройки, в котором при помощи нажатий на клавишу «ON» пользователь может выбрать одну из систем устройства для настройки (например, выставить время и дату или задать период проведения измерений датчиками).

Одной из ключевых функций устройства является фиксация считываемых параметров с датчиков, а также времени и даты на момент произведения замеров. Для обеспечения данного функционала используется модуль с microSD картой [7]. Данные записываются в фиксированном формате, что ускоряет и упрощает обработку полученной информации. В дальнейшем полученный файл с показаниями датчиков может быть обработан, а данные из файла представлены в виде графиков, гистограмм и т.д.

Система питания устройства организована при помощи пары источников: ионистор [2] и резервный аккумулятор. Ионисторы как источники питания не так известны широкой публике, однако имеют ряд существенных преимуществ, которые, в полной мере раскрываются в рамках данного проекта. К таким преимуществам можно отнести:

- Высокая скорость зарядки;
- Высокая ёмкость;
- Способность «отдавать» высокие токи;
- Значительно большее число циклов заряд-разряд.

Помимо основного источника питания предусмотрен резервный источник в виде аккумулятора. Применение дополнительного аккумулятора необходимо для обеспечения большей автономности работы устройства.

В свою очередь зарядка основного источника питания (ионистора) и резервного источника питания (аккумулятора) осуществляется при помощи солнечной панели.

Для сокращения расхода заряда источника питания используется режим пониженного энергопотребления микроконтроллера, а также эффективный с точки зрения энергопотребления алгоритм функционирования устройства. Режим пониженного энергопотребления подразумевает отключение всех потребителей, которые не выполняют «полезную» работу. Аналогичный процесс

происходит во внутренних системах микроконтроллера – в числе потребителей остаются только «жизнеобеспечивающие» системы. Однако по запросу пользователя или по наступлению времени замеров данных с датчиков устройство выходит из режима сна вместе со всей периферией и после выполнения замеров снова уходит в режим пониженного энергопотребления. Такая организация работы позволяет минимизировать потребление тока и увеличить период автономной работы.

В результате была реализована метеостанция-логгер на базе микроконтроллера ATmega328p. Устройство обладает всеми основными функциями аналогичных устройств и предлагает варианты расширения функционала. В работе приведена структура устройства, а также кратко описан обобщенный алгоритм работы. Корректность работы устройства подтверждена путём макетирования.

Список использованных источников:

1. Национальный доклад о состоянии окружающей среды республики Беларусь [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.ecoinfo.by/wp-content/uploads/2020/11/Национальный_доклад_2019.pdf
2. Статья о работе Белгидрометцентра [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.belta.by/society/view/>
3. Описание микроконтроллера ATmega328 [Электронный ресурс]. – Режим доступа: https://www.mouser.com/pdfdocs/gravitech_atmega328_datasheet.pdf
4. Описание датчика BME-280 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.mouser.com/datasheet/2/783/BST-BME280-DS002-1509607.pdf>
5. Описание датчика MH-Z19B [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.winsensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-ver1_0.pdf
6. Описание дисплея SD1306 [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>
7. Описание модуля microSD карты [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://5.imimg.com/data5/YP/WB/MY-1833510/micro-sd-card-module-for-arduino.pdf>

СОПРОЦЕССОР ДВУМЕРНОЙ СВЕРКИ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ ДЛЯ ПЛИС XILINX ZYNQ

Андросов Евгений Сергеевич

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Петровский Н.А. – канд. тех. наук

Аннотация. Понижение и повышение частоты дискретизации являются фундаментальными и широко используемыми операциями над изображениями, применяемыми в задачах, связанных с отображением, сжатием и последовательной передачей цифровых изображений. Понижающая дискретизация (*downsampling*) – уменьшение пространственного разрешения при сохранении пропорций двумерного представления сопряжена с явлением наложения спектров (*aliasing*), для борьбы с которым используются различные методы фильтрации на основе двумерной свертки.

Ключевые слова. Понижение частоты дискретизации, свертка.

Свертка изображения – это обобщённый алгоритм фильтрации двумерных сигналов, в частности изображений. Существующие ядра свёртки обеспечивают различные операции важные для обработки изображений, такие как: размытие, увеличение резкости и выделение границ. Дискретная свертка для 2-D сигнала описывается следующим выражением [2]:

$$I'_{m,n} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{m+i,n+j} \cdot H_{i,j} \quad (1)$$

где $I_{m,n}$ – значение исходного изображения с координатами (m, n) , $I'_{m,n}$ – результат свёртки в точке (m, n) , $H_{i,j}$ – весовые коэффициенты ядра фильтра размерностью 3×3 . Выражение (1) применяется для всех пикселей входного изображения I . В данной работе рассматриваются только квадратные свертки с разделимым ядром (значения матрицы $H_{i,j}$ могут быть получены как линейная комбинация функций в горизонтальном и вертикальном направлении).

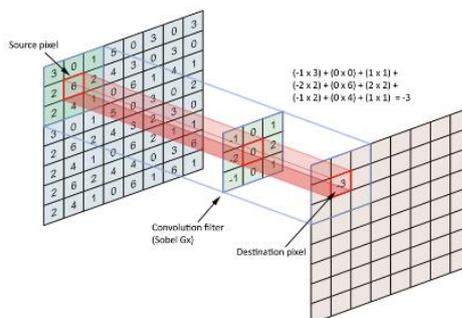


Рисунок 1 – Свертка изображения

Рисунок 1 иллюстрирует простой пример свертки изображения с одним цветовым каналом, где входное изображение I , размером 8×8 пикселей, ядро фильтра H размером 3×3 . Результатом операции полной свертки будет изображение, I' размером 7×7 пикселей. Выходное изображение всегда меньше входного изображения т.к. свёртка не выполняется на граничных пикселях, потому что некоторые из значений не определены. Свёртка выполняется только в том случае, если ядро имеет значения для всех коэффициентов ядра фильтра, таким образом граничные пиксели исключаются из свёртки изображения (рис. 2).

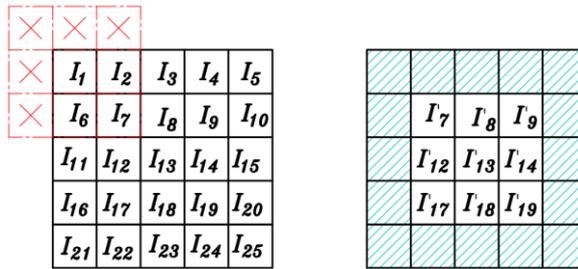


Рисунок 2 – Иллюстрация проблемы обработки граничных значений
 Метод расширения границы позволяет сформировать дополнительные граничные значения. Частным случаем такого подхода является расширения границы нулями (рис. 3).

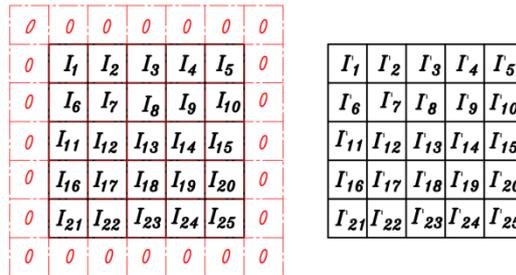


Рисунок 3 – Расширение границы изображения нулями

Аппаратная реализация обработки изображений заключается в особенностях организации потока данных, в частности, изображения подаются и выводятся из системы в построчном порядке, а также могут храниться во *frame*-буфере для работы встраиваемого ПО. Для минимизации дополнительных запросов, свертку необходимо применять в поточном режиме, используются промежуточные буферы для хранения предыдущих n строк предыстории. Таким образом, значение пикселей, загруженные в такой буфер, будут повторно использованы для корректной свертки со последующими строками [1].

Базовая архитектура предложенного процессора свертки использует кольцевой буфер для хранения строк предыстории, реализованный на блочной памяти ПЛИС, где необходимо будет обеспечить корректный расчет адреса по мере поступления входных данных. Выходные значения из памяти формирует данные для операции свертки (рис. 4).

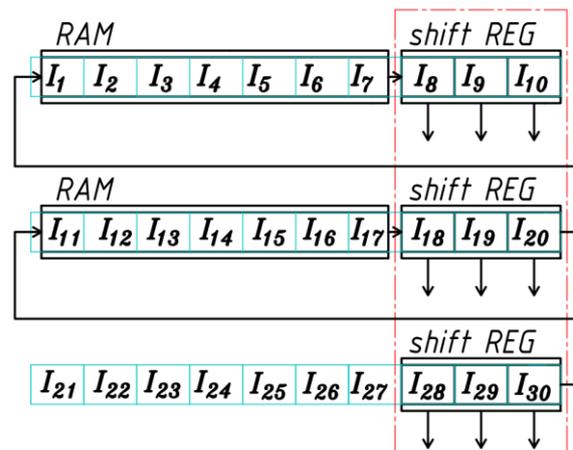


Рисунок 4 – Архитектура буфера

Базовая реализация модуля свёртки содержит n^2 умножителями. После параллельного умножения всех значений пикселей на соответствующий коэффициент свертки H , результаты

умножения передаются на вход дерева сумматоров из $n^2 - 1$ блоков. Результат операции фиксируется в выходном регистре (рис. 5).

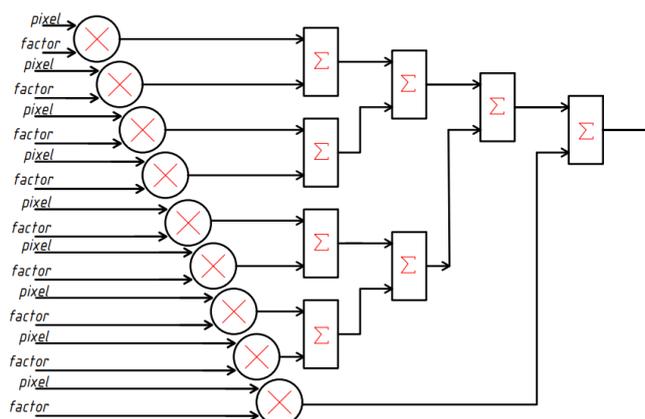


Рисунок 5 – Архитектура модуля свертки

Прореженное и обработанное таким образом изображение в результате получает пониженное разрешение, в общем случае, в два раза, за счет удаления части пикселей. Свертка во временной области с прореживанием является одним из наиболее удобных методов понижения разрешения изображений без существенной потери качества.

Список использованных источников:

1. A Rotation-based Data Buffering Architecture for Convolution Filtering in a Field Programmable Gate Array / Zhijian Lu [et al.]// JOURNAL OF COMPUTERS, VOL. 8, NO. 6, JUNE 2013
2. A Parallel FPGA Implementation of Image Convolution / Henrik Ström // Master of Science Thesis Linköping University, 2016
3. An FPGA 2D-convolution unit based on the CAPH language / Abiel Aguilar-González [et al.]// Journal of Real-Time Image Processing manuscript

UDC

FPGA IMPLEMENTATION OF 2-D CONVOLUTION COPROCESSOR FOR COLORED IMAGES BASED ON XILINX ZYNQ

Androsov E.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Petrovsky N.A. – PhD in Computer science

Annotation. Downsampling and upsampling are fundamental and widely used in image display, compression, and progressive transmission. Downsampling is the reduction in spatial resolution while keeping the same proportions of two-dimensional representation leads to aliasing effect. Various filtering methods, based on convolution are used to deal with this effect.

Keywords. Downsampling, convolution.

УДК 004.855

РЕАЛИЗАЦИЯ АЛГОРИТМА ID3 НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

Габриельчик П.В. и Ермакович В.А., студенты гр.250702

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Вашкевич М.И. – доктор техн. наук

Аннотация. Данная научная работа посвящена изучению основ машинного обучения. В ней исследуются принципы машинного обучения, а также представляется реализация алгоритма ID3 как способа построения дерева решений для классификации данных. Реализация алгоритма производится на языке Python. Результаты эксперимента показывают эффективность и точность алгоритма ID3 при решении задач классификации. Исследование позволяет расширить понимание основ машинного обучения и применять полученные знания на практике.

Ключевые слова. Машинное обучение, Искусственный Интеллект, теория информации, алгоритм ID3, энтропия, информационное обучение, прирост информации, дерево решений, классификация данных, переобучение и недообучение.

1. Введение

Машинное обучение (machine learning) — это раздел искусственного интеллекта, который позволяет компьютерным системам извлекать знания и предсказывать результаты на основе анализа большого количества данных. Машинное обучение используется для решения широкого круга задач, включая классификацию, регрессионный анализ, кластеризацию, обнаружение аномалий.

Существуют различные типы машинного обучения, такие как обучение с учителем (supervised machine learning), обучение без учителя (unsupervised machine learning) и обучение с подкреплением (reinforcement machine learning). В обучении с учителем модель обучается на основе размеченных данных, то есть данных, которые содержат правильные ответы. В обучении без учителя модель пытается найти структуру в данных без разметки. В обучении с подкреплением модель обучается на основе взаимодействия с окружающей средой и получает награду за правильные решения.

Машинное обучение нашло широкое применение в различных отраслях, таких как медицина, банковское дело, транспорт, производство, маркетинг. С помощью машинного обучения можно сократить время и затраты на анализ данных, повысить точность предсказаний, автоматизировать процессы и многое другое.

Одним из основных принципов машинного обучения является создание моделей, которые могут обучаться на данных и применять полученные знания для решения новых задач. В настоящее время существует множество алгоритмов машинного обучения, включая деревья решений, нейронные сети, метод опорных векторов, случайные леса и другие.

Цель и задачи исследования. Цели данного исследования: изучить основы машинного обучения, рассмотреть алгоритм ID3, используемый для построения деревьев решений, а также разработать собственный алгоритм для классификации данных.

Для достижения этих целей были поставлены следующие задачи:

- 1) Изучить основные понятия машинного обучения, такие как энтропия, информационное обучение, прирост информации и деревья решений.
- 2) Изучить алгоритм ID3 и его реализацию для построения деревьев решений.
- 3) Проанализировать примеры использования алгоритма ID3 для классификации данных.
- 4) Разработать код реализации алгоритма ID3 на языке Python.
- 5) Протестировать реализацию алгоритма ID3 на реальных данных и оценить ее точность.
- 6) Сделать выводы об эффективности и применимости алгоритма ID3 для построения деревьев решений в различных областях.

Обоснование выбора алгоритма ID3 для классификации данных. Алгоритм ID3 является одним из самых простых и эффективных алгоритмов для построения деревьев решений на основе данных с категориальными признаками. Его основное преимущество заключается в том, что он может автоматически извлекать наиболее значимые признаки из данных и использовать их для построения оптимального дерева решений.

Алгоритм ID3 использует информационный критерий прироста информации для выбора наилучшего разделения данных на каждом узле дерева. Он также способен обрабатывать отсутствующие данные и имеет возможность обработки больших объемов данных.

В связи с этим, выбор алгоритма ID3 для реализации дерева решений имеет ряд преимуществ:

1. Простота реализации и использования. ID3 легко реализуется на любом языке программирования, в том числе и на Python, и не требует специальных знаний в области машинного обучения.

2. Высокая точность. Алгоритм ID3 обладает высокой точностью и может давать хорошие результаты при правильном выборе параметров.

3. Универсальность. ID3 может быть использован для построения деревьев решений в различных областях, включая банковское дело, медицину, транспорт и т.д.

Таким образом, алгоритм ID3 является оптимальным выбором для реализации деревьев решений на основе категориальных данных.

2. Классификация данных и информационное обучение

Классификация данных — это процесс определения к какому классу относится определенный объект или набор данных. Классификация является одной из основных задач машинного обучения.

В задаче классификации, на основе известных данных, необходимо построить алгоритм, который может определять класс, к которому относится новый объект данных. Классификация данных может быть двуклассовой, когда необходимо отнести объект к одному из двух классов, или многоклассовой, когда необходимо отнести объект к одному из нескольких классов.

Для классификации данных могут использоваться различные алгоритмы машинного обучения, такие как деревья решений, нейронные сети, метод опорных векторов и другие. Каждый из этих алгоритмов имеет свои преимущества и недостатки, и выбор конкретного алгоритма зависит от характеристик данных и требований к точности и скорости работы.

Классификация данных находит широкое применение в различных областях, таких как биология, медицина, экономика, финансы, маркетинг и многие другие. Примеры задач классификации данных включают определение злокачественных опухолей, распознавание рукописных цифр, категоризацию товаров в интернет-магазинах, определение кредитоспособности заемщиков.

Информационное обучение — это подход к машинному обучению, основанный на теории информации, который использует понятие энтропии для описания степени неопределенности данных. Информационное обучение стремится минимизировать энтропию данных, то есть уменьшить их неопределенность путем построения модели, которая наилучшим образом описывает данные.

Энтропия — это мера неопределенности данных. В теории информации энтропия используется для описания количества информации, содержащейся в некотором сообщении. Чем больше энтропия, тем больше неопределенность в данных. В машинном обучении энтропия используется для оценки неопределенности данных в задачах классификации.

В контексте алгоритма ID3, энтропия используется для оценки неопределенности различных атрибутов и выбора наилучшего атрибута для разделения данных на подмножества. Идея заключается в том, что при выборе атрибута, который имеет наименьшую энтропию, можно разделить данные на наиболее однородные группы. Это позволяет построить более точную модель для классификации новых данных.

Информационное обучение и энтропия являются важными понятиями в машинном обучении и широко используются в различных алгоритмах, таких как деревья решений, наивный Байесовский классификатор и другие. Они позволяют учитывать неопределенность данных и создавать более точные модели для классификации и прогнозирования.

3. Деревья решений: принцип обучения

Дерево решений — это метод машинного обучения, который использует древовидную структуру для принятия решений на основе последовательного применения набора правил. Оно может быть использовано для классификации или регрессии данных.

Принцип построения деревьев решений заключается в последовательном разбиении данных на более мелкие подмножества, до тех пор, пока каждое подмножество не будет однородным или достигнет заданного критерия остановки. Это позволяет построить дерево решений, где каждый узел представляет собой проверку значения некоторого признака, а каждый лист соответствует конечному результату, который можно использовать для принятия решений.

Процесс построения дерева решений может быть формализован с помощью алгоритма, который выбирает лучший признак для разбиения данных на каждом шаге. Для этого можно использовать различные критерии, такие как энтропия, прирост информации или индекс Джини, которые оценивают качество разбиения данных.

Построение дерева решений для прогнозирования по тестовому экземпляру начинается с проверки значения признака в корне дерева. Результат этого теста определяет, в какой из дочерних узлов корня необходимо перейти. Затем проверка значения признака и переход вниз по дереву

повторяются до тех пор, пока процесс не достигнет листа, на котором может быть сделано предсказание.

Важно учитывать, что построение деревьев решений может привести к переобучению, когда модель слишком точно подстраивается под обучающие данные и теряет способность обобщать новые. Для решения этой проблемы можно использовать различные методы, такие как ограничение глубины дерева, сокращение или отбор признаков, а также использование ансамблей деревьев решений, таких как случайный лес или градиентный бустинг.

Проблемы переобучения и недообучения являются ключевыми проблемами при построении моделей машинного обучения, в том числе и при использовании деревьев решений.

Переобучение возникает, когда модель слишком хорошо подстраивается под тренировочные данные и с высокой точностью предсказывает их значения, но при этом плохо работает на новых, ранее неизвестных данных. Это происходит, когда модель слишком сложна и адаптируется к шуму в тренировочных данных вместо того, чтобы обобщать закономерности.

Недообучение происходит, когда модель недостаточно обучена на тренировочных данных и не в состоянии уловить сложные зависимости в данных.

Для решения проблемы переобучения можно использовать методы регуляризации, такие как обрезание деревьев, добавление штрафов за сложность модели и т.д. Для решения проблемы недообучения можно использовать более сложные модели, более мощные алгоритмы оптимизации, увеличение объема тренировочных данных.

Важно понимать, что выбор модели зависит от конкретной задачи и доступных данных, и не существует универсального способа решения проблем переобучения и недообучения, но в общем случае нужно стараться достичь баланса между сложностью модели и ее способностью обобщать закономерности в данных.

4. Алгоритм ID3

Алгоритм итеративного дихотомизатора 3 (Iterative Dichotomiser 3 – ID3) является одним из классических алгоритмов машинного обучения, который используется для построения деревьев решений. Он был разработан Россом Кузе в 1960-х годах. ID3 является алгоритмом обучения с учителем, который основывается на концепции прироста информации (information gain).

Алгоритм ID3 строит дерево в рекурсивном порядке обхода в глубину, начиная с корня узла и заканчивая листьями. Алгоритм начинается с выбора наилучшего признака для тестирования. Этот выбор делается путём вычисления прироста информации благодаря признакам в обучающем множестве. Затем корень добавляется в дерево и помечается выбранным проверяемым признаком. После этого обучающее множество разбивается на части с использованием теста. Для каждого возможного результата теста создаётся одно подмножество, которое содержит обучающие экземпляры, возвращающие этот результат. Для каждого подмножества из узла вырастает новая ветвь. Затем этот процесс повторяется для каждой ветви с использованием соответствующего подмножества обучающего множества после исключения соответствующего признака из дальнейшего тестирования. Этот процесс повторяется до тех пор, пока все экземпляры в подмножестве не будут иметь одинаковое значение целевого признака, и в этот момент создаётся лист, помеченный этим значением.

Особенностью алгоритма ID3 является механизм, используемый для определения того, какой признак является наиболее информативным для тестирования в новом узле. Алгоритм ID3 использует метрику прироста информации, чтобы выбрать лучший признак для тестирования на каждом узле дерева. Следовательно, выбор лучшего признака для разделения множества данных основан на однородности результирующих подмножеств в множествах данных. В результате прирост информации для определённого признака может отличаться на разных узлах в дереве. Одним из следствий этого является то, что признак с низким приростом информации в корне может иметь высокий показатель прироста информации в одном из внутренних узлов, поскольку он является прогнозирующим в подмножестве рассматриваемых экземпляров в этом внутреннем узле.

Алгоритм ID3 может использоваться для решения задач классификации и прогнозирования. Он может быть эффективным при работе с данными, которые имеют небольшое количество признаков и где признаки являются категориальными. Однако, если признаки являются числовыми, то необходимо использовать алгоритмы, способные работать с такими данными, например, алгоритм C4.5, который является усовершенствованной версией ID3.

Алгоритм ID3 может быть применен на различных типах данных, включая категориальные и числовые данные. Рассмотрим несколько примеров его применения:

1. Классификация пациентов на основе медицинских данных:

В этом примере, алгоритм может быть использован для классификации пациентов на группы в зависимости от их состояния здоровья. Входными данными являются медицинские показатели, такие как пульс, кровяное давление, уровень сахара в крови и т.д. Алгоритм использует эти данные для построения дерева решений, которое помогает врачам принимать более точные решения в отношении диагностики и лечения пациентов.

2. Классификация посетителей веб-сайта на основе действий:

В этом примере, алгоритм может быть использован для классификации посетителей веб-сайта на группы в зависимости от их действий, например, покупок или регистраций на сайте. Входными данными являются действия посетителей, такие как клики на определенные элементы, время нахождения на страницах, данные о покупках и т.д. Алгоритм использует эти данные для построения дерева решений, которое помогает владельцам веб-сайтов принимать более эффективные маркетинговые решения.

3. Классификация рисков при инвестировании на основе финансовых данных:

В этом примере, алгоритм может быть использован для классификации инвестиционных возможностей на группы в зависимости от их риска. Входными данными являются финансовые показатели, такие как прибыль, уровень инфляции, стоимость акций и т.д. Алгоритм использует эти данные для построения дерева решений, которое помогает инвесторам принимать более обоснованные решения по выбору инвестиционных возможностей.

Кроме того, алгоритм ID3 может быть использован в любых других областях, где необходимо классифицировать данные на основе их признаков.

Анализ времени работы алгоритма ID3 является важной частью его оценки производительности. Время работы алгоритма зависит от размера обучающей выборки, количества признаков и числа возможных значений каждого признака.

В худшем случае, когда каждый элемент обучающей выборки уникален по значениям всех признаков, алгоритм ID3 может построить дерево решений с глубиной, равной количеству элементов в обучающей выборке. Это может привести к значительному увеличению времени работы алгоритма и к переобучению модели.

Для уменьшения времени работы и избежания переобучения, можно использовать различные методы оптимизации, например, прореживание дерева (pruning) или ограничение глубины дерева. Также можно использовать более эффективные алгоритмы, такие как C4.5 и CART, которые являются усовершенствованными версиями алгоритма ID3.

В целом, алгоритм ID3 показывает хорошие результаты на средних и маленьких обучающих выборках, однако на больших выборках может быть неэффективен. Поэтому, при выборе алгоритма для конкретной задачи, необходимо учитывать размер выборки и характеристики данных.

5. Реализация алгоритма ID3 для классификации данных на языке Python

Для реализации алгоритма ID3 был использован язык программирования Python. В качестве основных инструментов были использованы стандартные библиотеки Python, такие как numpy и pandas.

Библиотека numpy была использована для работы с массивами данных, матрицами и вычислений. Библиотека pandas - для анализа и обработки данных.

В целом, для реализации алгоритма ID3 использовались основные инструменты и библиотеки, доступные в языке Python.

Подготовка данных для обучения. Для подготовки данных для обучения в машинном обучении важно, чтобы данные были в правильном формате и содержали необходимую информацию. В данном случае мы рассмотрим подготовку данных для обучения дерева решений с помощью библиотеки pandas.

При работе с таблицей данных в формате xls, которая содержит результаты обследований голосовых характеристик, можно использовать библиотеку pandas для чтения данных из файла и преобразования их в удобный формат. Для этого используем следующий синтаксис:

```
train_data = pd.read_excel('train.xlsx')
```

После выполнения этой команды, данные из файла Excel будут загружены в DataFrame train_data. Теперь мы можем использовать этот DataFrame для анализа данных, включая обработку, визуализацию и обучение моделей машинного обучения.

Затем данные необходимо преобразовать из числового типа в категориальный. Для этого воспользуемся методом .qcut из библиотеки Pandas, применяя его ко всем столбцам таблицы train_data. Например, для столбца 'Jitter' реализация метода .qcut будет выглядеть следующим образом:

```
train_data['Jitter'] = pd.qcut(train_data['Jitter'], q=6,  
                             labels=['1', '2', '3', '4', '5', '6'])
```

Здесь числовой признак "Jitter" разбивается на 6 интервалов равной длины.

Применяем данный метод ко всем столбцам и получаем измененную таблицу, которую можно использовать для построения дерева решений.

Реализация алгоритма. После чтения и обработки данных можно начать реализацию алгоритма.

Сперва необходимо реализовать функции для выбора лучшего критерия для разделения множества на наиболее однородные подмножества:

1. Вычисление энтропии всего набора данных.

2. Вычисление энтропии для отфильтрованного набора данных.
3. Вычисление прироста информации.
4. Поиск наиболее информативного признака (признак с наибольшим приростом информации).

Затем необходимо добавить узел, помеченный признаком с наибольшим приростом информации. Если любое значение признака представляет только один класс, то можно сказать, что значение признака представляет чистый (однородный) класс. Если значение признака не представляет чистый класс, придется расширять его дальше, пока мы не найдем чистый класс.

После выбора чистого класса мы должны удалить строки из набора данных, соответствующие значению выбранного признака.

Процесс построения дерева решений согласно алгоритму ID3 можно представить, как рекурсивное пошаговое выполнение следующих действий:

- 1) Поиск наиболее информативного признака
- 2) Создание узла дерева с именем признака и значениями признака в качестве ветвей.
 - Если класс чистый, добавление листового (конечного) узла к узлу дерева
 - Если класс нечистый, добавление расширяемого узла к узлу дерева
- 3) Сокращение/обновление набора данных в соответствии с чистым классом
- 4) Добавление узла к ветвям в дерево
- 5) Расширение ветви следующего нечистого класса с обновленным набором данных

Условия выхода из рекурсии:

- Набор данных становится пустым после обновления
- Нет расширяемой ветви (все классы чистые)

Оценка точности и эффективности реализации. Для оценки точности и эффективности реализации алгоритма ID3 можно использовать метрики, такие как точность (accuracy), полноту (recall), точность предсказания положительного класса (precision), F1-меру (F1-score) и ROC-кривую.

Для этого необходимо подготовить данные для тестирования, которые должны быть разделены на обучающую и тестовую выборки. Обучающая выборка используется для обучения модели, а тестовая выборка - для оценки ее эффективности и точности.

Для оценки эффективности и точности модели можно использовать кросс-валидацию, которая позволяет оценить ее работу на различных разбиениях данных.

Также можно провести анализ времени работы алгоритма на различных объемах данных и оптимизировать его для улучшения скорости работы.

В целом, рекомендуется использовать алгоритм ID3 для задач классификации на небольших объемах данных, когда требуется простая и понятная модель, которую легко интерпретировать. Однако, при работе с большими объемами данных или задачами, требующими высокой точности, рекомендуется использовать более сложные и точные алгоритмы, такие как C4.5 или CART.

6. Заключение

Основные результаты исследования. К основным результатам нашего исследования относятся:

1. Реализация алгоритма ID3 для задачи классификации на языке Python.
2. Применение алгоритма на примере реальных данных обследований речевых характеристик.
3. Анализ времени работы алгоритма на различных объемах данных.
4. Обнаружение того, что алгоритм ID3 имеет хорошую точность классификации данных, но при этом может быть неэффективен на больших объемах данных из-за вычислительной сложности.
5. Выявление того, что алгоритм ID3 не всегда лучший выбор для задач классификации, так как существуют другие алгоритмы, которые могут иметь более высокую точность классификации и/или более эффективно работать на больших объемах данных.

Выводы и рекомендации по применению алгоритма ID3.

Выводы:

- Алгоритм ID3 является простым и эффективным инструментом для построения деревьев решений на основе информационного обучения.
- ID3 может быть использован для классификации данных в различных областях, включая медицину, бизнес, финансы и другие.
- Для достижения наилучших результатов при использовании алгоритма ID3 необходимо правильно подготовить данные для обучения, выбрать наиболее информативные признаки и оптимально задать пороговые значения для разбиения данных на подгруппы.
- Важным фактором при использовании алгоритма ID3 является контроль за переобучением модели.

Рекомендации:

- При использовании алгоритма ID3 рекомендуется тщательно подготовить данные для обучения, провести предварительный анализ данных и выбрать наиболее информативные признаки для построения дерева решений.

- Необходимо учитывать, что алгоритм ID3 может быть чувствителен к шуму в данных, поэтому желательно провести их предварительную обработку.

- Для контроля за переобучением модели рекомендуется использовать кросс-валидацию и регуляризацию.

- При работе с большими объемами данных, необходимо учитывать время работы алгоритма ID3, который может быть неоптимальным для больших и сложных данных.

- Рекомендуется проводить дополнительные эксперименты с различными параметрами алгоритма и альтернативными методами классификации данных для достижения наилучших результатов.

Список использованных источников:

1. *Основы машинного обучения для аналитического прогнозирования: алгоритмы, рабочие примеры и тематические исследования: Пер. с англ. / Джон Д. Келлехер, Брайан Мак-Нейми, Аоифе д'Арсси – СПб.: ООО «Диалектика», 2019. – 656с: ил.*

UDC 004.855

PYTHON IMPLEMENTATION OF THE ID3 ALGORITHM

Habryelchyk P.V. & Ermakovich V.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vashkevich M.I. – PhD

Annotation. This research paper is devoted to the study of the basics of machine learning. It explores the principles of machine learning and presents an implementation of the ID3 algorithm as a way to build a decision tree for data classification. The implementation of the algorithm is done in Python. The experimental results show the effectiveness and accuracy of the ID3 algorithm in solving classification problems. The research allows us to expand our understanding of the basics of machine learning and to apply the obtained knowledge in practice.

Keywords. Machine learning, Artificial Intelligence, information theory, ID3 algorithm, entropy, information learning, information gain, decision tree, data classification, overtraining and undertraining.

УДК 621.396

СИСТЕМА ОБРАБОТКИ НАВИГАЦИОННЫХ ДАННЫХ: АППАРАТНАЯ АРХИТЕКТУРА И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Гомолко В. А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Вашкевич М.И. – канд. тех. наук, доцент

Аннотация. Статья описывает структуру навигационного приемника. Представленная реализация навигационного приемного устройства обладает высокой универсальностью и гибкостью благодаря возможности модификации навигационных алгоритмов.

Ключевые слова. ГНСС, обработка навигационного сигнала, навигационное приемное устройство.

Вводная часть.

Спутниковые системы навигации предоставляют различные услуги и возможности для разнообразных отраслей и применений, таких как авиация, морская транспортировка, спутниковая связь, а также для частных пользователей. Современные глобальные навигационные спутниковые системы (далее – ГНСС) предназначены для определения геопозиции объектов, скорости, времени, а также направление движения навигационного приемного устройства. Благодаря относительно высокой точности и доступности ГНСС стали неотъемлемой частью современных систем разных областей.

На 2023 год три основные спутниковые системы обеспечивают практически полное покрытие и непрерывное функционирование на глобальном уровне – *GPS* (США), *ГЛОНАСС* (Россия), *BeiDou* (Китай), *GALILEO* (Европейский союз). ГНСС не требуют от пользователей передачи каких-либо данных, что обеспечивает простоту и удобство использования. ГНСС непрерывно вещают сигналы на различных частотах, которые могут быть приняты пользователем, обладающим спутниковым приемником. За изменение параметров ГНСС ответственны органы, отвечающие за их обслуживание и функционирование. Например, система *GPS* контролируется военно-воздушными силами США.

В статье рассматриваются ключевые принципы обработки сигналов ГНСС и приводится пример аппаратной архитектуры на программируемых логических интегральных схемах (далее – ПЛИС). Реализация включает в себя минимальный тракт для получения сигнала от спутника и преобразования в цифровой формат для дальнейшей обработки на процессоре общего или специализированного назначения.

Принципы обработки сигналов ГНСС.

Обработка навигационного сигнала, полученного со спутника, состоит из несколько основных стадий:

1. Радиочастотная часть: на данном этапе осуществляется прием сигнала, усиление, фильтрация и перенос на промежуточную частоту.
2. Аналого-цифровое преобразование: аналоговый сигнал преобразуется в цифровой формат для использования цифровых методов обработки.
3. Цифровая обработка сигнала: включает в себя процессы поиска, слежения за сигналом и извлечение навигационных данных.
4. Решение навигационной задачи: на этом этапе из навигационного сообщения извлекаются данные о местоположении спутников и с использованием этих данных вычисляется геопозиция пользователя на Земле.

Современные спутниковые приемники осуществляют обработку в цифровой части из-за преимуществ подхода, поскольку цифровые методы позволяют реализовывать сложные алгоритмы обработки и анализа, что обеспечивает высокую точность и надежность работы навигационной системы. В зависимости от архитектуры приемника обработка после аналого-цифрового преобразователя осуществляется с использованием отдельных корреляционных каналов либо процессорных ресурсов.

Для получения навигационных данных и определения геопозиции, необходимо определить, какие спутники находятся в видимом звездном созвездии. Этот процесс осуществляется при помощи процесса поиска. Поиск зависит от доступной информации и условий работы приемника, может быть в условиях неопределенности (англ. *cold start*) или в условиях априорной информации (англ. *hot start*):

1. В условиях неопределенности приемник не имеет предварительной информации о спутниках, текущем времени, своем местоположении и т.п. В данном случае приемник выполняет полный поиск по всем спутниковым сигналам, обычно до нескольких минут, из-за большого объема вычислений.
2. В условиях априорной информации приемник уже имеет информацию о спутниках, текущем времени, своем местоположении и т.п. Это может быть вызвано тем, что спутник работал некоторое время или при поддержке вспомогательной системы (англ. *assisted GNSS*), что ускоряет процесс поиска и обработку сигналов [1].

Поиск – стадия, на которой приемник определяет наличие сигнала от спутника и его приблизительные параметры, такие как частота и фазовое смещение [2]. Данный процесс начинается с генерации локальных сигналов, которые соответствуют спутниковым сигналам с различными параметрами фазового смещения и частоты Доплера.

Чтобы не потерять возможность извлечь полезные данные из спутникового сигнала, после успешного приобретения предварительных данных требуется постоянная синхронизация, этот процесс называется слежением. Приемник стремится максимально точно синхронизироваться с полученным сигналом. Это означает определение точного фазового смещения кода и частоты несущей. Для этого, обычно, используются две замкнутые петли: петля фазовой автоподстройки частоты (англ. *phase-locked loop*) и петля слежения за кодовой задержкой (англ. *delay-locked loop*) [3].

ФАПЧ – петля фазовой автоподстройки частоты. Петля отслеживает изменение частоты через разницу фаз. Разница между локально-сгенерированным и входным сигналом – фазовая ошибка. Управляющий сигнал зависит от фазовой ошибки, чем меньше ошибка, тем меньше значение управляющего сигнала подается на генератор сигнала. Этот процесс автоподстройки продолжается до тех пор, пока локальная частота не синхронизируется с частотой входного сигнала.

Петля слежения за кодовой задержкой отслеживает изменение задержки спутникового кода входного сигнала и подстраивает под этот сигнал локально генерируемую задержку. Архитектура петли использует структуру ранний-поздний-точный:

1. Ранний коррелятор: сравнивает локальный код с входным, сдвинутым на несколько кодовых символов назад;
2. Поздний коррелятор: сравнивает локальный код с входным, сдвинутый на несколько кодовых символов вперед;
3. Точный коррелятор: сравнивает локальный код с входным сигналом без задержки.

На основе разницы между ранней и поздней корреляцией, фильтр генерирует сигнал со скорректированной задержкой.

Параллельно процессу слежения происходит извлечение навигационных данных. Извлечение начинается после успешной синхронизации с телеметрическим словом (англ. *telemetry word*). Телеметрическое слово состоит из предопределенной последовательности бит, на основе которой приемник определяет начало подкадра. После успешной синхронизации с телеметрическим словом приемник извлекает битовую последовательность, которая затем объединяется в слова. Полученные слова собираются в структуру навигационного сообщения и обрабатываются в соответствии с интерфейсным документом на спутниковую систему для решения навигационной задачи – определение геопозиции объекта, скорости и времени.

Аппаратная архитектура навигационного приемника

В навигационных приемниках цифровая обработка сигналов может быть реализована как на аппаратном уровне, так и на уровне программного обеспечения, для исполнения инструкций процессором общего назначения вместе с сопроцессорами, при необходимости. До возможной обработки аппаратная архитектура должна как минимум иметь минимальный тракт для получения сигнала от спутника и преобразования в цифровой формат [4]. Этот тракт включает в себя радиочастотный усилитель, фильтры, гетеродин для переноса сигнала на промежуточную частоту

и аналого-цифровой преобразователь. Аппаратно-программная структура представлена на рисунке 1.

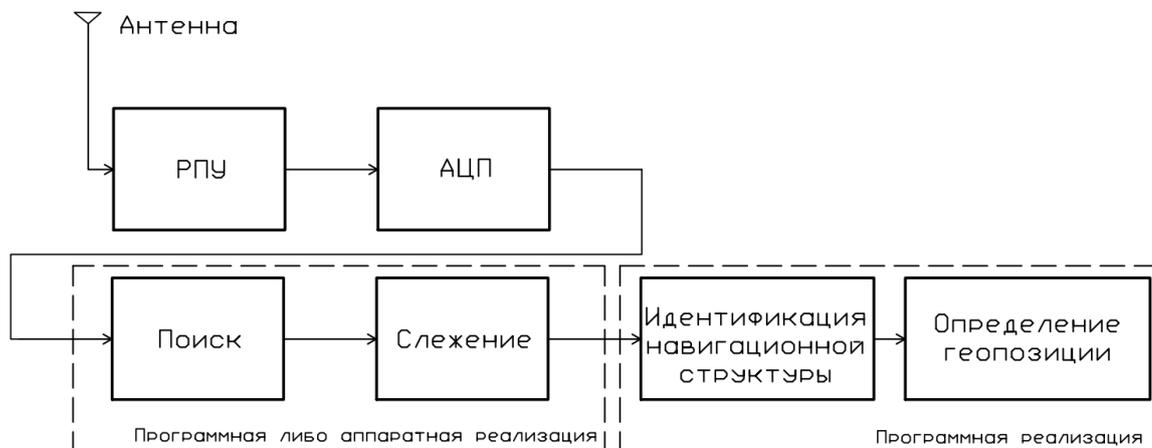


Рисунок 1 – Структура навигационного приемного устройства

В данной реализации тракт до поиска разработан аппаратно, вычисления по цифровым данным происходят с использованием процессорных ресурсов *ARM Cortex A9*, что позволяет получить высокую производительность, энергоэффективность и эффективное использование памяти для встроенной системы [5]. Данное решение позволяет без особой сложности вносить изменения в навигационные алгоритмы при необходимости, что делает приемник универсальным, а программное обеспечение кроссплатформенным: программное обеспечение может работать на большинстве устройств с архитектурой *ARM* без необходимости адаптации программного обеспечения.

Аппаратные компоненты были описаны на языке описания аппаратуры. Ключевыми компонентами являются модуль ввода и цифровой канал обработки. Модуль ввода выполняет функцию первичной настройки всех компонентов системы, таких как радиочастотная схема для приема данных и цифрового канала обработки, а также калибровку АЦП.

В цифровом канале происходит перенос на нулевую промежуточную частоту и демодуляция сигнала с использованием квадратурного гетеродина. После демодуляции сигнал преобразуется в две составляющие: *I* и *Q*. Нежелательные гармоники подавляются низкочастотным фильтром с децимацией, что позволяет снизить частоту дискретизации и сэкономить вычислительные ресурсы. Затем комплексные отчеты передаются через контроллер прямого доступа к памяти в оперативную память системы для дальнейшей обработки при помощи программного обеспечения.

После того, как комплексные отчеты были переданы в оперативную память, происходит навигационная обработка: определения первичных параметров по полученным данным, таких как частота и фазовое смещение. После успешного определения первичных параметров происходит стадия слежения, где происходит постоянная синхронизация с сигналом и последующее извлечение навигационных данных с целью определения геопозиции объекта, скорости и времени.

Аппаратные ресурсы.

Далее представлена таблица 1, которая показывает количество и типы ресурсов, используемых в данной аппаратной архитектуре навигационного приемного устройства на *Zynq 7015*.

Таблица 1 – Аппаратные затраты на ПЛИС *Zynq 7015*

Ресурс	Количество	Доступно
<i>LUT</i>	8257	46200
<i>LUTRAM</i>	511	14400
<i>FF</i>	12372	92400
<i>BRAM</i>	8	95
<i>DSP</i>	54	160
<i>IO</i>	32	150
<i>MMCM</i>	1	3

Вывод.

В статье были рассмотрены ключевые принципы обработки сигналов глобальных навигационных спутниковых систем и рассмотрен пример аппаратной архитектуры на программируемых логических интегральных схемах. Обработка навигационного сигнала включает несколько основных стадий, таких как радиочастотная часть, аналого-цифровое преобразование, цифровая обработка сигнала и решение навигационной задачи.

Список использованных источников:

1. Frank Van Diggelen (2009). *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House.
2. Гомолко В. А. Быстрый поиск GPS-сигнала. *Материалы 58-й научной конференции аспирантов, магистрантов и студентов БГУИР*. – Минск, 2022. Руководитель: Вашкевич М.И.
3. Roland E. Best (2007). *Phase-Locked Loops: Design, Simulation, and Applications*. McGraw-Hill Education.
4. Doberstein D. *Fundamentals of GPS receivers : a hardware approach* – New York : Springer, ©2012. С
5. James A. Langbridge (2014). *Professional Embedded ARM Development*. Wrox.

UDC 621.396

NAVIGATION DATA PROCESSING SYSTEM: HARDWARE ARCHITECTURE AND SOFTWARE

Gomolko V. A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vashkevich M.I. – PhD in Physics and Mathematics

Annotation. The article describes the structure of a navigation receiver. The presented implementation of the navigation receiver has high universality and flexibility due to the ability to modify navigation algorithms with minimal effort.

Keywords. GNSS, navigation signal processing, navigation receiver.

УДК

ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ, ПОСТРОЕННЫХ НА БАЗЕ КОМБИНИРОВАННОГО ГЕНЕРАТОРА

Кайкы М.Н.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь**Иванюк А.А. – доктор тех. наук, профессор каф. Информатики*

Данная работа направлена на исследование характеристик случайности и уникальности физически неклоняемых функций, построенных на основе схемы комбинированного генератора. В работе рассмотрены структура схемы комбинированного генератора и режимы его работы. Представлены экспериментальные результаты, полученные при исследовании характеристик рассматриваемого типа физически неклоняемых функций на различных экземплярах программируемых логических интегральных схем - ПЛИС Xilinx семейства Zynq7000.

Введение.

В повседневной жизни люди используют электронные устройства для совершения покупок, перевода денег, записи информации. В компаниях и банках данные обрабатываются и хранятся электронным оборудованием. К сожалению, как и программное обеспечение, аппаратное обеспечение имеет высокие риски в области безопасности. Безопасность цифровых устройств – по сей день является актуальной темой для большинства людей и компаний, принимающих участие в цифровизации современного общества. Поскольку масштабы интегральных схем (ИС) быстро растут, а режим производства становится более гибким, основные проблемы безопасности ИС связаны с внедрением вредоносных схем, называемых аппаратными троянами (Hardware Trojan) [1]. С целью повышения уровня безопасности современных ИС, а также недопущению несанкционированного использования последних – применяются методы аппаратной идентификации и аутентификации, использующие методы физической криптографии для получения неповторимых и уникальных последовательностей. Данные последовательности могут выступать как в роли идентификатора цифрового устройства, так и в качестве источника энтропии, например для генерации закрытых ключей в алгоритмах шифрования. Для получения описанных последовательностей современные методы физической криптографии применяют понятие физически неклоняемых функций (ФНФ), впервые описанных в работе [2]. В данном исследовании, рассматривается возможность применения одной из реализаций ФНФ на базе ячеек комбинированного генератора случайных чисел, предложенных в работе [3].

Комбинированный генератор на базе нескольких ФНФ.

В работе [3], предлагается подход к построению генераторов истинно случайных чисел, предлагая совместить в одной базовой схеме генератора различные структуры физически неклоняемых функций. Как известно, каждая схема ФНФ обладает своими свойствами и особенностями [4, 5]. Комбинированная схема (Рис. 1), совмещает в себе такие виды ФНФ – статическая память, кольцевой генератор и элемент постобработки на базе Т-триггера.

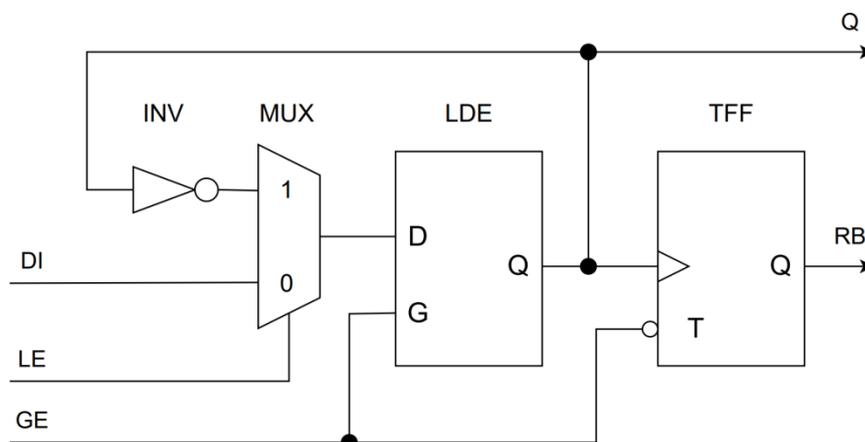


Рисунок 1 – Схема базового элемента комбинированного генератора

Данная схема базового элемента комбинированного генератора может работать в четырёх режимах:

1. *Режим загрузки данных.* При установке управляющих сигналов $LE = 0$ и $GE = 1$, значение, поданное на вход DI будет записано в триггер LDE и отображено на выходе Q . При этом, критический путь для записи будет обусловлен не только временами предустановки и удержания триггера LDE , а и логическими вентилями, образующими мультиплексор, что снижает максимально возможную частоту работы данной схемы в качестве запоминающего устройства.
2. *Режим кольцевого осциллятора.* При переходе в данный режим работы ($LE = 1$, $GE = 1$), базовый элемент комбинированного генератора начинает вырабатывать на выходе Q значения, изменяемые с частотой F_Q , определяемой задержками распространения сигнала через структуру элемента. Такое поведение обусловлено комбинационной обратной связью по пути $INV \rightarrow MUX \rightarrow LDE \rightarrow INV$.
3. *Режим хранения данных.* ($GE = 1 \rightarrow 0$, $LE = 0$) Данный режим необходим для фиксации данных, полученных в режиме загрузки или кольцевого осциллятора.
4. *Режим инициализации.* При включении питания на экземпляре цифрового устройства, и управляющего сигнала $GE = 0$ схема будет находиться в режиме инициализации, что приведёт к её функционированию как ФНФ статической памяти.

Экспериментальная установка.

Для проведения экспериментальной части исследования характеристик физически неклонированной функции на базе базовых элементов, предложенных в работе [3], была спроектирована цифровая система на базе программируемой логической интегральной микросхемы (ПЛИС) компании Xilinx – ZYNQ7000 [6], кристалл – xc7z010clg400-1. Выбранный кристалл ПЛИС располагался на отладочной плате, разработанной компанией Digilent – Zybo Z7 [7]. Для получения доступа к расположенным на элементной базе ПЛИС физически неклонированным функциям была разработана IP компонента, позволяющая передавать на ФНФ входные последовательности от хост-процессора при помощи интерфейса AXI4-Lite. Также, в состав IP компоненты был добавлен набор устройств фиксации результата и систему анализа частот для каждого выхода генератора. Полученные в результате экспериментов данные обрабатывались встроенным в кристалл ПЛИС процессором ARM Cortex-A9, а затем передавались на рабочую станцию при помощи интерфейса UART со скоростью передачи 115200 бод и моста USB – UART на базе микросхемы конвертера FT232RL. Схема генератора и описанных выше систем была описана на языке SystemVerilog и использованием готового решения от компании Xilinx – AXI4 BRAM Controller в виде подключаемой к проекту IP компоненты, структурная схема экспериментальной установки изображена на рисунке 2. Всего в эксперименте принимало участие

$N = 32$ базовых элемента. Для контроля частот работы элементов в режиме кольцевого генератора была применена схема с двумя счётчиками – счётчик импульсов и счётчик временного окна, схема их подключения к базовому элементу приведена на рисунке 3. При функционировании элемента в режиме КО – импульсы с выхода Q поступают на вход счётчика импульсов до тех пор, пока тот не перейдёт в своё конечное состояние счёта ($counter = 2^{32} - 1$) или пока счётчик временного окна не достигнет заданного хост-контроллером значения. Всего, для размещения экспериментальной установки на кристалле, при $N = 32$ понадобилось 4334 6-ти входных таблиц истинности (LUT) и 3425 триггеров.

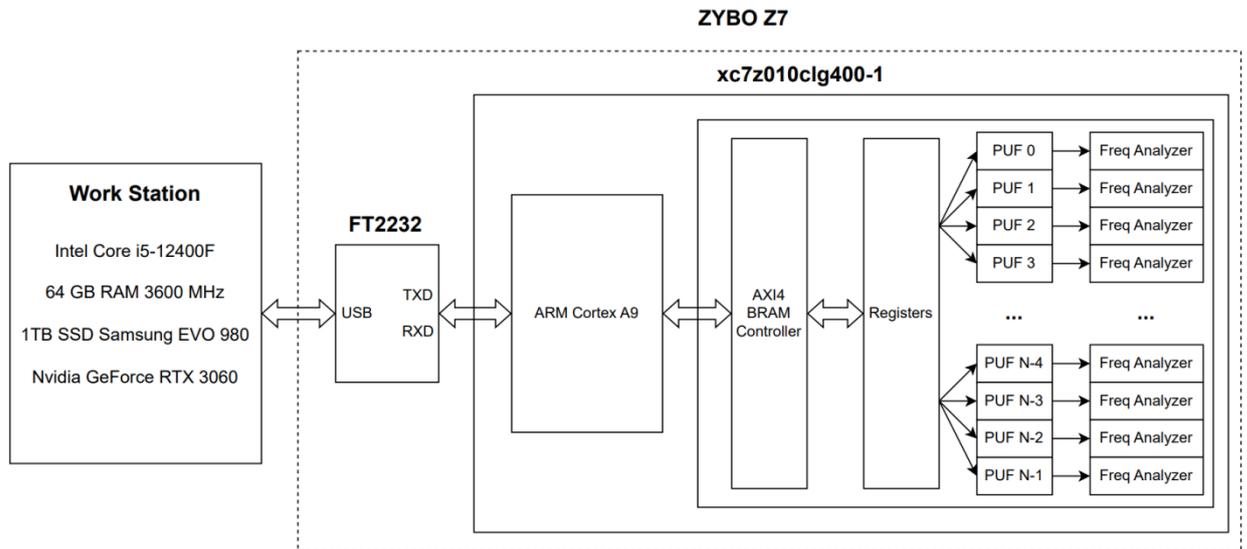


Рисунок 2 – Схема экспериментальной установки на базе ПЛИС

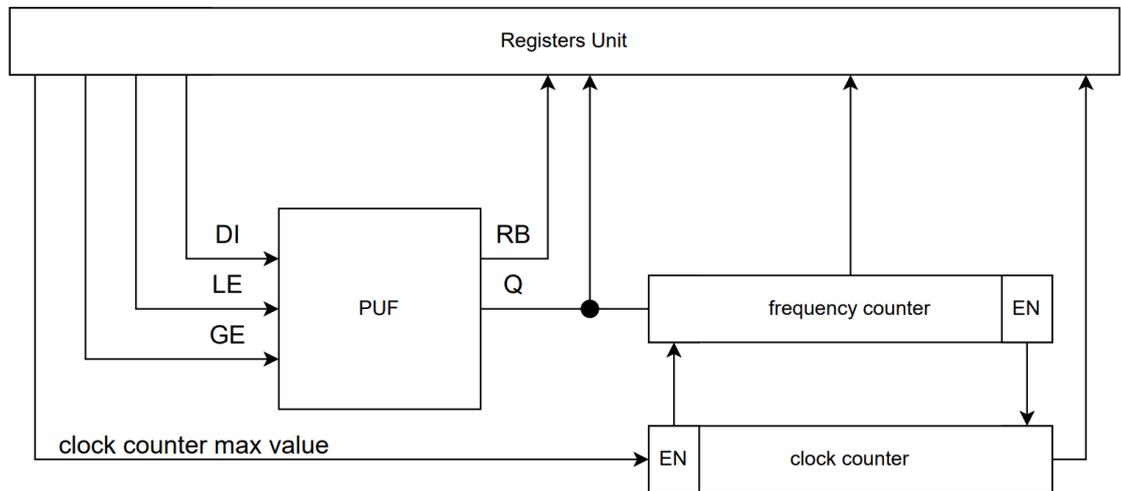


Рисунок 3 – Схема подключения анализатора частот к базовому элементу

Ход проведения эксперимента.

В ходе проведения ряда экспериментов были получены выходные последовательности набора базовых элементов комбинированного генератора, работающего в режимах инициализации и кольцевого генератора.

В первом эксперименте, при анализе данных, полученных в режиме инициализации оценивалась встречаемость единичного символа (P) выходе Q для каждого из базовых элементов (N). Для этого, были произведены циклы перезагрузки платы с ПЛИС, при помощи TCL скрипта, работающего в режиме отладки процессорного ядра ARM Cortex-A9. В результате проведения данного эксперимента был получен набор инициализационных значений для каждой ячейки в зависимости от номера цикла перезагрузки экземпляра ПЛИС, который позволил построить тепловую карту значений при инициализации – рисунок 4.

N	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P_0	1.0	1.0	0	0	0	1.0	1.0	0	0	0	1.0	0	0	0.67	0	1.0	1.0	0	0	1.0	0	1.0	0	0	1.0	0	0	0.94	1.0	0	0	1.0
P_1	1.0	1.0	1.0	0.19	0	1.0	0	0	1.0	0	1.0	0	0	0.83	0	1.0	1.0	0	0	0	0	1.0	0	1.0	1.0	0	0	1.0	1.0	0	1.0	1.0

Рисунок 4 – Тепловая карта инициализационных значений для двух реализованных систем

Для полученных данных проведем оценку основных характеристик ФНФ: стабильность, уникальность, единообразие. Как видно из рисунка 4, большинство базовых элементов являются

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

стабильными – 93.75% ячеек не изменяют своих значений в зависимости от цикла перезапуска. Такое поведение связано непосредственно со структурой бистабильного элемента на базе асинхронного D -триггера и наличием асимметрии в его реализации, а меньшая асимметрия приводит к попаданию схемы в метастабильное состояние, что доказывают значения $0 < P < 1$ [3].

В результате проведения оценки стабильности полученных идентификаторов были сформированы два базовых вектора по мажоритарному правилу:

$V = \{id_0, id_1, id_2, \dots, id_{N-1}\}$, где $id_i = 0$, если $P_i < 0,5$, иначе $id_i = 1$, i – номер разряда в идентификаторе.

Для оценки уникальности будем использовать удельное расстояние по Хэммингу (HD) (формула 1) между данными векторами.

$$Uniq = \frac{1}{N} HD(V_0, V_1) \quad (1),$$

где HD – расстояние по Хэммингу между векторами V_0 и V_1 разрядности N , где V_0 – идентификатор, полученный на первом экземпляре ПЛИС, V_1 – на втором. В результате анализа было получено среднее значение метрики межкристальной уникальности $HD = 0,1875$ между полученными идентификаторами в процессе проведения эксперимента.

Расчёт метрики единообразия проводился по формуле 2, данная метрика отражает соотношения нулей и единиц в каждом из полученных идентификаторов.

$$U_i = 100 \times \left(1 - 2 \times \left| \frac{WH(V_i)}{N} \right| \right) \% \quad (2),$$

где $WH(V_i)$ – вес бинарного вектора V по Хэммингу; i – номер экземпляра ПЛИС; N – разрядность вектора (идентификатора). Среднее единообразие составило: $U_0 = 93.78\%$ $U_1 = 98.12\%$

Вторым этапом проведения экспериментов стало изучение значений частот работы кольцевых генераторов, имеющих в структуре базовых элементов. Для этого, базовые элементы переводились хост-процессором в режим КО во временном окне $W = k \times P_{SYS_CLK} = 1.19$ мс для двух экземпляров генератора, где k – регулируемый коэффициент масштабирования (максимальное значение счётчика тактов), P_{SYS_CLK} – период системного синхросигнала, равного 100 МГц. Результаты измеренных частот приведены на рисунке 5. Измерения проводились при последовательном переключении генератора из режима инициализации в режим КО на временное окно W и далее в режим сохранения данных.

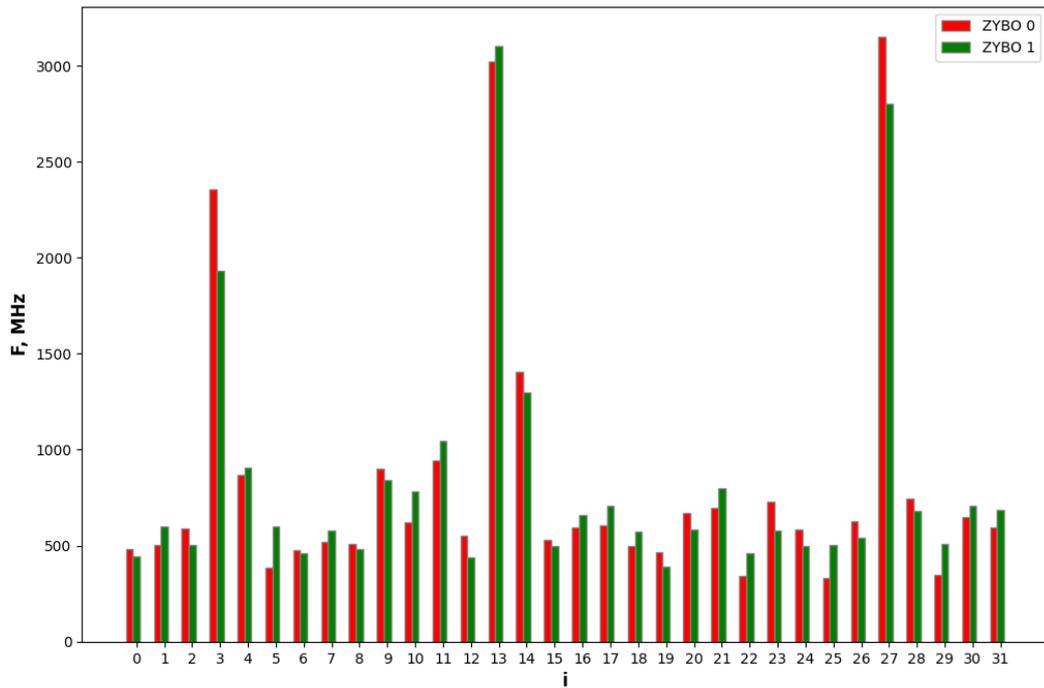


Рисунок 5 – Значения измеренных частот на двух различных экземплярах ПЛИС

Присутствующие выбросы на разрядах **3, 13, 27** можно связать с высокочастотными колебаниями на выходе Q базовых элементов, которые привели к аномальному функционированию измеряющих счётчиков, также, можно заметить, что данные выбросы были зарегистрированы на ячейках, имеющих низкую стабильность в режиме инициализации, что также подтверждает данную гипотезу в виду высокой степени симметрии полученных структур.

Как упоминается в работах [8, 9], подобные высокочастотные колебания на информационном входе триггера/защёлки способны вводить его в метастабильное состояние, имеющее форму затухающих автоколебаний. Так как реализация аналого-цифровых преобразователей высокого разрешения внутри современных интегральных микросхем по сей день является трудно решаемой задачей, а в исследуемых образцах ИС отсутствуют системы регистрации сигналов со схожими АЦП характеристиками, было предложено регистрировать подобные автоколебания при помощи счётчиков передних фронтов сигналов.

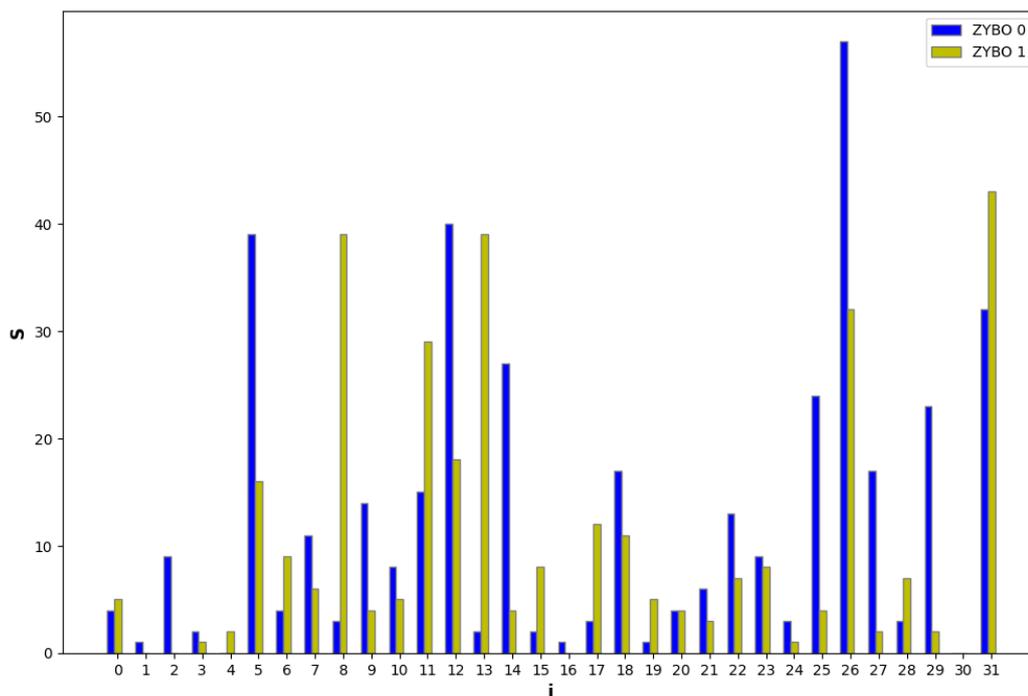


Рисунок 6 – Средние значения количества фронтов автоколебаний

Для регистрации данного эффекта автоколебаний, разработанные ячейки ФНФ переводились в режим функционирования как кольцевые осцилляторы во временном окне равном 1.19 мс, затем, режим работы изменялся на режим хранения с включением систем регистрации передних фронтов на временное окно 600 мс. Данное временное окно было выбрано исходя из времени переходного процесса на триггере – мы не можем знать точное время выхода триггера из метастабильного состояния, поэтому регистрируем фронты колебаний как можно дольше, с целью убедиться в том, что все триггеры перешли в устойчивое состояние. На рисунке 6 приведены полученные в результате эксперимента средние значения количества фронтов автоколебаний для каждой ячейки и экземпляра ПЛИС. Как видно из рисунка 6, почти все ячейки ФНФ генерировали автоколебательные процессы на выходе Q, что подтверждает гипотезу о колебательных процессах в защёлке при нарушении параметров предустановки и удержания входных данных. Стоит отметить, что все элементы ФНФ, участвующие в экспериментах, в конечном итоге пришли к своему стабильному состоянию, что также подтверждает гипотезу о форме колебаний как затухающих.

Заключение.

В результате проведения исследований характеристик физически неклонированных функций на базе предложенного в работе [3] комбинированного генератора были получены значения стабильности, уникальности и единообразия для базовых ячеек ФНФ, работающих в режиме инициализации, а также измерены частоты колебаний данных ячеек в режиме работы как кольцевые осцилляторы. В режиме инициализации полученные структуры ведут себя свойственно статической памяти и обладают схожими с ней характеристиками [10], они достаточно стабильны между перезагрузками (93.75%), имеют хорошие показатели единообразия ($U_0 = 93.78\%$ $U_1 = 98.12\%$). В режиме работы в качестве кольцевого осциллятора базовые элементы показывают среднее значение частот колебаний равное 821 и 818 МГц для каждого из экземпляров ПЛИС, однако наблюдались и высокочастотные выбросы, приводящие к сбоям в работе измеряющих структур. Как показали проведённые эксперименты, базовые ячейки, предложенные в работе [3] – могут выступать как в качестве источников энтропии для генераторов случайных чисел, так и как идентификаторы в аппаратном обеспечении. Отдельно стоит обратить внимание на третью часть эксперимента – исследование эффекта автоколебательных процессов при завершении работы базового элемента в режиме кольцевого генератора. При переходе из режима работы КО в режим хранения данных, защёлка LDE переходила в режим метастабильности с генерацией колебательных процессов на её выходе. Полученный эффект остаточных колебаний связан непосредственно с асимметричной структурой D-защёлки, и позволяет получать неконтролируемые, уникальные для каждого экземпляра, затухающие колебания на выходе базового элемента. Данные колебания могут стать основой для будущих систем генерации истинно случайных чисел, так как не подвержены прямому влиянию управляющей системы и обладают высокой степенью уникальности.

Список использованных источников:

1. Sumathi G., Srivani L., Murthy D.T., Madhusoodanan K., Murty S.A.V.S. A Review on HT Attacks in PLD and ASIC Designs with Potential Defence Solutions. *IETE Tech. Rev.* 2018; 35:64–77. [Google Scholar]
2. Pappu, R. *Physical One-Way Functions: Ph.D. thesis / R. Pappu // MIT. – Boston, USA, 2001.*
3. Иванюк А.А. Комбинированный генератор случайных чисел на программируемых логических интегральных схемах. *Цифровая трансформация.* 2023;29(1):36-47. <https://doi.org/10.35596/1729-7648-2023-29-1-36-47>
4. Claes, M., van der Leest, V., Braeken, A. (2012). Comparison of SRAM and FF PUF in 65nm Technology. In: Laud, P. (eds) *Information Security Technology for Applications. NordSec 2011. Lecture Notes in Computer Science, vol 7161.* Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-29615-4_5
5. Abhranil Maiti, Vikash Gunreddy, and Patrick Schaumont. A Systematic Method to Evaluate and Compare the Performance of Physical Unclonable Functions <https://eprint.iacr.org/2011/657.pdf>
6. Семейство ПЛИС – ZYNQ7000 [Электронный ресурс]. – Режим доступа: <https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html>. – Дата доступа - 15.03.2023
7. Отладочная плата на базе ПЛИС – ZYBO Z7 [Электронный ресурс]. – Режим доступа: <https://digilent.com/reference/programmable-logic/zybo-z7/start> – Дата доступа - 15.03.2023
8. Kasprzak T. (1988) Analysis of Oscillatory Metastable Operation of an RS Flip-Flop. *IEEE Journal of Solid-State Circuits.* 23 (1), 260–266.
9. Zalivaka S. S., Ivaniuk A. A., Chang Ch. H. (2018) Reliable and Modeling Attack Resistant Authentication of Arbiter PUF in FPGA Implementation with Trinary Quadruple Response. *IEEE Transactions on Information Forensics and Security.* 14 (4), 1109–1123.
10. Кайки, М. Н. Сравнение характеристик ФНФ статической памяти с использованием плюс и промышленных микросхем = Comparison of static memory puf characteristics using FPGA and industrial lcs / М. Н. Кайки, А. А. Иванюк // *Приборостроение-2022 : материалы 15-й Международной научно-технической конференции, 16-18 ноября 2022 года, Минск, Республика Беларусь / редкол.: О. К. Гусев (председатель) [и др.]. – Минск: БНТУ, 2022. – С. 37-39.*

UDC

INVESTIGATION OF THE CHARACTERISTICS OF A PHYSICALLY UNCLONABLE FUNCTION BUILT ON THE BASIS OF A COMBINED GENERATOR

Kaiky M.

*Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus*

Ivaniuk A.A. – Dr. of Sci. (Eng.), Associate Professor, Professor at the Comp. Sci. Department

This work is aimed at studying the characteristics of randomness and uniqueness of a physically uncloneable function built on the basis of a combined generator circuit. The paper considers the structure of the combined generator circuit and its operating modes. Experimental results obtained in the study of the characteristics of this physically uncloneable function on various copies of programmable logic integrated circuits - FPGA Xilinx of the Zynq7000 family are presented.

РАСПОЗНАВАНИЕ ЭМОЦИЙ С ИСПОЛЬЗОВАНИЕМ КЕПСТРАЛЬНОГО ПРЕДСТАВЛЕНИЯ РЕЧЕВОГО СИГНАЛА И МЕТОДА ОПОРНЫХ ВЕКТОРОВ

Краснопрошин Д.В., магистрант гр.255741

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Вашкевич М.И. – доктор. техн. наук

Аннотация. Экспериментально исследуется возможность применения метода опорных векторов (МОВ) для классификации эмоций в человеческой речи. Представлен вариант реализации классификатора (на основе МОВ) с использованием линейной ядерной функции. Показано, что полученная модель позволяет определять эмоции с точностью до 85%.

Ключевые слова. Метод опорных векторов, МОВ, распознавание, цифровая обработка сигналов, машинное обучение.

Введение

Одной из актуальных прикладных задач, связанных с созданием эффективного человеко-машинного взаимодействия является построения интерфейса, приближенного к естественным условиям. Для решения данной задачи требуется, чтобы компьютер был способен воспринимать текущую ситуацию и реагировать в соответствии с этим восприятием. Одним из условий для адекватного восприятия является понимание эмоционального состояния пользователя.

Среди основных способов выражения человеческих эмоций важная роль отводится его речи. За последние годы было проведено большое количество исследований по распознаванию (классификации) эмоций на основе речи [1-2].

Существуют различные варианты решения данной проблемы. В частности, можно отметить подходы, основанные на использовании нейронных сетей, байесовского классификатора в сочетании с методом максимального правдоподобия, скрытых марковские модели и т. д. [1]

В данной работе предлагается подход для классификации человеческих эмоций с использованием метода опорных векторов (МОВ).

Набор данных

При проведении исследования в качестве исходного набора данных использовался Toronto emotional speech set (TESS) [3].

Набор данных TESS представляет собой речевые аудиофайлы в формате .wav (16 бит, 48 кГц). Общее количество файлов: 2800. Озвучка была выполнена двумя профессиональными актрисами (в возрасте 26 и 64 лет), озвучивающих два лексически совпадающих высказывания с нейтральным североамериканским акцентом. Для обеих актрис английский язык является родным. Обе имеют университетское и музыкальное образование. Речевые эмоции включают выражения спокойствия, счастья, грусти, гнева, страха, удивления и отвращения. Каждое выражение производится на двух уровнях эмоциональной интенсивности (нормальный, сильный) с дополнительным нейтральным выражением.

Анализ речевого сигнала

Для построения системы по распознаванию эмоций в речи требуется провести предобработку исходных данных. Основной задачей предобработки является удаление шума, повышение высоких частот сигнала и получение плоского частотного спектра сигналов и частотных характеристик.

Еще одним важным шагом является выделение и выбор признаков. Обычно выделяется тональность и ее изменение, скорость произношения и другие спектральные характеристики.

В рамках данной работы для извлечения признаков использовалась техника на основе расчета Мел-частотных кепстральных коэффициентов. Данная техника подразумевает следующие шаги:

1) *Предыскажение*: увеличивает величину энергии на более высокой частоте. В случаях, когда рассматривается частотная область звукового сигнала для звонких сегментов, таких как гласные, видно, что энергия на более высокой частоте намного меньше, чем энергия на более низких частотах. Повышение энергии на более высоких частотах повысит точность и производительность модели;

2) *Кратковременное преобразование Фурье*: это особый вид преобразования Фурье, благодаря которому можно узнать, как частоты в сигнале меняются во времени. Он работает, разрезая ваш сигнал на множество небольших сегментов и выполняя преобразование Фурье каждого из них. В результате обычно получается каскадный график, показывающий зависимость частоты от времени;

3) *Расчет набора из М-фильтров*: используется для моделирования свойств человеческого слуха на этапе выделения признаков, что позволяет улучшить производительность модели. Поэтому мы будем использовать мел-шкалу, чтобы сопоставить фактическую частоту с частотой, которую воспринимают люди. Формула отображения приведена ниже:

Отметим, что человеческий слух менее чувствителен к изменению энергии звукового сигнала при более высокой энергии по сравнению с более низкой энергией. Логарифмическая функция также имеет аналогичное свойство, при низком значении входного x градиент логарифмической функции будет выше, но при высоком значении входного градиента значение меньше. Поэтому мы применяем \log к выходу Mel-фильтра, чтобы имитировать человеческий слух.

4) *Дискретное косинусное преобразование (ДКП)*: Проблема с полученной спектрограммой заключается в том, что коэффициенты банка фильтров сильно коррелированы. Поэтому нам нужно декоррелировать эти коэффициенты. Для этого применяется ДКП.

В результате мы получим набор чисел, являющихся мел-частотными кепстральными коэффициентами (МЧКК).

Классификация

Метод опорных векторов выполняет классификацию путем построения N -мерных гиперплоскостей, которые оптимально разделяют данные на отдельные категории. Классификация достигается путем построения в пространстве входных данных линейной (или нелинейной) разделяющей поверхности. Идея данного подхода заключается в преобразовании (с помощью функции ядра) исходного набора данных в многомерное пространство признаков. И уже в новом пространстве признаков добиться оптимальной в определенном смысле классификации.

В качестве ядра используется любая симметричная, положительно полуопределенная матрица K , которая составлена из скалярных произведений пар векторов x_i и x_j , где $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, характеризующих меру их близости. А ϕ является функцией, формирующее ядро. В частности, примерами таких функций являются:

- **линейное ядро**:

$$K(x_i, x_j) = x_i^T x_j,$$

что соответствует классификатору на опорных векторах в исходном пространстве

- **полиномиальное ядро со степенью p** :

$$K(x_i, x_j) = (1 + x_i^T x_j)^p$$

- **гауссово ядро (радиальная базисная функция)**:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$$

В качестве ядра для модели на основе МОВ была выбрана линейная функция. Значение параметра C (cost) (допустимый штраф за нарушение границы зазора) было равно единице.

Построение классификатора на опорных векторах с использованием перечисленных выше ядер можно, в частности, осуществить с помощью библиотеки `sklearn`, написанной на языке Python.

Описание эксперимента

Исходный набор данных был разбит на тренировочную (70%) и тестовую (30%) выборки.

Для оценки качества работы модели было вычислено среднее арифметическое (невзвешенное) полноты рассчитанной для каждого распознанного класса.

Полнота представляет собой отношение ИП/(ИП + ЛО), где ИП — количество истинных положительных результатов, а ЛО — количество ложноотрицательных результатов. Также под полнотой понимается интуитивно способность классификатора находить все положительные образцы.

Значение полноты лежит в диапазоне от 0 до 1.

В результате построения и обучения модели был получен классификатор, точность предсказаний которого при использовании тестового набора данных и вышеуказанной метрики качества достигала 85%.

Далее будет представлена мультиклассовая матрица спутывания (англ. Multiclass Confusion Matrix) представляющая собой таблицу или диаграмму, показывающая точность прогнозирования классификатора в отношении двух и более классов. Ячейки таблицы заполняются количеством прогнозов классификатора. Правильные прогнозы идут по главной диагонали от верхнего левого угла в нижний правый.

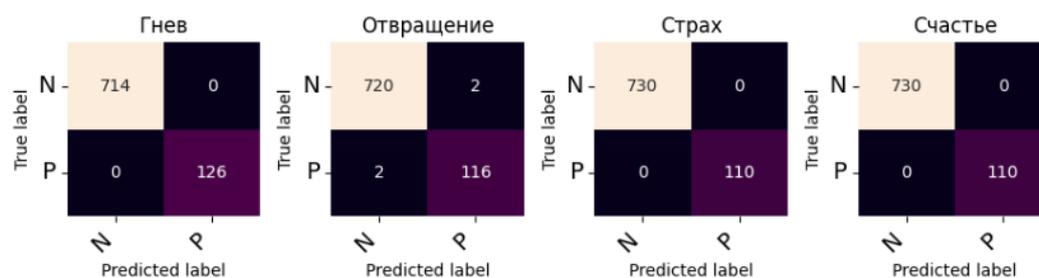


Рисунок 1 – Мультиклассовая матрица спутывания (часть 1)

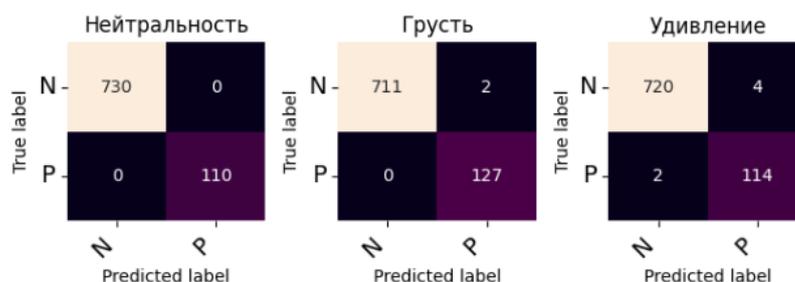


Рисунок 2 – Мультиклассовая матрица спутывания (часть 2)

Выводы

Анализ полученных результатов показал, что метод опорных векторов достаточно успешно справляется с задачей распознавания речевых эмоций. Тем не менее, для более комплексных входных данных (большее количество актеров разного пола и возрастов) этого метода может оказаться недостаточно. В связи с этим для решения обозначенной задачи, возможно, следует попробовать более сложные модели. Таковыми, например, являются скрытые Марковские модели, сверточные нейронные сети и долговременная память (особая разновидность архитектуры рекуррентных нейронных сетей, способная к обучению долговременным зависимостям), поскольку они лучше отражают временную динамику, включенную в речь человека.

Список использованных источников:

1. L. Chen, X. Mao, Y. Xue, and L. Cheng, "Speech emotion recognition: Features and classification models," *Digital Signal Processing*. Vol. 22, No. 6, pp. 1154-1160, 2012.
2. D. Ververidis and C. Kotropoulos, "Emotional speech recognition: Resources, features, and methods," *Speech Communication*. Vol. 48, No. 9, pp. 1162-1181, 2006.
3. Toronto emotional speech set (TESS) / Kaggle – Режим доступа: <https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess>. – Дата доступа: 03.03.2023.

UDC 621.3.049.77–048.24:537.

RECOGNITION OF EMOTIONS USING THE CEPSTRAL REPRESENTATION OF A SPEECH SIGNAL AND THE SUPPORT VECTOR MACHINE

Krasnoproshin D.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vashkevich M.I. – PhD

Annotation. The possibility of using the support vector machine (SVM) for the classification of emotions in human speech is experimentally studied. A variant of the implementation of the classifier (based on SVM) using a linear kernel function is presented. It is shown that the resulting model allows you to classify emotions with an accuracy of up to 85%.

Keywords. support vector machine, SVM, recognition, digital signal processing, machine learning.

ИСПОЛЬЗОВАНИЕ РЕЗЕРВУАРНОГО ПОДХОДА НА ОСНОВЕ ОСЦИЛЛЯТОРА ЧЖУА ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ

Кресс К.Н.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Вашкевич М.И. – доктор тех. наук, доцент

В работе рассматривается практическое применение хаотического осциллятора Чжуа в качестве средства обработки информации и сигналов. В работе показано, что осциллятор Чжуа может быть использован в качестве основы для построения вычислительных устройств, так как ему присущи способность выполнять нелинейное преобразование входных сигналов, зависимость выходного сигнала от истории сигналов на входе и кратковременная память.

Резервуарные вычисления – это метод обработки информации, основанный на использовании динамической системы с большим числом нелинейных элементов, называемой резервуаром. Резервуарные вычисления основаны на идее, что входные данные преобразуются в высокоразмерное состояние резервуара, который представляет собой сеть связанных нелинейных элементов, а затем выходные данные получаются путем линейной комбинации состояний резервуара. Резервуар может быть реализован как аналоговая или цифровая схема, химическая реакция или любая другая система, способная генерировать сложные временные ряды в ответ на внешние воздействия. Резервуарные вычисления позволяют решать задачи классификации, прогнозирования, распознавания и генерации сигналов с минимальным обучением и адаптацией.

Основные требования к резервуарам являются: нелинейность, зависимость выходного сигнала от истории сигналов на входе, кратковременная память. Далее в работе будет показано, что свойства осциллятора Чжуа соответствуют критериям резервуара.

Осциллятор Чжуа — это нелинейная динамическая система, которая может демонстрировать хаотическое поведение и сложные аттракторы. Уравнения движения осциллятора Чжуа имеют вид [1]:

$$\begin{aligned}x' &= \alpha(y - x^3 - c * x) + \delta \\y' &= x - y + z \\z' &= -\beta y\end{aligned}\quad (1),$$

где x, y, z – переменные состояния, α, β, c – параметры, а δ – это сигнал управления. Осциллятор Чжуа может иметь различные режимы колебаний в зависимости от значений параметров и начальных условий.

Для того, чтобы показать нелинейность резервуара по одному из некоторых параметров, можно построить бифуркационные диаграммы состояния от каждого из параметров. На рисунках 1–2 показана зависимость переменной состояния x (и её среднего значения – на рисунках выделено красным) от значений управляющих параметров δ и α .

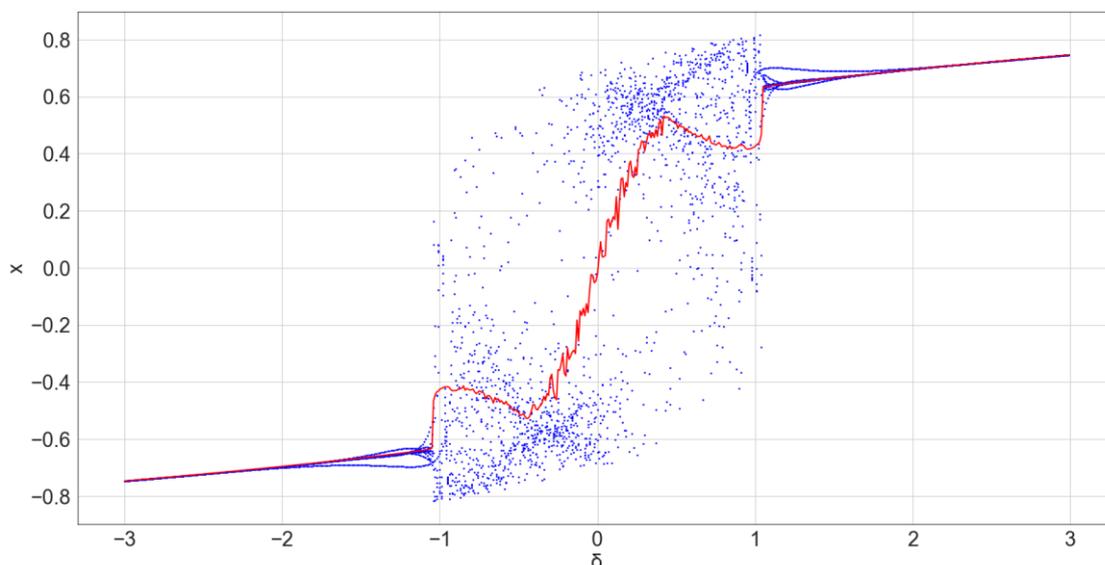


Рисунок 1 – Бифуркационная диаграмма при управлении по δ -параметру

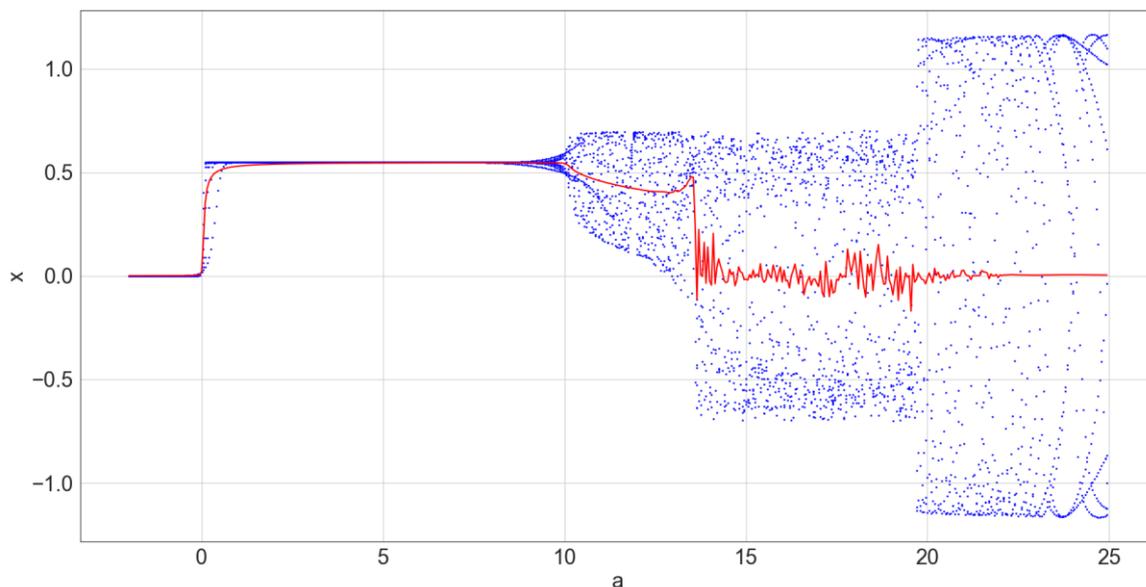


Рисунок 2 - Бифуркационная диаграмма при управлении по α -параметру

Как можно видеть из диаграмм, при переходе системы в хаотичный режим работы видна нелинейная зависимость среднего значения x от значения δ -параметра, в то время как при управлении по α -параметру среднее значение колеблется около нуля без каких-либо видимых закономерностей.

Эффект кратковременной памяти и зависимости выходного сигнала от истории сигналов на входе можно продемонстрировать зависимостью плавающего среднего значения переменной x от различных импульсных сигналов. Для проверки было подано три различных сигнала: без импульса, кратковременный импульс от 20с до 21с, долговременный импульс от 20с до 40с. На рисунке 3 можно видеть, что несмотря на заметные различия значений плавающего среднего в момент подачи импульсных сигналов, начиная с примерно 400с значения становятся малоразличимыми и сходятся к нулю.

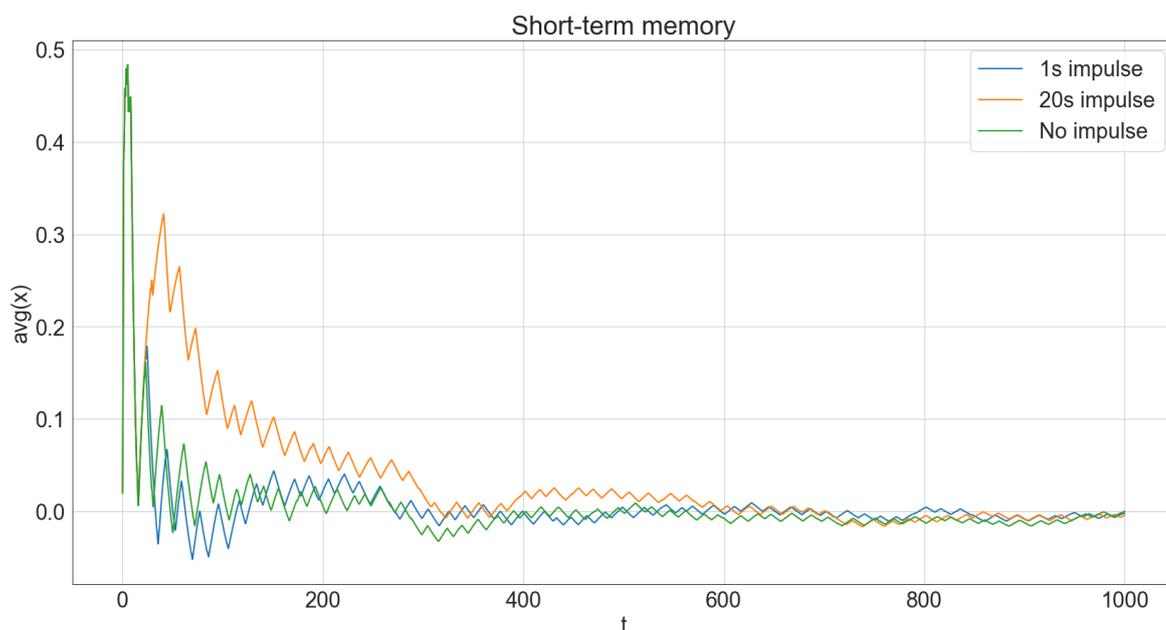


Рисунок 3 – Плавающее среднее при различных импульсных сигналах

Таким образом, исходя из полученных результатов можно сделать вывод, что осциллятор Чжуа может использоваться в качестве резервуара в рамках резервуарного подхода.

Список использованных источников:

1. Крот, А. М. Спектральный анализ хаотических колебаний в имитационной модели схемы Чжуа, разработанной на основе матричной декомпозиции / А. М. Крот, В. А. Сычев // Информатика. – 2019. – Т. 16, № 1. – С. 7–23.

ПОСТРОЕНИЕ ВИДЕОПОТОКА ДЛЯ СИСТЕМЫ РАСПОЗНАВАНИЯ ОБЪЕКТОВ

Мармузевич М.А., магистрант гр.255741

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Качинский М.В. – кандидат технических наук

Аннотация. В докладе рассматривается построение цифрового видеопотока с использованием технологии PYNQ для упрощения разработки и удобства отладки низкоуровневого описания схемы

Задача распознавания объектов является основой систем технического зрения, чаще всего создаваемых посредством реализации алгоритмов предобработки, фильтрации и распознавания на универсальных ПЭВМ (персональных электронных вычислительных машин) с использованием GPU. Это создаёт ограничения, такие как повышенное энергопотребление, массо-габаритные характеристики и высокие требования к производительности системы, для достижения требуемой точности результата. Решением является использование ПЛИС (программируемых логических интегральных схем), для проверки работоспособности используется отладочная плата PYNQ-Z2 [1] с ПЛИС ZYNQ-7020 представленная на рисунке 1.



Рисунок 1 – Отладочная плата PYNQ-Z2

Разработка системы технического зрения начинается с построения цифрового потока данных, для последующей обработки. Интерфейсная часть представляет собой HDMI блок с возможностью подстройки разрешения изображения, путём настройки соответствующих модулей. Для проверки работоспособности использовался пример из стандартной библиотеки PYNQ.lib [2] HDMI-video pipeline. Код на языке python:

```
from pynq.overlays.base import BaseOverlay
from pynq.lib.video import *

base = BaseOverlay("base.bit")
hdmi_in = base.video.hdmi_in
hdmi_out = base.video.hdmi_out
hdmi_in.configure()
hdmi_out.configure(hdmi_in.mode)

hdmi_in.start()
hdmi_out.start()
hdmi_in.tie(hdmi_out)
```

Рассмотрим структуру видеопотока в среде разработки Vivado. Для тактирования работы всех блоков используется сигнал PixelCLK генерируемый блоком конвертирования интерфейса DVI, на основе которого построен HDMI, в формат RGB. Синхросигнал получается путём конвертирования sys_clock частотой 125 МГц в 200 МГц используя блок умножителя частоты Clocking Wizard. Структура входного канала представлена на рисунке 2.

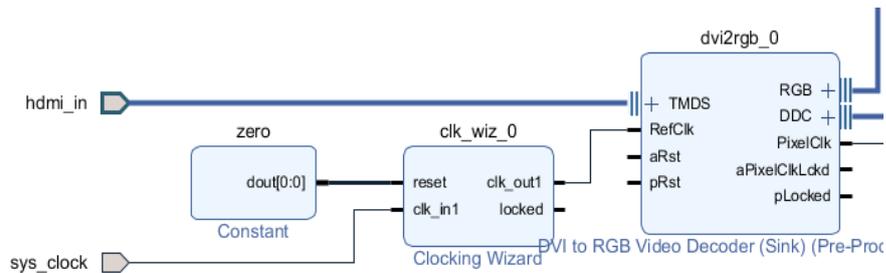


Рисунок 2 – Входной канал видеопотока

Разработанное решение позволяет с минимальной задержкой пропускать видеоизображение. Для последующей обработки видео необходимо кадровать на отдельные картинки, которые возможно беспрепятственно использовать для обработки. Для хранения изображений используется блок VDMA, связываемый с видеопотоком по интерфейсу AXI (Advanced extensible interface). Для подключения блока VDMA необходимо использовать конвертеры интерфейсов RGB2AXI, AXI2RGB, а также Video Timing Controller, для синхронизации вывода изображения при использовании дополнительной обработки. Структурная схема данного решения представлена на рисунке 3.

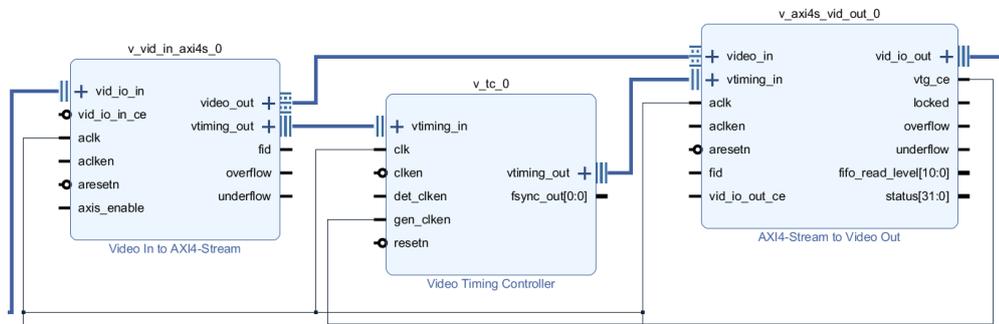


Рисунок 3 – Каскад конвертирования сигнала

На выходе видеопотока используется конвертер из RGB формата в DVI. Блок представлен на рисунке 4.

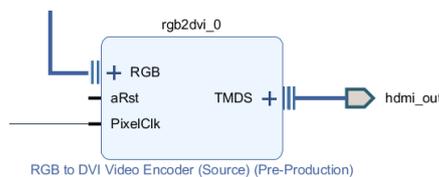


Рисунок 4 – Выходной канал видеопотока

Подводя итог, библиотеки, предоставляемые компанией Xilinx позволяют создавать высокоэффективные аппаратные решения, оптимизированные под семейство ПЛИС самим изготовителем, а также использование плат разработки поддерживающих технологию PYNQ позволяет значительно уменьшить время прототипирования системы и проверить работоспособность отдельных блоков.

Список использованных источников:

1. Support Xilinx [Электронный ресурс]. – Режим доступа: <https://www.xilinx.com/support/university/xup-boards/XUPPYNQ-Z2.html>. – Дата доступа: 25.03.2023.
2. What is PYNQ? [Электронный ресурс]. – Режим доступа: <https://pynq.io>. – Дата доступа: 01.04.2023.

АНАЛИЗ ПОДХОДОВ ДЛЯ ПОСТРОЕНИЯ СИСТЕМ ОБНАРУЖЕНИЯ И РАСПОЗНАВАНИЯ ЛИЦ

Мусский Н.А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Лихачёв Д.С. – доцент, канд. тех. наук

Аннотация. Системы обнаружения и распознавания лиц используются в различных приложениях, от систем безопасности до социальных сетей. Эти системы используют алгоритмы компьютерного зрения для обнаружения и распознавания лиц людей на цифровых изображениях или видео. Алгоритмы работают путем анализа различных черт лица, таких как расстояние между глазами, форма носа и контуры лица. В данной работе рассмотрен общий подход к проектированию системы обнаружения и распознавания лиц, а также сравнение существующих алгоритмов обнаружения и распознавания лиц.

Алгоритмы обнаружения лиц используются для идентификации человеческих лиц на изображении или видеокadre. Алгоритм анализирует изображение на наличие признаков, характерных для лиц, таких как положение глаз, носа и рта, а также других признаков, таких как оттенок кожи и цвет волос. Наиболее часто используемые алгоритмы обнаружения лиц приведены в таблице 1.

Таблица 1 — Алгоритмы обнаружения лиц

Алгоритм	Достоинства	Недостатки
Метод Виолы-Джонса	Быстрый и эффективный, подходит для распознавания лиц в реальном времени.	Может давать ложные срабатывания или пропускать некоторые лица.
Гистограмма направленных градиентов(НОГ)	Точный и эффективный, может обрабатывать окклюзии и изменения в освещении.	Более медленный, чем метод Виолы-Джонса, может потребовать больше вычислительных ресурсов.
Свёрточные нейронные сети (CNN)	Высокая точность и устойчивость к изменениям позы, выражения лица и освещения.	Требует больших объемов данных для обучения и вычислительных ресурсов.
Масштабно-инвариантная трансформация признаков (SIFT)	Устойчив к масштабированию и вращению, хорошо работает с частично закрытыми гранями.	Может быть медленным и может плохо работать с изображениями с низким разрешением.

Алгоритмы распознавания лиц используются для идентификации и проверки личности человека путем анализа его черт лица. Эти алгоритмы обычно используют базу данных известных лиц для сопоставления с лицом, обнаруженным на изображении или видеокadre. Наиболее часто используемые алгоритмы распознавания лиц приведены в таблице 2.

Таблица 2 — Алгоритмы распознавания лиц

Алгоритм	Достоинства	Недостатки
Eigenfaces	Быстрый и простой, требует минимальных вычислительных ресурсов.	Может быть чувствителен к изменениям освещения, выражениям лица и положению.

Fisherfaces	Устойчив к изменениям освещения и выражениям лица, может работать с небольшими обучающими наборами.	Могут возникать проблемы с большими изменениями положения или заслонениями.
Локальные бинарные шаблоны (LBP)	Быстрый и эффективный, работает хорошо с изображениями низкого разрешения.	Может давать ложноположительные результаты или пропускать некоторые лица в сложных сценариях.
Свёрточные нейронные сети (CNN)	Высокая точность и устойчивость к изменениям позы, выражения лица и освещения. наборах.	Требует больших объемов данных для обучения и вычислительных ресурсов.

Создание системы обнаружения и распознавания лиц является сложной задачей. Обобщённый алгоритм построения такой системы выглядит следующим образом:

1. Определение требований к системе обнаружения и распознавания лиц, таких как точность, скорость и доступные ресурсы.
2. Сбор набора данных о лицах для обучения моделей обнаружения и распознавания лиц. Набор данных должен включать лица с разных ракурсов, условия освещения и оттенки кожи.
3. Обработка/нормализация набора данных.
4. Обучение модели обнаружения лиц. Модель должна научиться обнаруживать лица на изображении и выводить местоположение каждого лица.
5. Обучение модели распознавания лиц. Модель должна научиться извлекать черты из лица и выводить уникальные характеристики для каждого лица.
6. Интеграция моделей обнаружения и распознавания лиц с системой, которая принимает изображение или видеопоток в качестве входных данных и выводит личности обнаруженных лиц.
7. Тестирование системы на различных изображениях и видео, оценка точности и производительности. Доработка системы при необходимости.

Наиболее распространенным подходом для обнаружения лиц является использование комбинации каскадов Хаара и алгоритма Виолы-Джонса. Этот метод заключается в сканировании изображения окном определенного размера и формы, которое перемещается по всему изображению, и использовании машинного обучения для определения, содержится ли в окне лицо или нет. Данный подход имеет высокую скорость обработки, устойчивость к изменениям изображения, высокую точность и низкие требования к памяти. Для распознавания лиц наиболее часто используются глубокие подходы на основе сверточных нейронных сетей (CNN). Эти подходы позволяют извлекать сложные признаки из изображений и обучать модели распознавания лиц на большом количестве данных. Кроме того, для обучения CNN используется также механизм обратного распространения ошибки, который позволяет улучшить точность модели. Глубокие подходы на основе машинного обучения показывают более высокую производительность по сравнению с традиционными методами в задачах распознавания лиц. Они также могут обрабатывать изображения с большой вариативностью в углах обзора, освещении и других факторах, что делает их более универсальными и применимыми в различных условиях.

Список использованных источников:

1. OpenCV documentation: https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html
2. OpenCV documentation: <https://docs.opencv.org/2.4/modules/dnn/doc/dnn.html>
3. OpenCV documentation: https://docs.opencv.org/2.4/modules/nonfree/doc/feature_detection.html
4. Face Recognition Using Eigenfaces: <http://www.face-rec.org/algorithms/PCA/jcn.pdf>
5. Fisherfaces for Face Recognition:
https://www.researchgate.net/publication/228996276_Fisherfaces_for_Face_Recognition
6. Face Recognition with Local Binary Patterns: <http://www.face-rec.org/algorithms/LBP/lbp.pdf>

УДК

УСТРОЙСТВО ДЕТЕКТИРОВАНИЯ И ДЕКОДИРОВАНИЯ DTMF-СИГНАЛОВ

Осипов А.С., студент гр.150701

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Вашкевич М.И. – докт. техн. наук

Аннотация: В работе представлена реализация устройства детектирования и декодирования DTMF-сигналов. Для декодирования DTMF-сигналов предложено использовать дискретную систему, которая основана на алгоритме Герцеля. В работе приводится результат моделирования устройства в системе Simulink.

Ключевые слова: Двухтональный сигнал, DTMF-сигнал, декодирование, алгоритм Герцеля, моделирование в Simulink.

Введение

DTMF (тональный набор) – метод кодирования символов в аналоговые сигналы различные по частотному составу с целью дальнейшей передачи по линиям связи. Тональный набор получил широкое применение во время развития первых средств телефонии [1]. Сейчас тональный набор используется в системах умного дома для установления соединения между устройствами, а также для обеспечения ручного ввода команд пользователем [2].

DTMF позволяет закодировать 16 символов используя 8 различных частот. Для кодирования символа требуется сложить две синусоиды различных частот. Частоты выбираются согласно таблице 1, из столбца и строки соответствующего символа.

Таблица 1 – Символы и соответствующие частоты для кодирования методом DTMF

Частота, Гц	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D

Алгоритм расшифровки DTMF-сигналов

Расшифровка двухтональных сигналов осуществляется посредством алгоритма Герцеля [3]. Реализация алгоритма представляет собой рекурсивный фильтр второго порядка и блок извлечения амплитуды, которые характеризуются следующими уравнениями:

$$y(-2) = y(-1) = 0;$$

$$y(n) = x(n) + \alpha \cdot y(n-1) - y(n-2);$$

$$A^2(n) = y^2(n) - \alpha \cdot y(n) \cdot y(n-1) + y^2(n-1),$$

где $x(n)$ – сигнал на входе фильтра в такт работы n ; $y(n)$ – сигнал на выходе фильтра в такт работы n ; $A^2(n)$ – квадрат амплитуды в такт работы n (для целей декодирования достаточно использовать квадрат амплитуды); α – действительный коэффициент фильтра. Коэффициент α является индивидуальным для фильтров разных частот и рассчитывается следующим образом:

$$\alpha = 2 \cdot \cos\left(\frac{2\pi f}{f_s}\right),$$

где f – частота, на которую настроен фильтр; f_s – частота дискретизации входного сигнала.

Структурная схема устройства представлена на рисунке 1. Фильтр настраивается на определенную частоту; таким образом, для детектирования DTMF-сигнала требуется 8 фильтров.

На рисунке 2, а представлен пример работы устройства над сигналом определенной длины; на выходе устройства получаем колеблющийся сигнал с постепенно возрастающей амплитудой. При подаче серии сигналов получим результат, показанный на рисунке 2, б. Из-за рекурсивной структуры фильтра, при отсутствии сигнала на входе, сигнал на выходе сохраняется. Для решения данной проблемы необходима доработка фильтра.

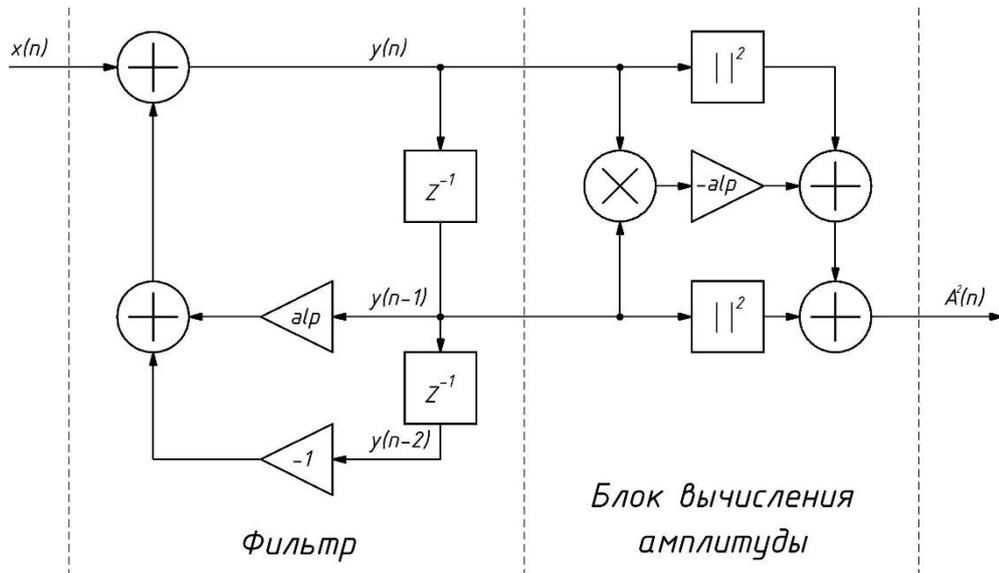
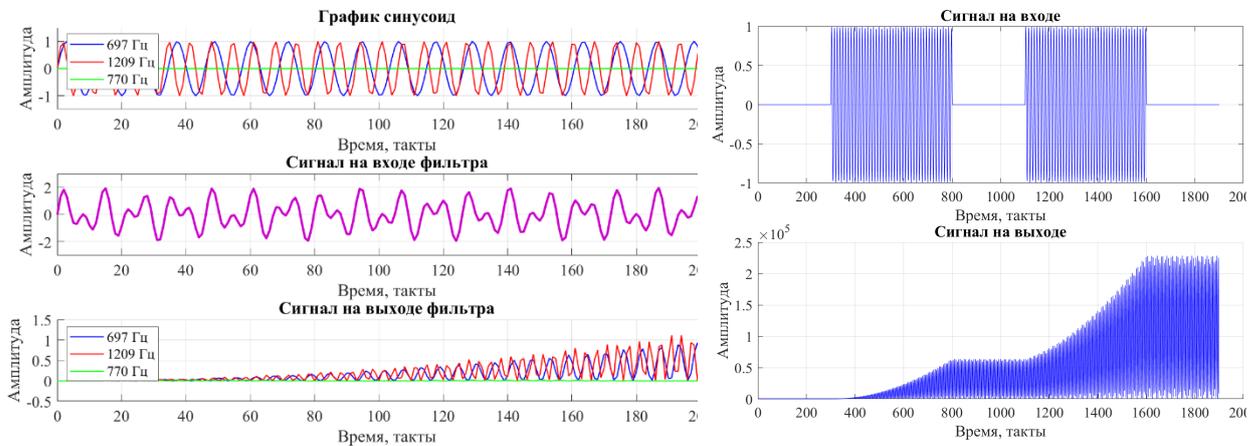


Рисунок 1 – Структурная схема детектора



а

б

Рисунок 2 – Работа фильтра Герцеля при подаче на вход сигнала: а – один сигнал; б – серия сигналов

Модификация фильтра по алгоритму Герцеля состоит из следующих нововведений:

- затухание в рекурсивной части фильтра;
- нормирование сигнала на выходе фильтра;
- обработка выходного сигнала по методу угасающего максимума;
- обработка выходного сигнала по пороговой схеме.

Таким образом, схема устройства принимает новый вид, представленный на рисунке 3. Результат моделирования модифицированного устройства представлен на рисунке 4. Коэффициент затухания $fade$ был взят равным 0,96, коэффициент для метода скользящего максимума – 0,99 и значение порогового срабатывания – 0,8.

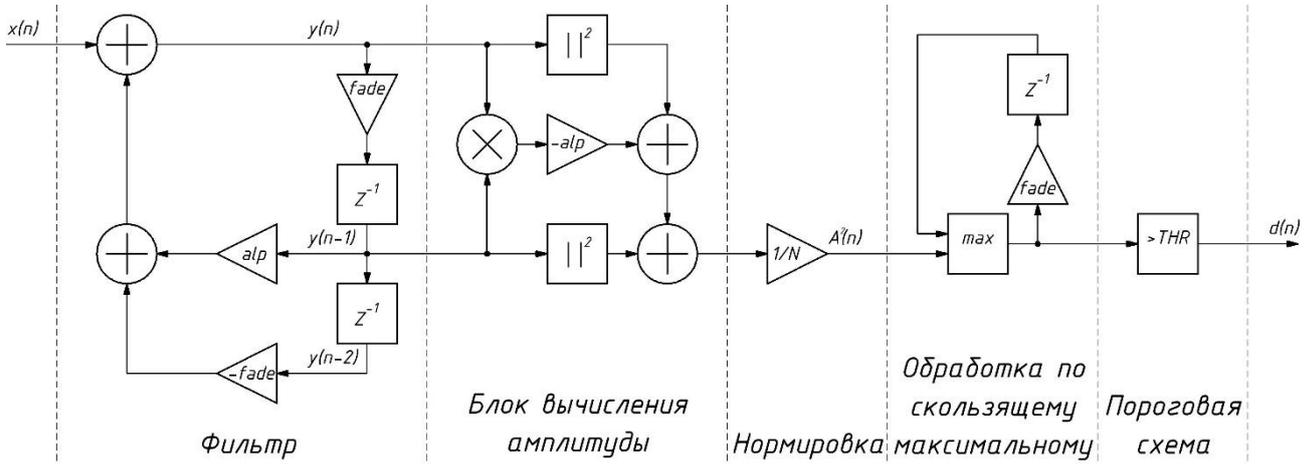


Рисунок 3 – Модифицированная схема фильтра Герцеля

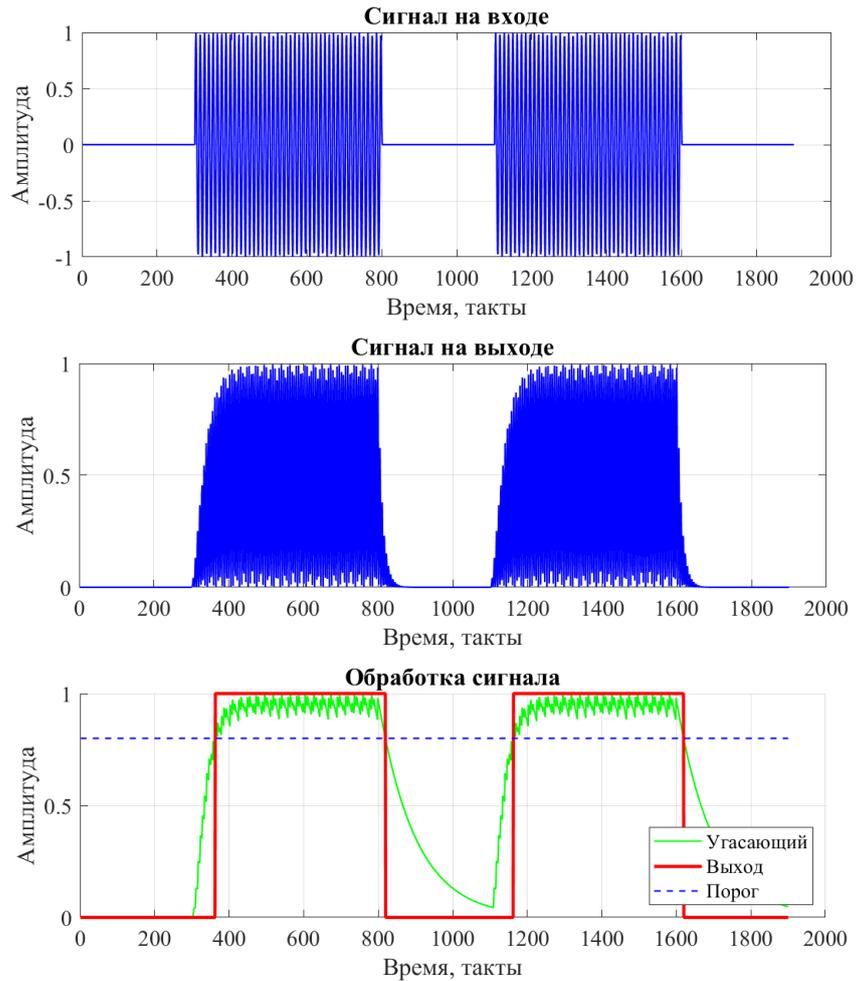


Рисунок 4 – Работа модифицированного фильтра Герцеля

Представленная выше схема была промоделирована в программе Simulink (см. рисунок 5) для каждой из восьми частот с использованием инструмента *Fixed-point Tool*. Параметры фильтров подобраны экспериментальным путем индивидуально для их более корректной работы.

На рисунке 6 представлены графики внутренних сигналов всех восьми фильтров после обработки методом скользящего максимума. Штриховой линией обозначен уровень порога равный единице. Из графиков видно, что сигнал в двух фильтрах превысил уровень порога, следовательно, на их выходах будет образовываться активный уровень сигнала.

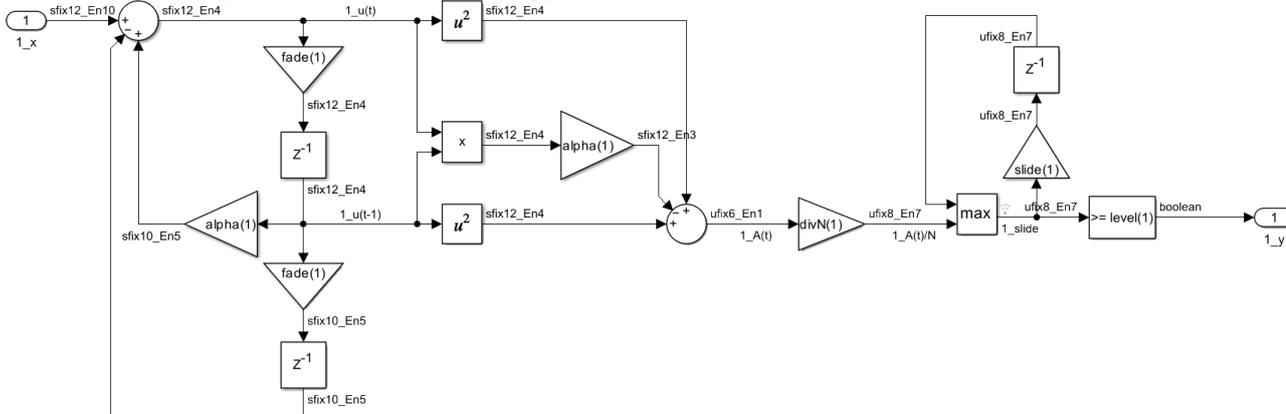


Рисунок 5 – Модель фильтра в среде Simulink

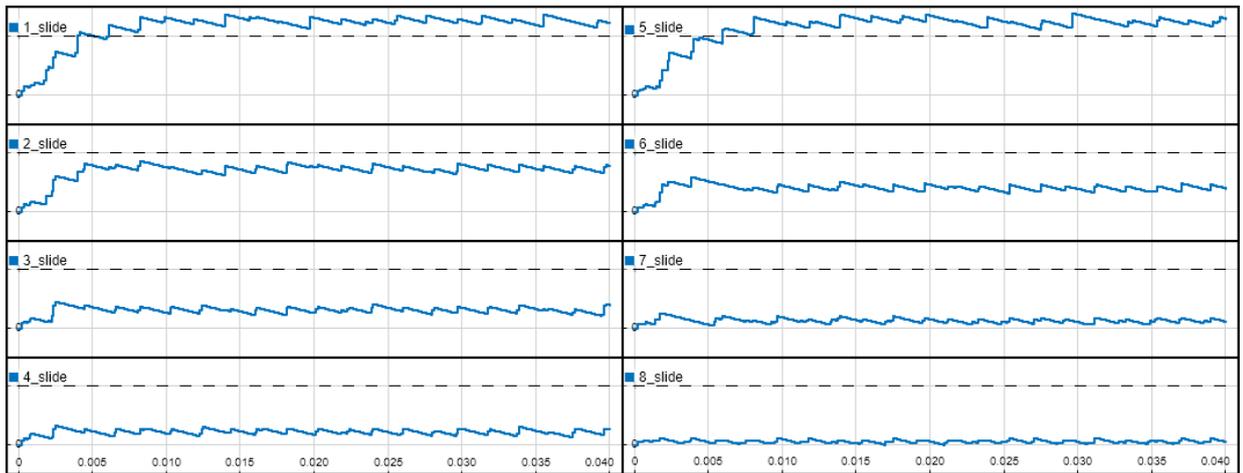


Рисунок 6 – Результат моделирования восьми фильтров

Сигналы с выходов фильтров можно разделить на две группы: сигналы верхних и нижних частот. Каждая группа представляет собой унитарный код из четырех переменных. Для преобразования двух унитарных кодов «1 из 4» в один унитарный код «1 из 16» воспользуемся двумя шифраторами «4 в 2» и одним дешифратором «4 в 16», как представлено на рисунке 7.

Помимо этого, на рисунке изображены способы защиты схемы при поступлении некорректных сигналов от фильтров: два элемента, определяющих равенство количества активных сигналов единице, для каждой группы сигналов; счетчик, предотвращающий прохождение коротких по длительности сигналов на выход схемы (данные сигналы могут образовываться при превышении амплитудой порогового уровня и падении обратно, из-за колебаний сигнала).

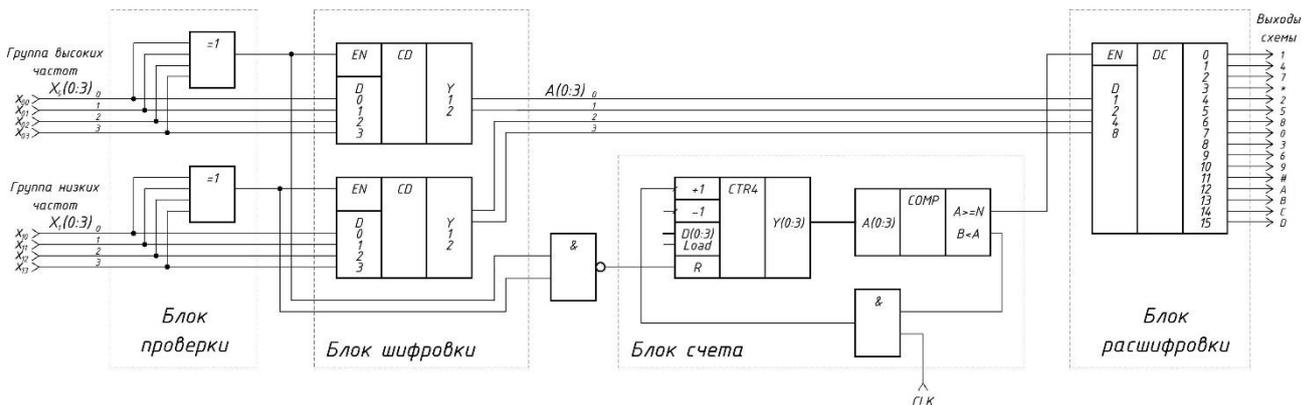


Рисунок 7 – Результат моделирования восьми фильтров

Заключение.

В ходе выполнения научной работы было разработано устройство расшифровки DTMF-сигналов по алгоритму Герцеля. Алгоритм Герцеля был дополнительно модернизирован с целью обеспечения непрерывной обработки сигнала в реальном времени. Экспериментальным путем были подобраны параметры фильтров для корректной работы устройства. Данное устройство было промоделировано в среде MatLab с использованием библиотеки для чисел с фиксированной запятой и в последствии описано на языке описания аппаратуры VHDL.

Список использованных источников:

1. S. L. Gay, J. Hartung, and G. L. Smith, "Algorithms for multi-channel DTMF detection for the WE DSP32 family," *Proc. IEEE Int. Conf. Acoust. Speech Signal Processing*, pp. 1134-1137, May 1989.
2. *Circuit Digest* [Электронный ресурс]. – Режим доступа : <https://circuitdigest.com/electronic-circuits/dtmf-based-home-automation-system>.
3. *Kazus* [Электронный ресурс]. – Режим доступа : <https://kazus.ru/articles/149.html>.

UDC

DTMF SIGNAL DETECTION AND DECODING DEVICE

Osipov A.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vashkevich M.I. – PhD

Annotation. The paper considers realization process of detection and decoding DTMF signals devices. To decode DTMF signals, it is proposed to use a discrete system based on the Goertzel algorithm. The paper presents the result of modeling the device in the Simulink system.

Keywords. Dual-tone signals, Goertzel algorithm.

УДК 004.934

СИСТЕМА МЕДИЦИНСКОЙ ДИАГНОСТИКИ ГОЛОСОВЫХ СИГНАЛОВ НА ОСНОВЕ МЕТОДОВ ОБРАБОТКИ, СИНХРОНИЗИРОВАННЫХ С ЧАСТОТОЙ ОСНОВНОГО ТОНА

Пастернак В. В., студент гр.950702

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Вашкевич М.И. – канд. техн. наук

Аннотация. В работе описывается алгоритм классификации пациентов с боковым амиотрофическим склерозом (ALS) на основе анализа звуковых сигналов, полученных при произнесении продолжительных гласных звуков. Для этого был использован метод вычисления формантных параметров звуковых сигналов, основанных на дискретном преобразовании Фурье и анализе спектральных пиков, соответствующих формантам. Алгоритм был проверен на базе голосовых записей здоровых людей и пациентов с боковым амиотрофическим склерозом. Полученные значения частот и амплитуд формантных пиков были использованы в качестве признаков для классификации пациентов.

Ключевые слова. Боковой амиотрофический склероз, БАС, акустический анализ, форманты, спектральные пики, дискретное преобразование Фурье.

Введение

В настоящее время боковой амиотрофический склероз (ALS) является серьезной проблемой в области неврологии. Он характеризуется постепенной дегенерацией моторных нейронов, что приводит к прогрессирующей слабости мышц и нарушению речи. В связи с этим, разработка эффективных методов диагностики и мониторинга ALS является актуальной задачей.

В последние годы акустический анализ звуковых сигналов, получаемых при произнесении голосовых звуков, стал широко используемым инструментом для оценки функционального состояния голосового аппарата и диагностики ряда заболеваний, включая ALS. Одним из методов анализа является вычисление формантных параметров звуковых сигналов.

Анализ гармонической структуры гласных

Гармоническая структура продолжительных гласных звуков была признана важной и информативной характеристикой для идентификации патологий голоса. Неполное закрытие голосовой щели во время фонации, позволяющее воздуху выходить, является одним из факторов, делающих голос более шумным. В частности, для гласной /a/ это приводит к нарушению гармонической структуры: амплитуда первой гармоники (H1) становится выше второй (H2).

Одним из важных аспектов качества голоса является стабильность гармонической структуры в процессе фонации. Оценка гармонической структуры может рассматриваться как характеристика для описания возбуждающего источника (силы, приводящей к производству звука). Сложность оценки гармонических параметров заключается в том, что они зависят от основной частоты F_0 . В данном исследовании проводился анализ голоса на основе фиксированного количества основных периодов. Было обращено внимание на извлечение среднего значения и стандартного отклонения (SD) первых десяти гармоник гласных. Для заданного голосового сигнала $s(n)$ процесс анализа может быть описан следующими шагами [1]:

1. Заданный голосовой сигнал $s(n)$ разбивается на периоды основного тона с помощью метода сопоставления волновых форм с ограничением фазы (рисунок 1).

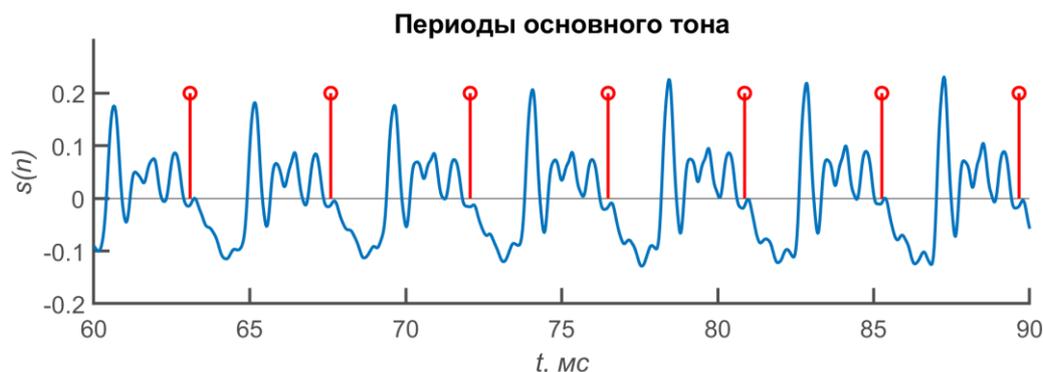


Рисунок 1 – Заданный сигнал с разметкой на периоды основного тона

Метод сопоставления волновых форм использует тот факт, что текущий период связан с предыдущим периодом. Недостатком такого подхода является то, что если возникнет ошибка при обнаружении текущего периода, то это повлияет на все последующие периоды. На практике это приводит к сдвигу фаз, если мы сравниваем циклы, находящиеся друг от друга на расстоянии. Метод сопоставления волновых форм с ограничением фазы избавился от этого недостатка [2].

2. Разделение сигнала $s(n)$ на N_f перекрывающихся кадров, содержащих N_c периодов основного тона с одним периодом перекрытия. Для каждого кадра $s^i(n)$, $i = 1 \dots N_f$ выполняются шаги 3-5.
3. Интерполяция $s^i(n)$ в $I * N_c$ равноудалённых временных точек: $s^i(n) \rightarrow \hat{s}^i(m)$.
4. Применение окна Хэмминга $h(m)$ к интерполированному кадру (рисунок 2) и вычисление дискретного преобразования Фурье (ДПФ) (рисунок 3):

$$\hat{S}^i(k) = \text{ДПФ}[\hat{s}^i(m) h(m)].$$

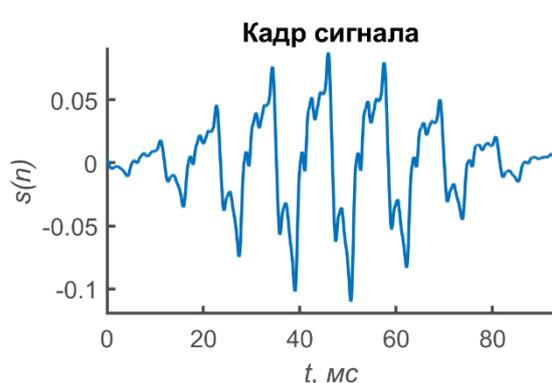


Рисунок 2 – Интерполированный кадр с примененным окном Хэмминга

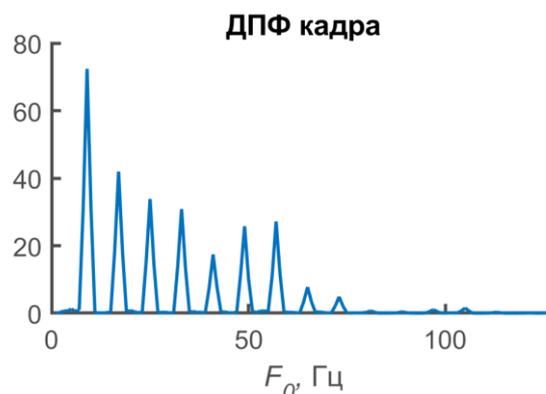


Рисунок 3 – Дискретное преобразование Фурье кадра

5. Извлечение гармонических амплитуд: $h_p(i) = |\hat{S}^i(p * I)|$ $p = 1, 2 \dots 10$.
6. Масштабировать гармонические амплитуды следующим образом:

$$\tilde{H}_p(i) = 20 \log_{10} \left(\frac{h_p(i)}{\max_{p \in [1, 10], i \in [1, N_f]} \{h_p(i)\}} \right).$$

7. Рассчитать среднее значение и стандартное отклонение для масштабированных гармонических амплитуд:

$$Hp^\mu = E\{\tilde{H}_p\}, Hp^\sigma = \sqrt{E\{(\tilde{H}_p - Hp^\mu)^2\}}.$$

8. Рассчитать дополнительный признак – обратное от суммы абсолютных значений Hp^μ и Hp^σ :

$$RelHp = \frac{1}{|Hp^\mu| + Hp^\sigma}.$$

Причина расчёта этого признака в том, что сильные и стабильные гармоники должны иметь низкие отмасштабированные амплитуды Hp^μ и низкие отклонения Hp^σ , а следовательно, высокие значения $RelHp$.

Алгоритм анализа гармонической структуры гласных реализован в *Matlab* следующим образом:

```
function [Hp_mean, Hp_SD, RelHp] =
harmonicAnalysis(fileName)
addpath('Troparion-master/IRAPT/IRAPT_web');
addpath('Troparion-master/Perturbation_analysis');

[s, fs] = audioread(fileName);
[Fo, ~, time_marks] = irapt(s, fs, 'irapt1', 'sustain phonation');

[Fo_periods] = WM_phase_const(s, Fo, time_marks, fs);
```

```

Fo_line = cumsum(Fo_periods);

Nc = 8;
Nf = floor((length(Fo_line) - 1)/(Nc - 1));
l = 512*Nc;

frames = zeros(Nf, l);
ffts = zeros(Nf, l);
window = transpose(hamming(l));
j = 1;
for i = 0:Nc-1:length(Fo_line)-(Nc)
    if i == 0
        temp = s(1:Fo_line(Nc));
    else
        temp = s(Fo_line(i):Fo_line(i+Nc));
    end
    k = linspace(1, length(temp), l);
    temp_interp = interp1(1:length(temp), temp, k);
    filtered = temp_interp .* window;
    frame_fft = fft(filtered);

    ffts(j, :) = abs(frame_fft);
    frames(j, :) = filtered;
    j = j + 1;
end
%% Harmonics
harm = 10;
hp = ffts(:,9:8:(1+8*harm));
Hp = 20*log10(hp/(max(hp, [], 'all')));

%% Rates

Hp_mean = mean(Hp);
Hp_SD = sqrt(mean((Hp - Hp_mean).^2));

RelHp = 1./(abs(Hp_mean) + Hp_SD);

end

```

Описание эксперимента

Для проведения эксперимента использовалась база голосовых записей здоровых людей и пациентов с боковым амиотрофическим склерозом. Результатом прохождения каждой голосовой записи через алгоритм были 30 значений признаков – Hp^μ , Hp^σ и $RelHp$ для каждой из 10 гармоник. В эксперименте использовались значения параметров $N_c = 8$ и $l = 512$.

Результаты эксперимента для голосовых записей здоровых людей и пациентов с БАС представлены на рисунках 4-9.

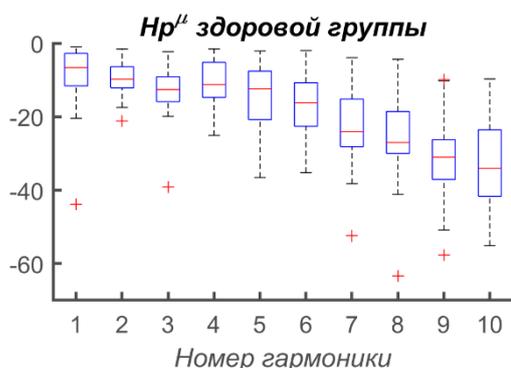


Рисунок 4 – Значения Hr^μ группы здоровых людей

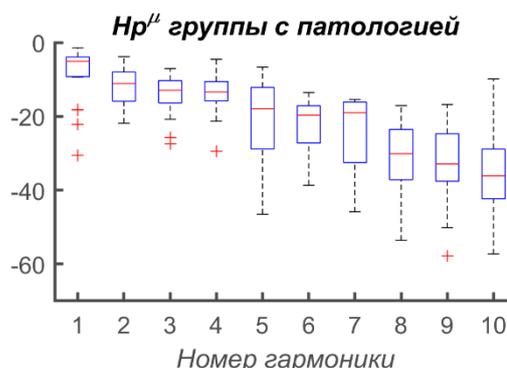


Рисунок 5 – Значения Hr^μ группы пациентов с БАС

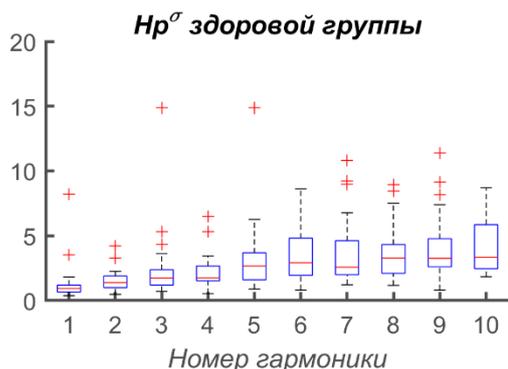


Рисунок 6 – Значения Hr^σ группы здоровых людей

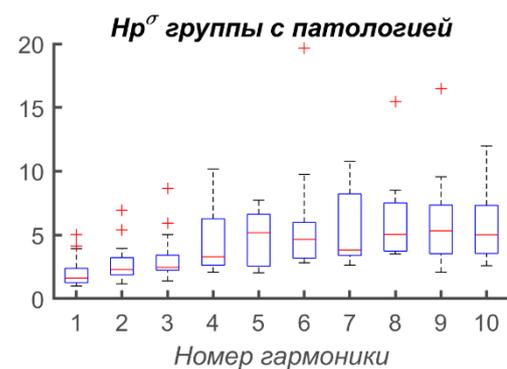


Рисунок 7 – Значения Hr^σ группы пациентов с БАС

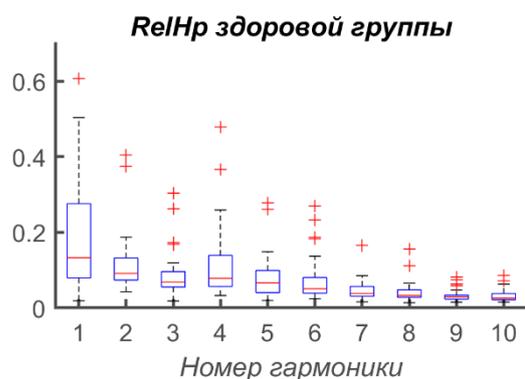


Рисунок 8 – Значения RelHr группы здоровых людей

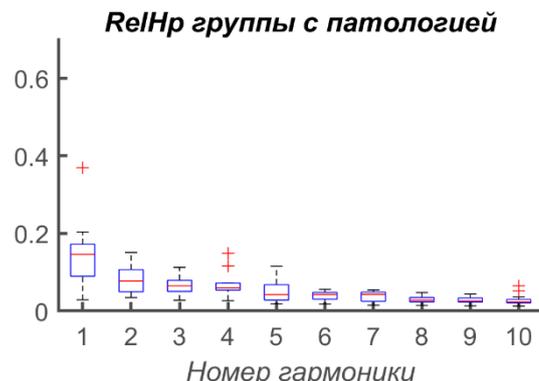


Рисунок 9 – Значения RelHr группы пациентов с БАС

Таким образом, в работе рассмотрена система диагностики голосовых сигналов на основе метода анализа гармонической структуры гласных. Из полученных графиков видно, что полученные признаки гармонических структур гласных у здоровых людей и пациентов с боковым амиотрофическим склерозом несколько отличаются. Это значит, что в дальнейшем эти признаки могут быть использованы методами машинного обучения для диагностики различных заболеваний. Работоспособность метода показана путём MATLAB-моделирования.

Список использованных источников:

1. Classification of ALS patients based on acoustic analysis of sustained vowel phonations / Vashkevich M., Rushkevich Y. // *Biomedical Signal Processing and Control*, 2021. – P. 65, 1-14.
2. M. Vashkevich, A. Petrovsky, Y. Rushkevich, Bulbar ALS detection based on analysis of voice perturbation and vibrato, in: *Proc. of Signal Processing: Algorithms, Architectures, Arrangements, and Applications, SPA, 2019*, pp. 267–272

MEDICAL DIAGNOSTIC SYSTEM BASED ON VOICE SIGNAL PROCESSING METHODS SYNCHRONIZED WITH FUNDAMENTAL FREQUENCY

Pasternak V.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vashkevich M.I. – PhD

Annotation. The paper describes an algorithm for classifying patients with amyotrophic lateral sclerosis (ALS) based on the analysis of sound signals obtained from the pronunciation of sustained vowel sounds. The method used for this was the computation of formant parameters of sound signals based on discrete Fourier transform and analysis of spectral peaks corresponding to formants. The algorithm was tested on a database of voice recordings from healthy individuals and patients with ALS. The obtained values of formant peak frequencies and amplitudes were used as features for patient classification.

Keywords. Amyotrophic lateral sclerosis, ALS, acoustic analysis, formants, spectral peaks, discrete Fourier transform.

АППАРАТНО-ПРОГРАММНЫЙ КОМПЛЕКС ОБНАРУЖЕНИЯ СЛАБЫХ СИГНАЛОВ КВАДРОКОПТЕРА НА БАЗЕ SDR КАНАЛА ПЕРЕДАЧИ ДАННЫХ НА ФОНЕ НЕПРЕДНАМЕРЕННЫХ ПОМЕХ

Стецко В.Ю., студент гр.950702

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Вашкевич М.И. – канд. техн. наук

Аннотация. В работе рассмотрен метод обнаружения сигналов беспилотных летательных аппаратов (БПЛА) – квадрокоптеров с помощью некогерентного накопления. Также рассмотрен метод обнаружения БПЛА в городских условиях (в условиях непреднамеренных помех). Результатом работы этих методов является успешное обнаружение (выделение) видеосигнала БПЛА. Методы проверялись в реальном времени (в прямом эфире) и на тестовых сигналах, которые были записаны с помощью *Software-Defined Radio* от компании *National Instruments USRP 2953 R*.

Ключевые слова. Некогерентное накопление, БПЛА, *USRP*, *LabVIEW*.

Введение

Повсеместное использование малых БПЛА помимо, несомненно, позитивных сторон породило ряд проблем, связанных с неадекватным поведением некоторых владельцев БПЛА, несанкционированным мониторингом объектов и территорий государственной важности, участвовавшими случаями вторжения в личную жизнь, возникновением возможности использованием БПЛА в террористических и разведывательных целях.

Последнее время число беспилотных летательных аппаратов, разнообразие возможных сценариев их применения в военных конфликтах для решения различных задач, и трудность их обнаружения, обусловленная в основном малыми размерами и высотами полета, определили необходимость разработки специальных методов и средств обнаружения БПЛА и борьбы с ними.

Анализ способов и результатов применения БПЛА в военных конфликтах и террористических атаках в последнее десятилетие показывает, что современная эффективно реагирующая на возникающие угрозы система борьбы с БПЛА должна оперативно в автоматическом режиме обеспечивать своевременное обнаружение и идентификацию беспилотных летательных аппаратов.

Методы и способы обнаружения БПЛА по излучениям их бортовых систем

Для связи с БПЛА и высокоскоростной передачи с него телеметрии и данных на наземный пункт управления (ПУ) используется командная радиолиния управления (КРУ).

В настоящее время рынок средств и комплексов радиопротиводействия квадрокоптерам представлен разработками, как зарубежных, так и отечественных производителей. Принципы функционирования известных образцов, в части поиска и обнаружения сигналов радиуправления, базируются на знании частотного диапазона и вида излучаемых сигналов, перечень которых, представлен в таблице 1.

Таблица 1 - Стандартные полосы частот радиоуправляемых БПЛА

Обозначение диапазона	Начало полосы частот	Конец полосы частот	Излучаемая мощность
433 МГц	433,05 МГц	434,79 МГц	10 – 14 дБм
900 МГц	868 МГц	915 МГц	до 21 дБм
2,4 ГГц	2,4 ГГц	2,4835 ГГц	до 26 дБм
5,8 ГГц	5,725 ГГц	5,85 ГГц	до 26 дБм

Представленные в таблице 1 данные показывают, что радиоуправление квадрокоптерами производится в полосе не лицензируемых диапазонов частот (*ISM*) с ограничением максимальной мощности излучения.

Информация для выявления и последующей пеленгации БПЛА может быть получена путем приема специальными средствами отраженной и излучаемой энергии во всех диапазонах спектра электромагнитных и акустических волн. Любому материальному объекту, в том числе и БПЛА, присущи демаскирующие признаки, которые выделяют его в окружающей среде, делая его заметным для наблюдения. В настоящее время существуют основные каналы обнаружения БПЛА.

- Акустический канал. Суммарный спектр акустического излучения тактического БПЛА обусловлен гармоническими и широкополосными составляющими. Он включает в себя гармонические составляющие излучения двигателя, шума оборотов винта, излучение механической природы, а также высокочастотную и низкочастотную составляющие шума двигателя с непрерывными по частоте спектрами.

- Оптический канал. Оптическое обнаружение БПЛА очень сильно зависит от факторов окружающей среды. Увеличение дальности обнаружения достигается за счет сужения поля зрения, уменьшения зоны обзора и увеличения времени поиска. Поэтому визуальные сенсоры являются неэффективными устройствами для проведения поиска. При поступлении внешних целеуказаний от более эффективного поискового средства оптические сенсоры могут быть эффективно использованы для сопровождения БПЛА [1]. Поскольку беспилотники являются значительно меньшими по размерам по сравнению с пилотируемыми средствами, то это усложняет их обнаружение с помощью оптических средств.

- Инфракрасный (ИК) канал. Тепло от БПЛА выделяется, в основном, силовой установкой и, в меньшей мере, электронными компонентами, а также точками торможения на несущих краях крыльев, пропеллеров и винтов. Разработчики беспилотников стараются предотвратить излучение в ИК диапазоне в направлении размещенных на земле приемников и направить это излучение в сторону неба [1]. Кроме того, используют материалы с малой излучательной способностью, такие как серебро и алюминий. В данном случае возможность БПЛА быть обнаруженным определяется его излучательной способностью, контрастом и площадью излучения.

- Радиотехнический канал. Поиск БПЛА с помощью активных радиолокационных станций достаточно продуктивный, так как они имеют относительно большой импульсный объем поиска и значительную дальность обнаружения. Однако РЛС могут быть определены противником по собственному излучению. Радиолокационное обнаружение БПЛА может быть приемлемым только тогда, когда не ставятся требования к скрытности работы или высокой мобильности.

- Канал радиотехнической разведки. БПЛА могут быть обнаруженными и средствами радиотехнической разведки путем приема и анализа радиосигналов линий связи и управления, радиолокационных высотомеров, постановщиков активных помех и радиолокационных станций. Однако этим методом можно установить лишь направление на БПЛА, причем точность определения повышается при увеличении времени наблюдения. Некоторые низкочастотные линии связи могут быть обнаружены на значительных дальностях. Излучение бортовых РЛС и постановка активных помех БПЛА могут быть обнаружены на еще больших дальностях. Этот метод требует минимального оборудования и позволяет быстро определить пеленг цели при дальнейшей выдаче целеуказаний на средства оптического или ИК наблюдения

Общие сведения, описание и принципы работы аппаратной части, применяемой для решения задач обнаружений БПЛА

Используемое аппаратное обеспечение:

- Устройство *USRP 2953 R* от *National Instruments*
- Блок питания *USRP* (входное напряжение: 100-240 В, выходное напряжение: 12 В)
- Сетевая карта *10 Gigabit Ethernet* от *Intel*
- Кабель *Ethernet 10 Gigabit*
- Направленная антенна *HyperLOG 7060* (диапазон частот: 700 МГц – 6 ГГц, коэффициент усиления: 5 дБи)

USRP (Universal Software Radio Peripheral) 2953 R представляет собой интегрированное аппаратно-программное решение для быстрого создания прототипов высокопроизводительных систем беспроводной связи, построенное на архитектуре ввода-вывода (*RIO*). *USRP RIO* позволяет исследователям быстрее создавать прототипы и сокращать время получения результатов.

Основные технические характеристики *USRP 2953 R* [8]:

- Диапазон рабочих частот: 10 МГц – 6 ГГц
- Диапазон усиления: 0 дБ – 37.5 дБ с допустимым шагом в 0.5 дБ
- Максимальная пропускная способность: 160 МГц
- Поддержка *1/10 Gigabit Ethernet*

Применение некогерентного накопления для увеличения дальности обнаружения излучений БПЛА

Принимая во внимание факт применения на всех современных квадрокоптерах радиомодемов, работающих в диапазонах 2,4 и (или) 5,8 ГГц, можно утверждать, что основными для поиска и обнаружения сигналов радиопередачи, являются именно эти диапазоны частот. При этом заявленные дальности, с возможной реализацией алгоритмов автоматического обнаружения и идентификации в этих диапазонах, не превышают полукруга, трех километров [3].

Невысокие показатели обнаружения присущие существующим системам радиомониторинга в первую очередь связаны спецификой самих обнаруживаемых сигналов (скачкообразное изменение частоты от посылки к посылке в пределах диапазона), что вынуждает вести поиск и обнаружение в полосе частот до 120 МГц. Из общей теории радиоприемных устройств известно, что реализация широкополосного приема приводит к увеличению спектральной плотности мощности собственных шумов (N_0) радиоприемного устройства (РПРУ) и снижению его чувствительности. Ориентировочный расчет чувствительности РПРУ в полосе 120 МГц с коэффициентом шума 7 – 10 дБ и отношении сигнал/шум (SNR) 10 дБ свидетельствует, что значение минимальной мощности принимаемого сигнала, будет варьироваться в пределах минус 103 – 106 дБВт. Кроме этого реализация цифровых алгоритмов обнаружения на основе быстрого преобразования Фурье (БПФ), применяемых для решения задачи обнаружения, не позволяет получить существенного выигрыша в отношении SNR согласно формулы

$$SNR = 10 \log_{10} \left(\frac{n}{2} \right) \quad (1)$$

где n – размер выборки (число точек) БПФ.

Для достижения более высоких показателей по дальности обнаружения многие разработчики систем радиомониторинга идут по пути наращивания числа каналов РПРУ, увеличения производительности систем обработки сигнала, повышения направленности приема, автоматизации процессов обнаружения с применением баз данных по типам и видам сигналов радиопередачи.

Повысить возможности существующих систем радиопротиводействия по дальности обнаружения квадрокоптеров возможно за счет совершенствования программных средств обработки сигнала на основе разработки и реализации алгоритмов некогерентного накопления (НКН), учитывающих специфику решаемой задачи.

Особенностью НКН широкополосных, псевдослучайных сигналов с перестройкой по частоте от посылки к посылке по отношению к процедурам межпериодного накопления, применяемых в радиолокации, является векторное суммирование элементов массива, полученного в результате реализации БПФ.

Решение этой задачи позволит компенсировать потери энергии сигнала, возникающие в процессе обработки, тем самым увеличить пороговую чувствительность на 10 – 20 дБ и обеспечить заданную точность измерения частотных параметров сигнала для последующего наведения передатчика помех на обнаруженные источники.

Из [4] известно, что при подаче на амплитудный детектор (АД) смеси полезного сигнала u_c длительностью t_n амплитудой U_m и шума $u_{ш}$ с нулевым средним значением и дисперсией $\sigma_{ш}^2$ при условии $u_{ш}^2 \ll 2U_c u_{ш}$ детектирование производится на участке характеристики детектора близкой к линейной. В этом случае отношение амплитуды сигнала к среднеквадратическому значению шума на выходе АД может быть рассчитано по формуле:

$$\frac{U_{с\text{ Вых}}}{\sqrt{u_{ш}^2}} \approx \frac{U_m^2}{2U_m \sigma_{ш}} = \frac{U_m}{2\sigma_{ш}} \quad (2)$$

Если за время накопления выполняется (2), можно считать, что процесс детектирования будет линейным. В этом случае последетекторное действие НКН заключается в увеличении SNR в N раз

по мощности, где N - количество накопленных выборок БПФ. При условии, что $\sigma_w^2 = N_0 \Delta f_{\text{ПРПУ}}$, расчет порогового отношения SNR на выходе конечного устройства может быть проведен с применением формулы [4]

$$q_n \approx \left(\frac{U_{c \text{ Вых}}}{\sqrt{u_{\text{ш}}^2}} \right)_{\min}^2 = \left(\frac{U_m^2}{4\sigma_{\text{ш}}^2} \right)_{\min} \cdot N = 0.5 \left(\frac{E_{c1}}{N_0} \right)_{\min} \cdot N \quad (3)$$

Учитывая, что реализация НКН производится в частотной области, то энергия одиночного импульса E_{c1} в формуле (3), может быть выражена через размер выборки БПФ с применением формулы

$$E_{c1} = \left(\frac{U_m}{\sqrt{2}} \right)^2 \frac{\Delta t \cdot n}{n_{\phi}} \quad (4)$$

где Δt – шаг дискретизации принятого сигнала;

n_{ϕ} – число фильтров БПФ в полосе частот принятого сигнала.

Из (4) следует, что в случае несогласованного приема энергия одиночного импульса, не зависит от размера выборки БПФ, соответственно значение SNR будет постоянным для любого n .

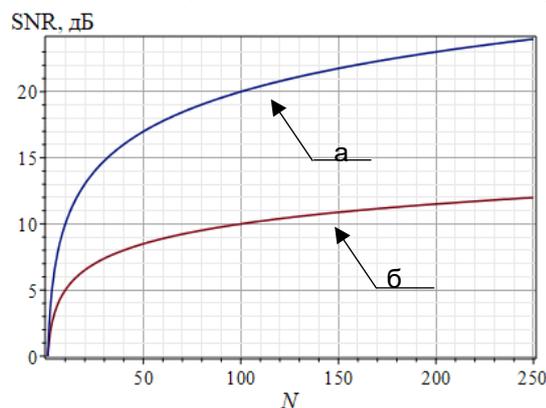
При малой величине SNR можно принять $u_w^2 \gg 2U_c u_w$. В этом случае детектирование является квадратичным [4]. Порогового отношения SNR на выходе оконечного устройства при квадратичном детектировании рассчитывается применением формулы [4]

$$q_n \approx \left(\frac{U_{c \text{ Вых}} \sqrt{N}}{\sqrt{u_{\text{ш}}^2}} \right)_{\min}^2 = 4 \left(\frac{E_{c1}}{N_0} \right)_{\min}^2 \cdot N \quad (5)$$

На основании результата, полученного при решении (5) можно утверждать, что пороговая энергия и мощность, рассчитанные для одной выборки в процессе НКН убывает обратно пропорционально \sqrt{N} .

Если считать, что пороговая величина SNR задана, тогда задача обнаружения сигналов радиуправления, с учетом компенсации потерь на обработку, сводится к определению числа выборок БПФ, позволяющих получить требуемое значение порога обнаружения.

На рисунке 1 представлены зависимости, связывающие SNR с количеством выборок БПФ при линейном и квадратичном накоплении, полученные в соответствии с (3) и (5).



а – линейное накопление; б – квадратичное накопление
Рисунок 1 – Зависимость SNR от числа накопленных выборок БПФ

Анализ зависимостей, представленных на рисунке 1 позволяет утверждать, что для компенсации потерь в 20 дБ необходимо произвести накопление не менее 100 выборок при линейном и около 10000 при квадратичном детектировании.

В свою очередь, зная N , можно определить размер выборки БПФ для заданной частоты оцифровки сигнала (тактовой частоты аналогово-цифрового преобразования) и известной длительности посылки. Например, в случае обнаружения сигнала длительностью миллисекунда при шаге дискретизации 8 нс, и длине выборки БПФ 16382 отсчета получим $N = 8$. Этот результат улучшает SNR при линейном детектировании на 9 дБ (см. рисунок 1) и на 4,5 дБ при квадратичном. При 512-и точечном БПФ $N = 244$, что в свою очередь при реализации НКН улучшает SNR на 24 дБ и 12 дБ для случаев линейного и квадратичного накопления соответственно.

Таким образом разработка и реализация алгоритмов НК при решении задач обнаружения сигналов радиуправления способствует не только получению существенного выигрыша SNR , но и значительному улучшению чувствительности приемника по выходу накопителя, а соответственно увеличению дальности обнаружения излучающих объектов.

Практическая реализация алгоритма обнаружения слабых сигналов БПЛА

1. Приём I/Q сигнала с *USRP 2953 R*, либо чтение бинарного файла, записанного с помощью *USRP 2953 R*, в котором находится I/Q сигнал.

Отрывок исходного I/Q сигнала представлен на рисунке 2.

2. Вычисления энергетического спектра сигнала.

Энергетический спектр характеризует распределение энергии сигнала по частоте. Спектр представлен на рисунке 3.

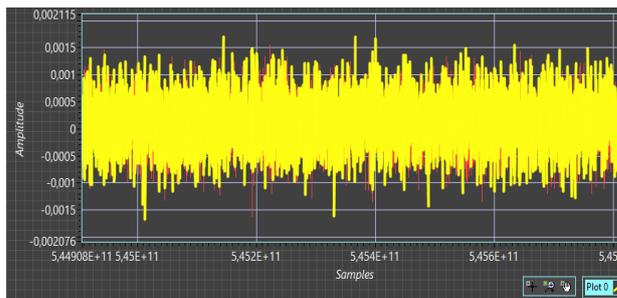


Рисунок 2 – Исходный I/Q сигнал

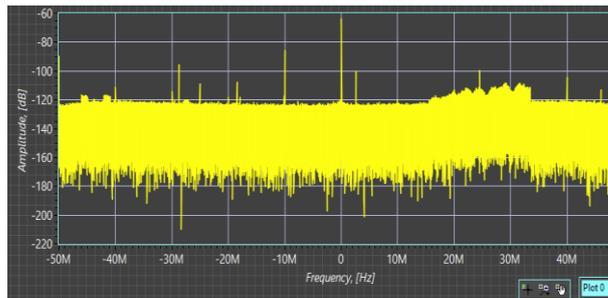


Рисунок 3 – Энергетический спектр I/Q сигнала

3. Разбиваем исходный I/Q сигнал на фреймы по 512 отсчетов.

Так как далее мы будем использовать быстрое преобразование Фурье (БПФ), то длина входных фреймов перед БПФ должна быть кратной двойке, т.е. допустимо провести БПФ с массивом из 2048, 4096, 8192 и т.д. точек сигнала, но не с массивом из 1000, 5000, 10000 и т.д. точек.

Мы используем подход отбрасывания части отсчетов данных так, чтобы длина оставшейся последовательности была равной степени двойки. В данном случае мы теряем часть отсчетов, но это не критично. В случае, если дополнять последовательность нулями, время работы алгоритма существенно увеличивается, что уже является критичной проблемой.

4. Избавление от постоянной составляющей

При спектральном анализе любое постоянное смещение проявляется в частотной области как энергия, сосредоточенная вблизи нулевой частоты, в отсчете спектра $X(0)$. Для N -точечного БПФ значение $X(0)$ пропорционально N и становится недопустимо большим при большом размере БПФ. При построении графика спектра программа масштабирует все отсчеты по большому значению $X(0)$, в результате остальные отсчеты, которые нас интересуют больше, оказываются плохо различимыми [5].

Если обработка выполняется не в реальном масштабе времени, и отсчеты сигнала поступают блоками (последовательностями фиксированной длины) по N отсчетов, то устранить или ослабить постоянную составляющую можно следующим образом:

- вычисляем среднее значение N отсчетов;
- вычитаем это среднее значение из каждого отсчета входного блока;
- получаем новую последовательность, постоянная составляющая которой мала.

5. Вычисление амплитудного спектра

Далее вычисляем амплитудный спектр сигнала – зависимость амплитуд гармоник сигнала от частоты. Вычисление амплитудного спектра производится путем взятия модуля отсчетов БПФ.

Для спектрального анализа сигналов используется алгоритм быстрого преобразования Фурье (БПФ или *FFT – Fast Fourier Transform*).

На рисунке 4 представлен амплитудный спектр сигнала.

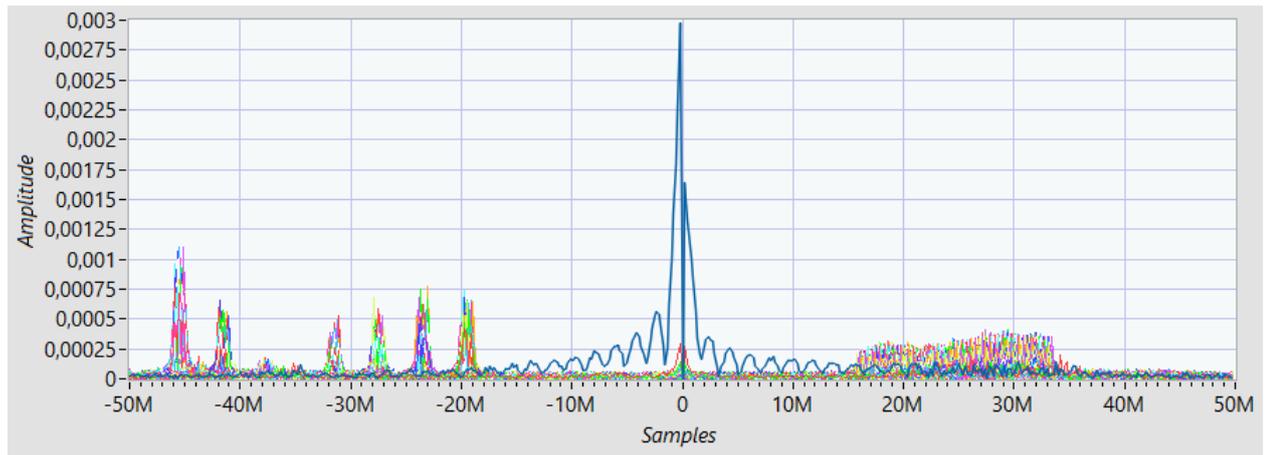


Рисунок 4 – Амплитудный спектр сигнала

6. Некогерентное накопление спектров

Некогерентное накопление (НКН) спектров – это векторное суммирование элементов массива, полученного в результате реализации БПФ.

Данные, которые нам нужны перед выполнением НКН:

- амплитудный спектр, который представлен двумерным массивом;
- время накопления (*accum_time*), которое мы задаем самостоятельно;
- частота дискретизации сигнала (*Fs*);
- количество выборок БПФ (*number_of_samples* = 512).

1) Вычисляем число спектров первичного накопления, будем принимать это значение за переменную *lk*:

$$lk = \text{floor}\left(\frac{\text{accum_time}}{\frac{\text{number_of_samples}}{Fs}}\right) = \text{floor}\left(\frac{\text{accum_time} * Fs}{\text{number_of_samples}}\right) \quad (6)$$

2) Вычисляем общее число накопления для всего двумерного массива, будем принимать это значение за переменную *xk*:

$$xk = \frac{\text{number_of_rows}}{lk} \quad (7)$$

где *number_of_rows* – количество строк входного двумерного массива, полученного в результате реализации БПФ.

3) Создаём цикл от 1 до рассчитанного общего числа накоплений всего двумерного массива (*xk*) – прогоняем двумерный массив блоками, количество спектров которых равняется *lk*.

Внутри цикла суммируем отдельно *lk* спектров, затем отдельно следующие *lk* спектров и т.д. Так повторяется, пока не просуммируются все блоки, т.е. *xk* раз. После проделанных действий получаем двумерный массив, размер которого *xk* строк длиной *number_of_samples* (512).

На рисунке 5 представлен некогерентно накопленный амплитудный спектр.

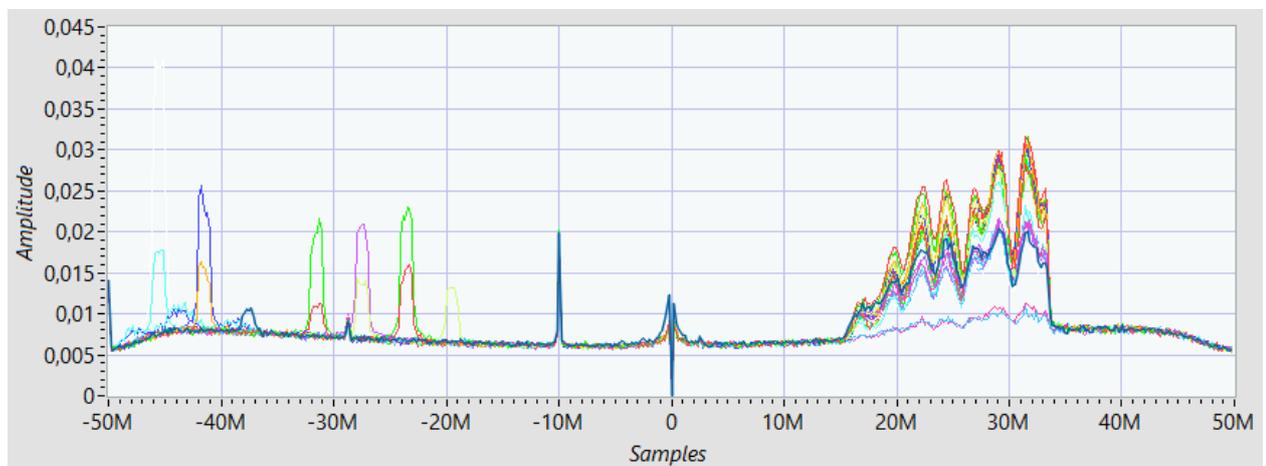


Рисунок 5 – Некогерентно накопленный амплитудный спектр сигнала

7. Фильтрация и расчёт порога

7.1. Фильтрация

После накопления спектров делаем фильтрацию накопленного результата с помощью фильтра Савицкого-Голея (*Savitzky-Golay*). Данный фильтр является одним из наиболее распространенных методов фильтрации сигналов в цифровой обработке сигналов.

Фильтруем некогерентно накопленный сигнал с параметрами фильтра: 3 порядок, количество точек – 27. Результат фильтрации представлен на рисунке 6. Дадим локальное название данной переменной *sig_after_savitzky*.

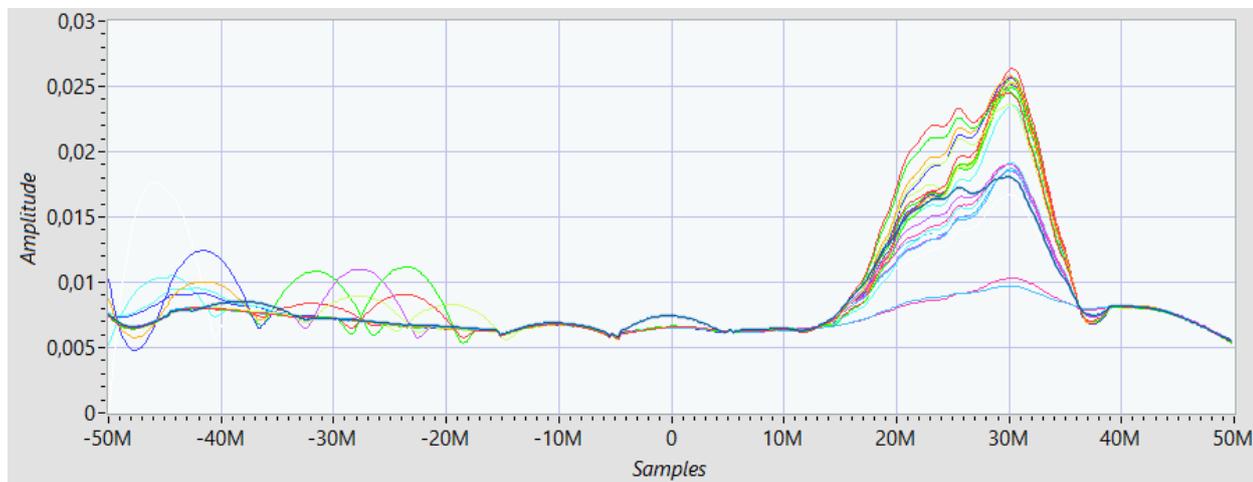


Рисунок 6 – Результат фильтрации фильтром Савицкого-Голея

7.2 Расчёт порога

Для расчёта порога сглаживаем некогерентно накопленный сигнал фильтром нижних частот Чебышева с параметрами: 3 порядок, нижняя частота среза – 100 мГц. Результат фильтрации представлен на рисунке 7.

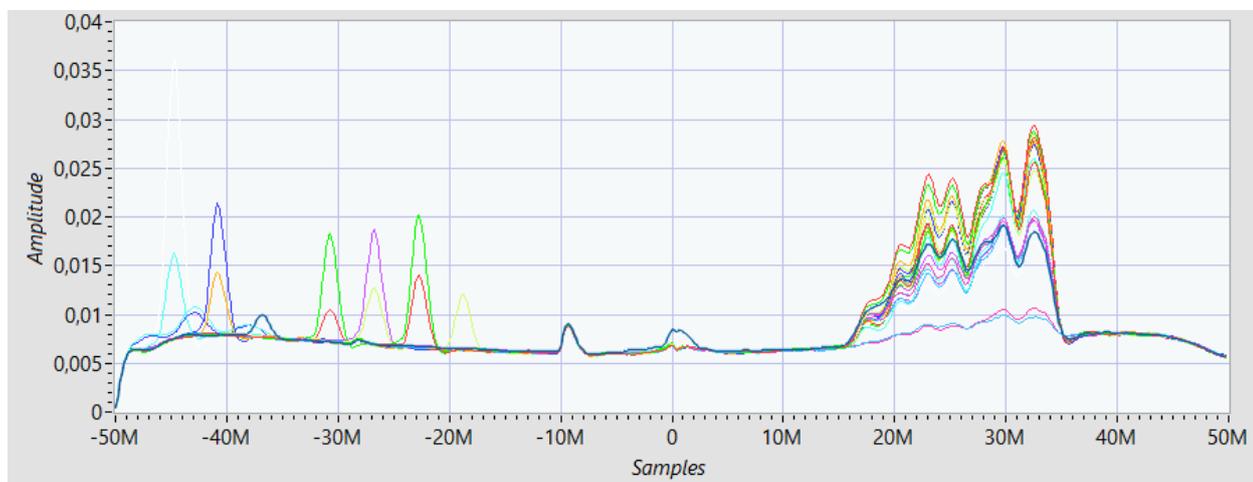


Рисунок 7 – Результат фильтрации фильтром нижних частот Чебышева

Так как результатом фильтрации всё так же является двумерный массив, то для расчёта порога проделываем следующие действия (формула 8):

- 1) Поиск максимальных значений из каждой строки, длина которых 512 отсчетов. Получаем одномерный массив, длина которого равно количеству строк двумерного массива (x_k).
- 2) Поиск минимального значения из этого одномерного массива.
- 3) Регулируем порог, домножая его на 1,4. Получаем конечный порог (a_m).

$$a_m = \min(\max(\text{sig_after_savitzky})) * 1.4 \quad (8)$$

8. Сравнение и зануление ненужных значений.

Сравниваем результат фильтрации фильтром Савицкого-Голея с рассчитанным порогом (a_m). Если значение сигнала меньше порога, то зануляем это значение, иначе оставляем значение таким же.

$$signal = \begin{cases} 0, & sig_after_savitzky \leq a_m \\ sig_after_savitzky, & sig_after_savitzky > a_m \end{cases} \quad (9)$$

9. Суммируем с накоплением (функция *cumsum*) полученный результат (*signal*) и переводим в логарифмический масштаб. Получаем результат работы алгоритма (*result*), который приведен на рисунке 8.

$$result = 20 * \log_{10}(cumsum(signal)) \quad (10)$$

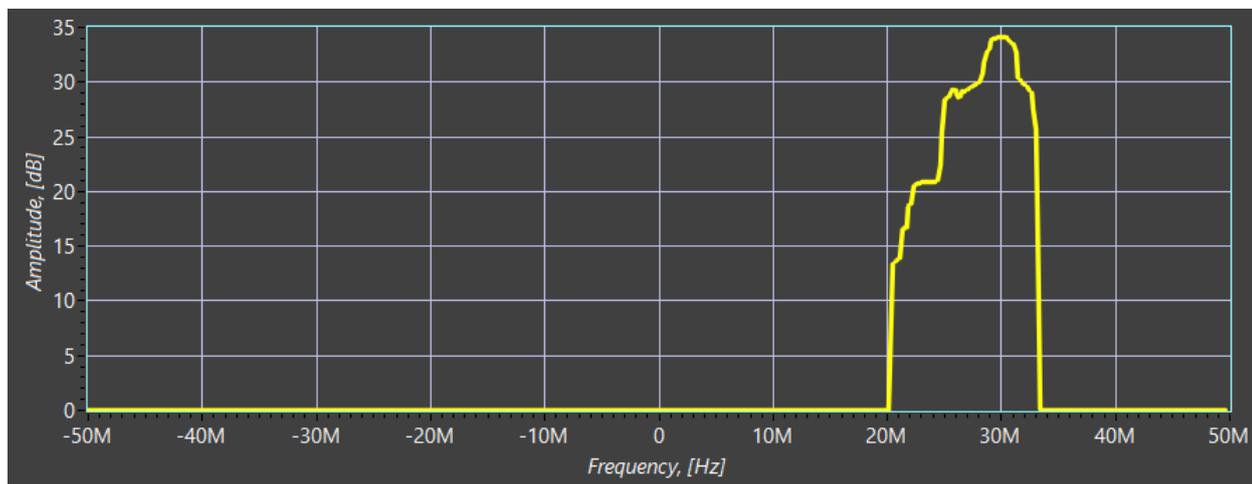


Рисунок 8 – Результат работы алгоритма

Исходя из графика, который показывает результат решения задачи обнаружения с применением некогерентного накопления, мы явно видим сигнал БПЛА, ширина которого около 17 МГц.

Таким образом, полученный результат (рисунок 8) позволяет утверждать, что реализация алгоритмов НКН при решении задач поиска и обнаружения сигналов БПЛА способствует увеличению *SNR* до 20-25 дБ, что в свою очередь значительно улучшает пороговую чувствительность РПРУ, тем самым, повышает возможности существующих систем радиоразведки по дальности обнаружения. Работоспособность алгоритма показана путём *LabVIEW* – моделирования.

Список использованных источников:

1. Соловьев В. А. Проблемы обнаружения беспилотных летательных аппаратов оптикоэлектронными устройствами / В. А. Соловьев // Электронный математический и медикобиологический журнал. – Т. 10, 2011. – Вып. 3. – С. 1–13.
2. National Instruments - Device Specification USRP 2953R, с. 2-4
3. Макушин, М. Системы обнаружения радиоуправляемых БПЛА / М. Макушин // Электроника: НТБ. – 2017. №6 [Электронный ресурс] – Режим доступа: www.electronics.ru. – Дата доступа 03.06.2019.
4. Финкельштейн, М.И. Основы радиолокации / М.И. Финкельштейн Учебник для вузов. – 2-е изд., перераб. и доп. – М.: Радио и связь, 1983. – 536 с.: ил.
5. Ричард Лайонс, Цифровая обработка сигналов: Второе издание, 2006 г. - 656 с.: ил.

UDC

HARDWARE AND SOFTWARE COMPLEX FOR DETECTING WEAK SIGNALS OF A QUADROPTER BASED ON SDR DATA TRANSMISSION CHANNEL AGAINST THE BACKGROUND OF UNINTENDED INTERFERENCE

Stetsko V.Y.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vashkevich M.I. – PhD in Technology

Annotation. The paper considers a method for detecting signals of unmanned aerial vehicles (UAVs) – quadrocopters using incoherent accumulation. The method of detecting UAVs in urban conditions (in conditions of unintended interference) is also considered. The result of these methods is the successful detection (isolation) of the UAV video signal. The methods were tested in real time (live) and on test signals that were recorded using Software-Defined Radio from National Instruments USRP 2953 R.

Keywords. Incoherent accumulation, UAV, USRP, LabVIEW.

РЕАЛИЗАЦИЯ МЕТОДА ГЕНЕРИРОВАНИЯ СИНУСОИДАЛЬНОГО СИГНАЛА В АРИФМЕТИКЕ С ФИКСИРОВАННОЙ ЗАПЯТОЙ

Фатеева И.С.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Вашкевич М.И. – докт. тех. наук

Аннотация. В докладе показана реализация метода генерирования синусоидального сигнала в арифметике с фиксированной запятой на языке Python. Полученные синусоидальные сигналы используются в DTMF-генераторе тональных сигналов.

Синусоидальный сигнал — это математическая кривая для плавного периодического колебания. Генератор синусоиды на основе рекурсивных вычислений выводится из формул синуса и косинуса суммы двух углов:

$$\sin(\alpha + \beta) = \sin \alpha \cdot \cos \beta + \cos \alpha \cdot \sin \beta$$

$$\cos(\alpha + \beta) = \cos \alpha \cdot \cos \beta - \sin \alpha \cdot \sin \beta$$

Данный способ используется для последовательного получения отсчетов одной синусоиды и позволяет вычислить следующий отсчет функции, например:

$$\sin(an) = \sin(a(n-1) + a),$$

через предыдущий $\sin(a(n-1))$. Используя $a(n-1)$ в качестве α и a в качестве β и обозначив $S_1 = \sin(a)$, $C_1 = \cos(a)$, $S(n) = \sin(an)$, $C(n) = \cos(an)$, получим следующее выражение для вычисления очередных значений синуса и косинуса.

$$S(n) = C_1 \cdot S_{n-1} + S_1 \cdot C_{n-1}$$

$$C(n) = C_1 \cdot C_{n-1} - S_1 \cdot S_{n-1}$$

Исходными данными для вычисления первой синусоиды с минимальной возможной частотой являются следующие значения:

$$S(0) = 0; \quad C(0) = 1; \quad a = \frac{2\pi}{N}; \quad S_1 = \sin a; \quad C_1 = \cos a.$$

Генератор синусоидального сигнала функционирует в соответствии со структурной схемой представленной на рисунке 1. Цифровые сигналы S_k и C_k поступают на регистры, а после на сумматоры с предварительным умножением на константы.

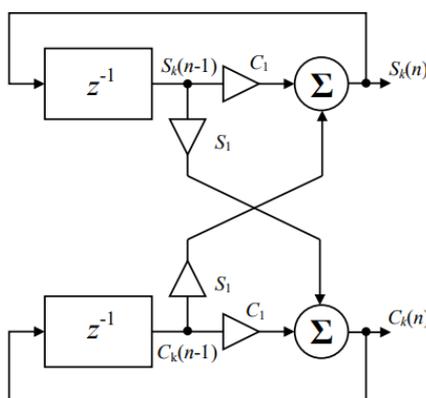


Рисунок 1 – Общая структурная схема генератора сигналов sin и cos

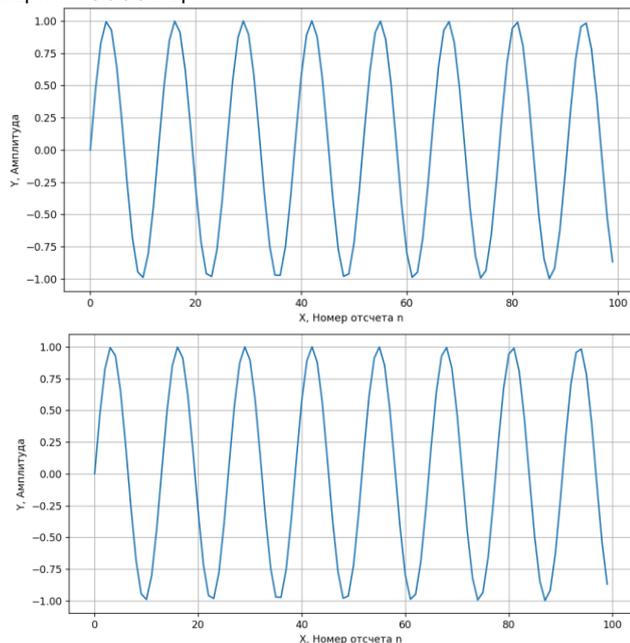
Под термином «число с фиксированной запятой» подразумевается способ представления чисел с фиксированным количеством чисел до и после десятичной точки. При представлении числа с плавающей точкой десятичная точка не имеет определенного места и может «плавать» относительно значащих цифр числа.

Для работы с фиксированной запятой в языке Python используется библиотека `numfi`. Это подкласс библиотеки `numpy` (фундаментальный пакет для научных вычислений). Особенности `numfi` являются:

1. Автоматически выполняет арифметические операции с фиксированной запятой с помощью перегруженных операторов.
2. Совместимость с другими библиотеками, такими как `numpy`.
3. Оптимизированная скорость вычислений.

Объект `numfi` можно использовать как обычный массив `numpy`. Любая арифметическая операция с `numfi` вернет объект `numfi` с заданной точностью.

Пример результата генерирования синусоиды с частотой 697 Гц в арифметике с фиксированной запятой и плавающей представлены на рисунке 2а и 2б соответственно. В данной работе было принято решение взять общую длину слова 24 бита, под дробную часть выделить 22 бита, частота дискретизации – 9000 Гц.

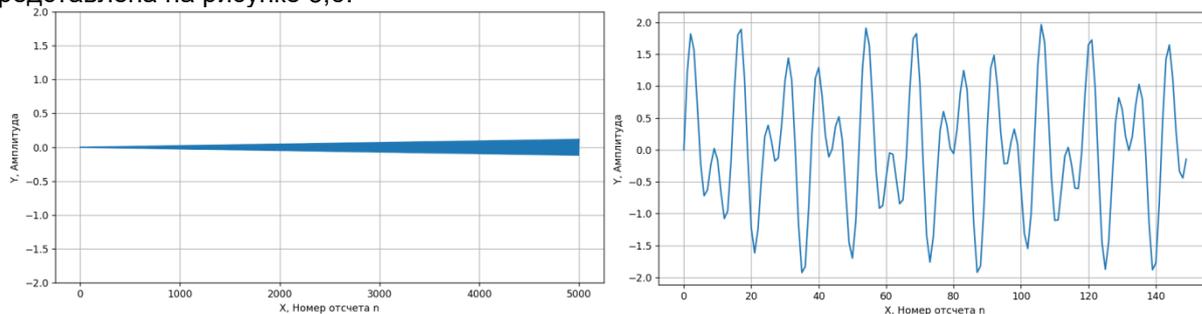


а)

б)

Рисунок 2 – Результаты генерации синусоид: а) с фиксированной запятой, б) с плавающей запятой

В арифметике с фиксированной запятой будет наблюдаться затухание амплитуды колебания со временем. Разница амплитуд между синусоидами с плавающей и фиксированной запятыми представлена на рисунке 3,б.



а)

б)

Рисунок 3 – Сигналы: а) Разница амплитуд между синусоидами с плавающей и фиксированной запятыми; б) DTMF-сигнал, соответствующий «1»

Тональный набор – двухтональный многочастотный аналоговый сигнал, используемый для набора телефонного номера. Сфера применения тональных сигналов: автоматическая телефонная сигнализация между устройствами. Для кодирования символа в DTMF-сигнал необходимо сложить два синусоидальных сигнала. Частоты синусоид берутся из приведенной ниже таблицы из столбца и строки, соответствующих передаваемому символу.

1	2	3	697 Гц
---	---	---	--------

4	5	6	770 Гц
7	8	9	852 Гц
*	0	#	941 Гц
1209 Гц	1336 Гц	1477 Гц	

Таблица 1 – Соответствие частот синусоид и символа

Для генерирования тонального сигнала были взяты следующие данные: частота дискретизации 9000 Гц и частоты синусоид, соответствующие значению «1», заданные в табл.1. Реализация метода генерирования синусоидального сигнала была проведена на языке Python [1]. Результат представлен на рисунке 3,б.

Список использованных источников:

1. Плас Дж. Вандер, *Python для сложных задач: наука о данных и машинное обучение.* – СПб.: ПИТЕР, 2023. – 576 с.

IMPLEMENTATION OF THE METHOD OF GENERATING A SINUSOIDAL SIGNAL IN FIXED-POINT ARITHMETIC

Fateeva I.S.

The report shows the implementation of the method of generating a sinusoidal signal in fixed-point arithmetic in Python. The resulting sinusoidal signals are used in a DTMF tone generator.

BUILDROOT ДЛЯ СИСТЕМ НА СКРИСТАЛЕ СЕМЕЙСТВА ALLWINNER F1CX00S

Ценцезицкий Д.А., магистрант гр.255741

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Шемаров А.И. – канд. техн. наук

Аннотация. Программирование встраиваемых систем является не простой задачей само по себе: ограниченное количество постоянной и оперативной памяти, отсутствие многих аппаратных блоков, малопроизводительные и низкочастотные архитектуры процессорных ядер. При всем при этом написание и отладка программного кода по так называемой baremetal технике доставляет множество неудобств в том числе связанных с отладкой. Для решения данной проблемы чаще всего прибегают к использованию операционных систем, таких как FreeRTOS или mbedOS, однако и они не всегда удобны и функциональны. Если же разработчику необходимо максимально абстрагироваться от аппаратной составляющей, то наилучшим решением будет использование Linux.

Ключевые слова. Linux, Allwinner, UBoot, buildroot, embedded, встраиваемые системы.

Прежде всего необходимо рассмотреть, что делает каждая часть Buildroot. На высоком уровне, рабочий процесс, который Buildroot автоматизирует, представлен на рисунке 1.

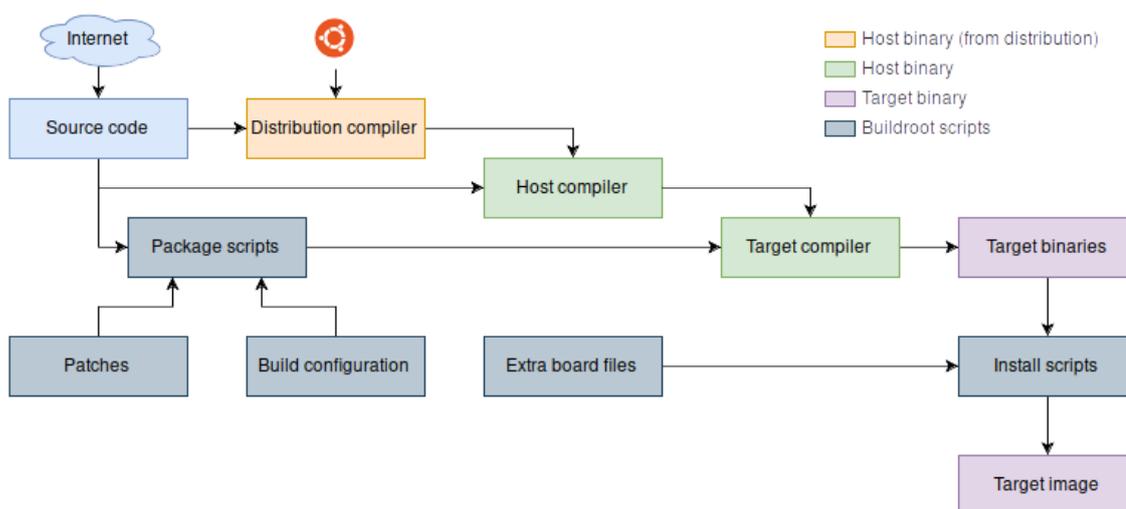


Рисунок 1 – Рабочий процесс автоматизируемый при помощи Buildroot

Типичный процесс сборки образа системы состоит из следующих шагов:

1. Buildroot создает набор инструментов, который представляет собой кросс-компиляторы и другие инструменты, необходимые для компиляции целевой системы (зеленые прямоугольники).
2. Исходный код (синие прямоугольники) для каждой части программного обеспечения загружается из интернета.
3. Используя скрипты Buildroot (серые прямоугольники), исходный код распаковывается, исправляется, настраивается, компилируется и устанавливается в целевой выходной каталог, который формирует корневую файловую систему (rootfs) для целевого объекта (фиолетовые прямоугольники).
4. Дополнительные файлы, такие как файлы конфигурации на устройстве, также копируются в целевой выходной каталог.
5. Наконец, скрипты собирают окончательный образ прошивки из этого корневого файла.

Есть некоторые исключения; иногда кросс-компилятор загружается без необходимости компиляции. Иногда производитель предоставляет целый Board Support Package (BSP), где вся компиляция уже выполнена за нас. Но, в конце концов, это просто сокращает эту блок-схему; все шаги все равно должны быть каким-то образом выполнены.

Наиболее важные каталоги в верхней части дерева Buildroot представлены в таблице 1. Во время своей работы Buildroot генерирует множество других директорий, однако представленные в таблице являются самыми частоиспользуемыми и полезными необходимыми в работе.

Таблица 1 – Результаты тестирования моделей на MS COCO 2017 на устройстве с Intel i7-8550U

Директория	Описание
board/	Файлы и скрипты для поддержки каждой целевой платы
configs/	Конфигурации сборки, такие как suniv-f1c100s_defconfig
package/	Определения пакетов
output/host/	Инструменты сборки, работающие на хотстовой машине
output/target/	Целевой выходной каталог, в котором размещаются целевые двоичные файлы
output/images/	Сюда выводятся образы файловой системы и окончательный образ прошивки

Файл конфигурации suniv-f1c100s_defconfig является одним из основных конфигурационных файлов при сборке системы, он содержит информацию о конкретных версиях ядра и uboot-а, пути к патчам, настройки целевой файловой системы, а так же ссылку на конфигурационный файл uboot.

1.Портирование U-Boot и Linux

Первым шагом при портировании Linux на любую плату является установка и запуск U-Boot.

Из-за ошибки около 70% усилий по программному обеспечению было потрачено на то, чтобы U-Boot общался с флэш-памятью. У U-Boot есть драйвер для работы SPI периферии, но он не работал. Симптом заключался в том, что периферийное устройство SPI могло обнаружить флэш-чип, но всегда считывало с нее мусор. Оказалось, что U-Boot сбрасывает периферийное устройство SPI, когда завершает работу с ним. По-видимому, это не вызвало никаких проблем с другими чипам Allwinner. Однако на F1C100 сброс периферийного устройства, что неудивительно, стирает его регистры конфигурации, особенно те, которые используются для настройки скорости. На рисунке 2 представлен патч устраняющий этот баг.

```

drivers/spi/spi-sunxi.c | 10 ++++++
1 file changed, 9 insertions(+), 1 deletion(-)

diff --git a/drivers/spi/spi-sunxi.c b/drivers/spi/spi-sunxi.c
index dbfeac77eec..2d02289d04d 100644
--- a/drivers/spi/spi-sunxi.c
+++ b/drivers/spi/spi-sunxi.c
@@ -35,6 +35,10 @@

DECLARE_GLOBAL_DATA_PTR;

+/* Forward declarations of some reused functions */
+static int sun4i_spi_set_speed(struct udevice *dev, uint speed);
+static int sun4i_spi_set_mode(struct udevice *dev, uint mode);
+
+/* sun4i spi registers */
#define SUN4I_RXDATA_REG    0x00
#define SUN4I_TXDATA_REG    0x04
@@ -300,7 +304,8 @@ static inline int sun4i_spi_set_clock(struct udevice *dev, bool enable)

static int sun4i_spi_claim_bus(struct udevice *dev)
{
- struct sun4i_spi_priv *priv = dev_get_priv(dev->parent);
+ struct udevice *bus = dev->parent;
+ struct sun4i_spi_priv *priv = dev_get_priv(bus);
    int ret;

    ret = sun4i_spi_set_clock(dev->parent, true);
@@ -317,6 +322,9 @@ static int sun4i_spi_claim_bus(struct udevice *dev)
    setbits_le32(SPI_REG(priv, SPI_TCR), SPI_BIT(priv, SPI_TCR_CS_MANUAL) |
                SPI_BIT(priv, SPI_TCR_CS_ACTIVE_LOW));

+ sun4i_spi_set_speed(bus, priv->freq);
+ sun4i_spi_set_mode(bus, priv->mode);
+
    return 0;
}

```

Рисунок 2 – Патч для работы с SPI flash

После этого патча все заработало. Linux 5.2 загрузился практически без проблем.

2. Создание брза флеш памяти

На отладочной плате установлено 8 МБ флэш-памяти. Она должна вмещать загрузчик, ядро, корневую файловую систему и небольшой постоянный раздел. Я решил использовать UBI для размещения всего, кроме загрузчика, который должен располагаться в самом начале флэш-памяти. Используя genimage, я определяю физические разделы как показано на рисунке 3.

```

image flash.bin {
    flash {}
    flashtype = w25q64
    partition uboot {
        image = "u-boot-sunxi-with-spl.bin"
        size = 256K
    }
    partition rootubi {
        image = root.ubi
        size = 0
    }
}

```

Рисунок 3 – Физические разделы флеш памяти

Затем root.ubi создается из образов разделов как показано на рисунке 4.

```

image root.ubi {
  ubi {}
  partition kernel {
    image = "zImage"
    read-only = true
  }
  partition dtb {
    image = f1c100s.dtb
    read-only = true
  }
  partition root {
    image = "rootfs.squashfs"
    read-only = true
  }
  partition flashdrive {
    image = flashdrive.img
    read-only = true
  }
  partition persist {
    image = persist.ubifs
  }
}

```

Рисунок 4 – Разделы root.ubi

Во время сборки genimage использует это определение для автоматической компиляции двоичного файла, готового к загрузке на чип флеш памяти.

3. Прошивка флеш памяти

Прошивка флеш памяти достаточно проста. При подключении процессора по аппаратному USB к компьютеру происходит следующее: процессор не может найти ничего для загрузки (флеш-память пуста), поэтому он запускает встроенный режим FEL, который позволяет использовать инструмент под названием sunxi-fel для отправки сгенерированного образа по USB прямо во флеш память, выполнив команду «output/host/bin/sunxi-fel -p spiflash-write 0 output/images/flash.bin». Прошивка нескольких мегабайт занимает около двух минут, потому что NOR flash имеет низкую скорость записи - около 100 КБ /с.

Таким образом после загрузки (около 6 сек) мы получаем полноценный Linux готовый к разработке новых программных модулей в знакомом окружении с возможностью удаленной отладки и прочих удобств унифицированной ОС в виде большого количества готовых утилит и драйверов.

Список использованных источников:

1. George Hilliard's blog about embedded systems and software engineering. [Электронный ресурс]. – Режим доступа: <https://www.thirtythreeforty.net/>. – Дата доступа: 12.04.2023.
2. Buildroot. – Режим доступа: <https://buildroot.org/>. – Дата доступа: 12.04.2023.
3. Linux kernel source tree. [Электронный ресурс]. – Режим доступа: <https://github.com/torvalds/linux>. – Дата доступа: 12.04.2023.
4. Das U-Boot Source Tree. [Электронный ресурс]. – Режим доступа: <https://github.com/u-boot/u-boot>. – Дата доступа: 12.04.2023.

UDC

SOFTWARE MODULE FOR PATTERN RECOGNITION FOR MICROCOMPUTERS BASED ON NEURAL NETWORKS OF YOLOV5 FAMILY

Tsentsevitski D.A.¹

Belarusian State University of Informatics and Radioelectronics¹, Minsk, Republic of Belarus

Shemarov A.I. – Candidate of Technical Sciences

Annotation. Programming embedded systems is not an easy task in itself: a limited amount of ROM and RAM, the absence of many hardware units, low-performance and low-frequency architectures of processor cores. With all this, writing and debugging program code using the so-called baremetal technique brings a lot of inconvenience, including those associated with debugging. To solve this problem, they most often resort to using operating systems such as FreeRTOS or mbedOS, however, they are not always convenient and functional. If the developer needs to abstract as much as possible from the hardware component, then the best solution would be to use Linux.

Keywords. Linux, Allwinner, UBoot, buildroot, embedded..

СЕКЦИЯ 3
**«ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

АВТОМАТИЗАЦИЯ РАБОТЫ ДИЗАЙНЕРА ИНТЕРЬЕРОВ С ИСПОЛЬЗОВАНИЕМ ПРОГРАММНОГО СРЕДСТВА НА БАЗЕ СТЕКА ТЕХНОЛОГИЙ ASP.NET CORE И ANGULAR

Фролов А.О.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Сурков Д.А. – ст. преподаватель кафедры ПОИТ

В современных условиях дизайнеры интерьеров сталкиваются с необходимостью управления сложными проектами и большим объемом информации. Для решения этих задач было разработано программное средство на базе технологического стека ASP.NET Core Web API и Angular, которое позволяет управлять временем, проектами и контактами. В результате применения данного программного средства удается значительно упростить процесс управления проектами, сократить временные затраты, а также повысить эффективность взаимодействия дизайнеров интерьеров друг с другом и с контрагентами.

В последние годы профессия дизайнера интерьеров становится все более востребованной, что подтверждается увеличением спроса на услуги дизайнеров интерьеров. Это обусловлено тем, что люди хотят, чтобы их дома, квартиры были красивыми и уютными, но зачастую они не знают, каким образом можно достигнуть этой цели, и поэтому нанимают дизайнеров интерьеров.

В настоящее время дизайнеры интерьеров сталкиваются со многими проблемами, одними из которых являются проблемы управления временем, проектами и контактами. Без преувеличения время имеет решающее значение при управлении проектами. Чем лучше дизайнер интерьеров сможет управлять своим временем и расставлять приоритеты в проектах, тем больше у него шансов привлечь новых клиентов. Правильное управление временем также гарантирует клиентам, что их деньги расходуются эффективно. Управление проектами заключается в планировании, комплектации, бюджетировании, управлении закупками, техническом и авторском надзорах. Грамотная проектная деятельность позволяет достигать запланированных результатов. Построение сети профессиональных контактов способствует поискам новых проектов и продвижению по карьерной лестнице [1].

Большинство существующих аналогов ориентировано не на частных дизайнеров интерьеров, а на целые студии дизайна интерьеров. По этой причине частным дизайнерам интерьеров приходится переплачивать за неиспользуемую ими функциональность приложения, либо вовсе тратить время на поиск программных средств, а затем комбинировать и приспосабливать найденные программные решения под свои нужды.

Предложено создание программного средства, которое обладает рядом функций для уменьшения накладных расходов, связанных с процессами управления проектами, временем и контактами дизайнера интерьеров, ключевыми из которых являются следующие:

- управление проектами;
- управление подрядчиками;
- управление файлами;
- функция календаря;
- функция совместной работы;
- обмен сообщениями и файлами.

В качестве похода к разработке веб-приложения выбрано одностраничное приложение [2]. Клиентская часть программного средства реализована с использованием веб-фреймворка Angular. В качестве основы серверной части приложения лежит веб-фреймворк ASP.NET Core Web API. Взаимодействие между клиентской и серверной частями приложения происходит с использованием REST API. Для сопоставления входящих данных от пользователя была использована библиотека AutoMapper. Для журналирования использовался встроенный поставщик ведения журнала ILogger. В качестве системы управления базами данных была выбрана Microsoft SQL Server. Основным способом взаимодействия с выбранной СУБД является объектно-реляционный модуль Entity Framework Core. Для отправки электронных писем используется сторонний поставщик SendGrid.

Разработанное программное средство позволяет дизайнерам эффективно управлять своей работой, увеличивать производительность и сокращать временные затраты. Кроме того, программное средство способствует повышению эффективности взаимодействия дизайнеров интерьеров друг с другом и с контрагентами.

Список использованных источников:

1. *12 Best Time Management Tips for Interior Designers* [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://foyr.com/learn/time-management-tips-for-interior-designers/>. Дата доступа: 02.04.2023.
2. *Characteristics of Modern Web Applications* [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://learn.microsoft.com/en-us/dot-net/architecture/modern-web-apps-azure/modern-web-applications-characteristics>. Дата доступа: 02.04.2023.

АЛГОРИТМЫ ГЕНЕРАЦИИ ФРАКТАЛЬНЫХ ЛАНДШАФТОВ

Барилко М.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Оношко Д.Е. – старший преподаватель

Произведена сравнительная характеристика алгоритма Diamond-Square и алгоритма с использованием шума Перлина. Сравнение производилось по таким параметрам как: сложность понимания и программной реализации, степень свободы изменения параметров генерации, ограничения алгоритмов.

Процедурная генерация контента – создание при помощи алгоритмов игрового контента с частичным или полностью отсутствующим участием человека. «Игровой контент» – это широкое определение, включающее в себя: уровни, карты, квесты, текстуры, персонажей, растительность, правила, разнообразную динамику и структуры, но не сам игровой движок или поведение не игровых персонажей.[1] Широкое применение получил шум Перлина. Он и его модификации позволяют создавать различные текстуры, анимировать и создавать облака и волны на воде или создавать разнообразные поверхности и ландшафты. В свою очередь алгоритм Diamond-Square, основанный на Midpoint displacement, направлен на генерацию именно поверхностей.

Было выделено четыре параметра, по которым можно сравнить данные алгоритмы:

- сложность понимания;
- сложность алгоритмов;
- количество степеней свободы изменения параметров генерации;
- ограничения алгоритмов.

Сложность понимания оценивалась на основе сложности математического аппарата необходимого для понимания и реализации данного алгоритма.

Реализация алгоритмов была произведена на языке ассемблера, а именно FASM. Для отрисовки конечного результата программы использовался программный интерфейс OpenGL.

Алгоритм Diamond-Square состоит из двух шагов: Diamond и Square. В плане сложности понимания алгоритм очень прост, так как он полностью помещается в один рисунок (см. рис.1). На рисунке показана очерёдность работы с узлами для поверхности 5 на 5. Сначала задаётся случайное значение угловых узлов. По шагу Square находится и задаётся центральный узел квадрата значением среднего арифметического 4 угловых узлов и случайного смещения. Затем шаг Diamond задаёт значение ещё четырём краевым точкам, находящимся на пересечении двух угловых и центральной, используя значение случайного смещения и среднего арифметического узлов, на пересечении которых находится данный узел. Затем шаг Diamond и Square повторяются до момента, когда расстояние между узлами не будет равно единице.

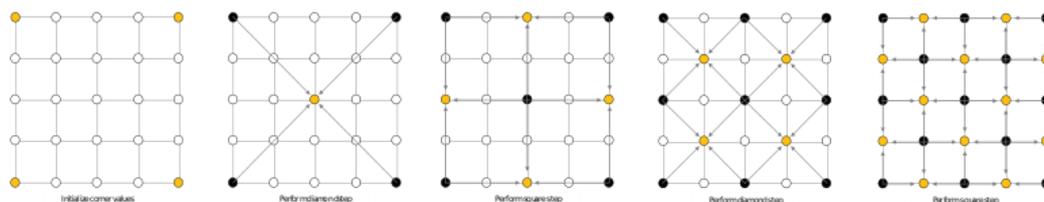


Рисунок 1 – алгоритм Diamond-Square[2]

Сложность данного алгоритма $O(n^2)$, так как нужно обратиться ко всем узлам. Степени свободы, которые влияют на вид поверхности, представлены коэффициентом шероховатости и разрешением. Коэффициент шероховатости отвечает за дельту высот между смежными узлами: чем он больше, тем больше будет разность высот. Разрешение – это количество узлов, которые будут иметь собственную высоту. Ограничения алгоритма – это то, что он подходит только для определённых размеров (длина стороны карты должна быть $2^n + 1$), и то, что после генерации необходима повторная обработка, так как на поверхности появляются пики, которые необходимо сглаживать.

Рассмотрим алгоритм на основе шума Перлина. Для понимания алгоритма необходимы базовые знания линейной алгебры и аналитической геометрии, т.е. требуется знать, что такое векторы и операции над ними и что такое интерполяция. В сравнении с предыдущим алгоритмом, этот требует знания выше школьной программы. Суть алгоритма для двумерной плоскости в том, что задаётся массив значений, представляющий из себя сетку как на рисунке 2, где каждое значение – это случайный двумерный юнит-вектор градиента. После для любой случайной точки из данной сети, причём не обязательно чтобы она лежала на узлах, нужно найти в какой ячейке данной сетки

находится данная точка. Далее для каждого угла необходимо вычислить вектор смещения относительно начала сетки/массива и посчитать скалярное произведение между смещением и вектором произведения и далее интерполировать между 4-мя значениями чтобы найти значение в данной точке.

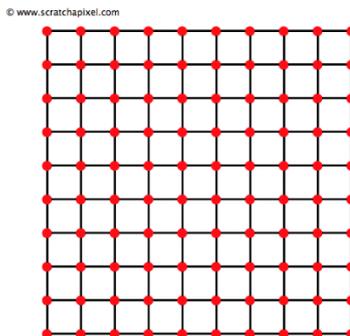
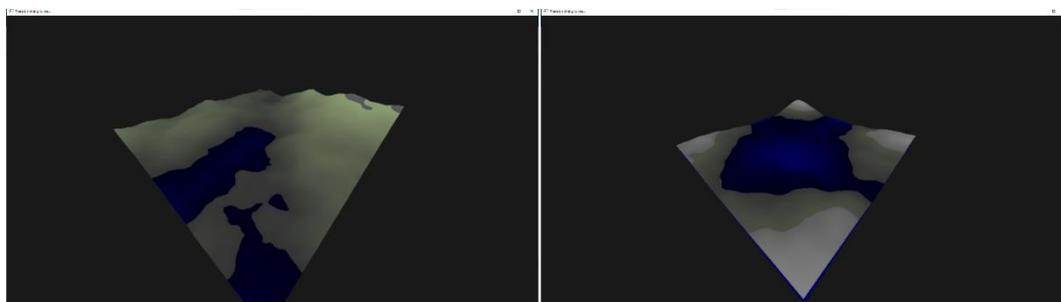


Рисунок 2 – представление массива в виде [3]

Сложность данного алгоритма $O(n^2)$ (Необходимо пройти по всем узлам, а для каждого узла сложность константная), однако для каждого узла необходимо рассчитать и сложить значения нескольких октав, чтобы получить поверхность, напоминающую ландшафт. Тогда можно ввести m – количество октав, и сложность алгоритма можно записать как $O(mn^2)$. Октавами называют функции шума Перлина, у которых частота различается в два раза. Степеней свободы у алгоритма множество и количество зависит от разрешения. Во-первых, на генерацию влияет количество октав, каждая октава добавляет шероховатость поверхности. Во-вторых, для каждой октавы может быть определён множитель, который влияет на то, насколько сильно октава будет выражена в конечной поверхности. Ограничениями данного алгоритма является память, ведь кроме самой поверхности необходимо хранить векторы-градиенты для каждой из октав. Однако в подготовленной реализации расход памяти был исправлен, при помощи функции генерации псевдослучайных чисел по двум координатам, которая позволила генерировать единичный вектор для каждой точки, в место того, чтобы хранить его в заранее подготовленном массиве. Таким же образом было встроено *зерно* для генерации, позволяющее переносить сгенерированный ландшафт, сохраняя только *зерно*, и, предположим, на другой машине, используя всё то же *зерно* получить идентичный ландшафт.

Следует отметить, что, несмотря на отличия двух алгоритмов, конечный результат у них схож (см. рис. 3 и 4). Тем не менее алгоритм Diamond-Square может позволить сгенерировать ландшафт без каких-либо углублённых знаний. В любом случае оба этих алгоритма подходят в учебных целях. С другой стороны, алгоритм на основе шума Перлина имеет значительное преимущество благодаря его степеням свободы. Это позволяет генерировать более разнообразные вариации поверхностей чем в предыдущем алгоритме. Также он имеет потенциал к модификации. Таким образом алгоритм на основанный на шуме Перлина можно сравнить с каркасом, на который можно и нужно навешивать разнообразные модификации, когда как алгоритм Diamond-Square не поддаётся модификациям столь же просто, однако представляет законченный алгоритм полностью выполняющий своё назначение.



Рисунки 3 и 4 – результаты работы алгоритма Diamond-Square и алгоритма основанного на шуме Перлина соответственно

Список использованных источников:

1. *Procedural content generation* [Электронный ресурс]. — Режим доступа: https://www.researchgate.net/publication/285878527_Procedural_content_generation_goals_challenges_and_actionable_steps — Дата доступа: 07.03.2023.
2. *Diamond-square algorithm* [Электронный ресурс].—Режим доступа: https://en.wikipedia.org/wiki/Diamond-square_algorithm. — Дата доступа: 09.03.2023.
3. *Scratchapixel* [Электронный ресурс]. — Режим доступа: <https://www.scratchapixel.com/lessons/procedural-generation-virtual-worlds/procedural-patterns-noise-part-1/introduction.html>. — Дата доступа: 15.03.2023.

АППАРАТНО-ПРОГРАММНОЕ СРЕДСТВО ТЕСТИРОВАНИЯ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ

Ахмед А.Н.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Иванюк А.А. – д-р техн. наук, профессор

В тексте доклада описывается программное средство тестирования физически неклонлируемых функций. Описываются экспериментальная установка, процесс сбора данных, структура хранимых данных и способ анализа получаемых данных.

Физически неклонлируемая функция — это аппаратное устройство, экземпляры которого имеют ряд уникальных параметров и характеристик, которые можно применить для генерации пар “запрос-ответ”, с помощью которых можно идентифицировать устройство. Использование ФНФ в криптографии основано на том, что практически невозможно физически скопировать ФНФ и сложно предсказать ответ на определенный запрос.

В силу изменения температуры окружающей среды, а также неизбежного износа и деградации кристалла интегральной схемы, генерируемые ФНФ идентификаторы являются нестабильными. С другой стороны, ФНФ являются хорошим источником случайности для построения на их основе генераторов случайных числовых последовательностей, однако их вероятностные характеристики не всегда соответствуют криптографическим стандартам. В свою очередь, увеличение стабильности ФНФ приводит к уменьшению ее случайности, что способствует уязвимости к криптографическим атакам. Для определения особенностей конкретной ФНФ необходимо вычислить следующие характеристики [3]: случайность (randomness), устойчивость (steadiness), правильность (correctness), диффузия (diffuseness), уникальность (uniqueness). При тестировании любого нового устройства ФНФ необходимо разработать протокол общения с ФНФ, определиться, как и в каком формате будут храниться собранные во время теста данные, выбрать инструмент для анализа полученных данных и визуализации результатов.

Разрабатываемое программное средство тестирования ФНФ должно решать задачу автоматизации вышеописанных действий.

Процесс тестирования можно разделить на два обособленных подпроцесса. Первый подпроцесс — сбор и хранение данных, второй — анализ данных.

На рисунке 1 представлена структура экспериментальной установки, потоки данных между хостом и установкой и внутри установки, которую будем использовать для тестирования. ФНФ примем за черный ящик, программному средству будет необходимо предоставить информацию о разрядности запросов CH (challenge) и ответов R (response). Тестируемые ФНФ чаще всего реализуют на программируемых логических интегральных схемы (FPGA). Это позволяет нам на той же ИС, на которой реализована ФНФ, реализовать и интерфейс для доступа к ФНФ.

При разработке данного программного средства будут проанализированы ФНФ, реализованные на FPGA Zybo Z7 и Nexys 4 от компании Digilent.

Для того чтобы программное средство могло отправлять запросы на генерацию данных и получать поток сгенерированных ответов, необходимо реализовать на встроенной процессорной системе генератор запросов и модуль, который будет получать и отправлять данные по UART. Для разработки на Zybo Z7 воспользуемся IP-ядром ZYNQ7 Processing System, предназначенным для объединения процессора ARM Cortex-A9 с программируемой логикой FPGA. В свою очередь, интерфейс на Nexys 4 разработаем с использованием MicroBlaze — семейства 32-разрядных микропроцессорных ядер, реализуемых на основе FPGA фирмы Xilinx. Использование данных IP-ядер даст возможность, вместо разработки и реализации конечного автомата, использовать программу, написанную на языке C, которая будет исполняться встроенным процессором.

Общение с FPGA будет происходить с использованием последовательного порта (COM-порта). Выбор данного интерфейса обусловлен простотой этого интерфейса с физической точки зрения, тем, что на FPGA можно воспользоваться уже реализованным интерфейсом UART и на любом компьютере USB-порт может использоваться для эмуляции COM-порта. Количество подключаемых устройств можно будет увеличить, подключив USB-хаб.

Для генерации запросов будет использован генератор M-последовательностей, так как с его помощью можно сгенерировать псевдослучайную двоичную последовательность. Данный вид генераторов является периодическим с периодом $N = 2^n - 1$, где n — длина регистра, с помощью которого формируются значения. Любые комбинации символов длины n на длине одного периода M-последовательности встречаются не более одного раза. В общем для тестирования подойдет генератор любой последовательности, состоящей из уникальных запросов CH.

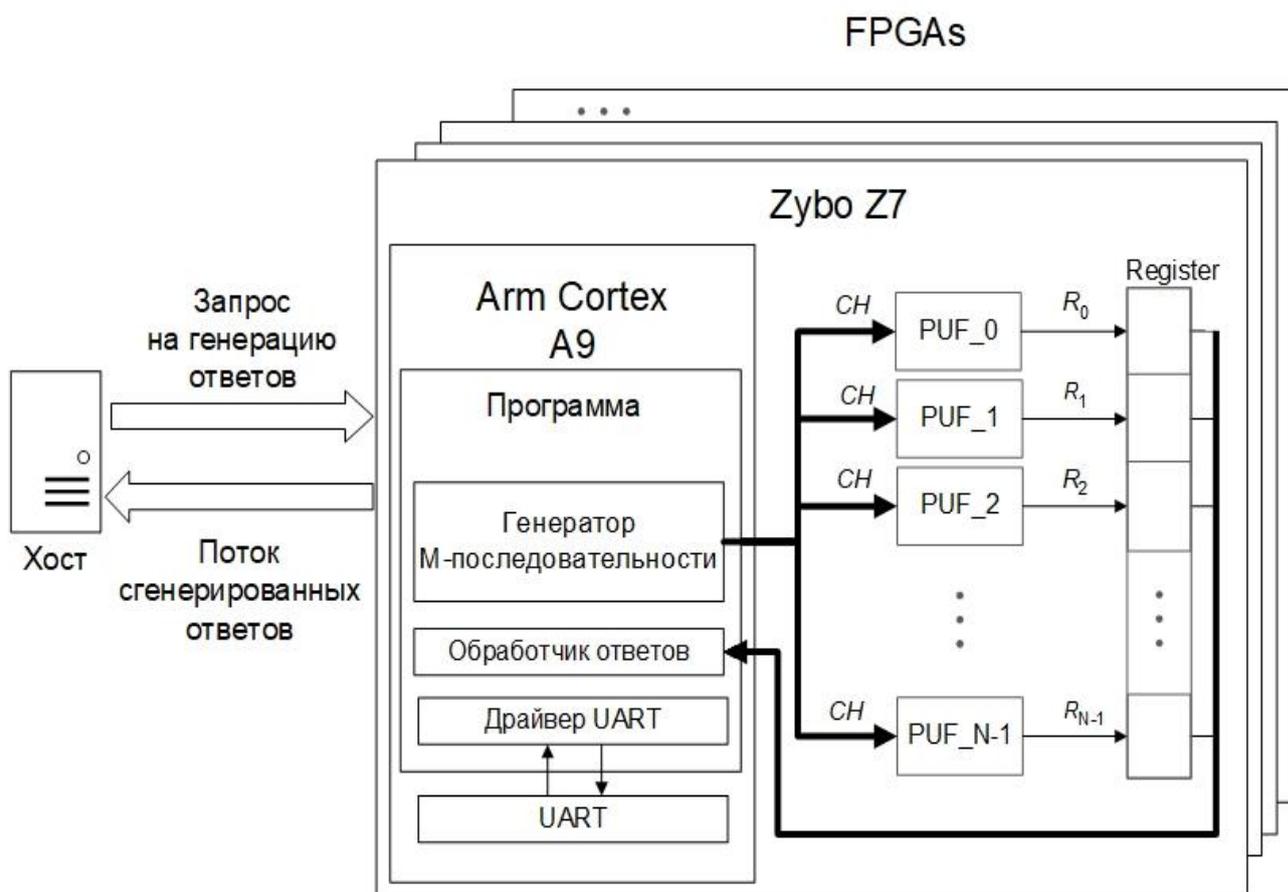


Рисунок 1 – Потоки данных в экспериментальной установке

Полученные ответы будут храниться в файлах. Метаданные об ответах, позволяющие их идентифицировать, будут храниться в базе данных SQLite, так как она не требует от пользователя установки или запуска в контейнере. В ней будут храниться таблицы с информацией об тестируемых ФНФ, названиями её экземпляров, реализованных на FPGA, названиями тестов и дополнительной информацией о них, путями к файлам с ответами.

В файлах будут последовательно записываться ответы, без соответствующего им запроса, это позволит экономить пространство на диске и время на чтение из файла данных и записи данных в файл. Кроме того, вычисление характеристик происходит с использованием порядковых номеров запросов. Ответы экземпляра ФНФ на последовательность запросов хранятся в одной последовательности. Порядок ответов соответствует порядку запросов, из которых они были получены. Для чтения данных из файла необходимо только знать длину ответа в байтах.

В качестве языка разработки программного средства выбран Python, так как он предоставляет широкий набор библиотек для работы со статистикой и числами.

Как отмечается в выводе статьи [1]: для вычисления тех же характеристик надежности и непредсказуемости ФНФ существует множество различных формул. Некоторые из этих формул могут быть алгебраически преобразованы друг в друга, что означает, что они несут одинаковую информацию о характеристике ФНФ, несмотря на разные числовые результаты.

В связи с этим, для гибкости анализа полученных данных, пользователь вместо того, чтобы использовать жестко закодированную логику обработки данных, будет описывать логику обработки математическими выражениями. Таким образом в программное средство будет частично реализовываться функционал таких систем, как Mathcad или Mathematica.

Аппаратно-программное средство тестирования ФНФ значительно повысит эффективность тестирования за счет автоматизации рутинных процессов, что позволит сократить время и улучшить качество работы. А описание логики обработки данных с использованием математических выражений позволит более эффективно модифицировать эту логику.

Список использованных источников:

1. Florian, K. A. Wilde. *Metrics for Physical Unclonable Functions* / Florian K. A. Wilde – Munich, Technical University of Munich, 2021 – P. 16
2. Maes, R. *Physically Unclonable Functions: Constructions, Properties and Applications. PhD thesis* / Maes, R – Belgien. Katholieke Universiteit Leuven, 2012.

3. Y. Hori, T. Yoshida, T. Katashita and A. Satoh, "Quantitative and Statistical Performance Evaluation of Arbitrarily Physical Unclonable Functions on FPGAs," 2010 / Y. Hori, T. Yoshida, T. Katashita and A. Satoh – International Conference on Reconfigurable Computing and FPGAs, Cancun, Mexico, 2010

БАЗОВЫЙ МОДУЛЬ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ МИКРОНТРОЛЛЕРА STM32

Авсяник Е.С., Деменковец Д.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Бранцевич П.Ю. – К.Т.Н., доцент

Предложена структура базового модуля принятия решений на основе экспериментальных данных, получаемых от микроконтроллера STM32. Рассмотрены принцип функционирования аппаратной части модуля и его составные элементы. Представлены примеры применения модуля.

Современные встроенные устройства в режиме реального времени получают информацию о состоянии различных объектов, осуществляют её обработку и принимают решения по формированию ответных управляющих или исполнительных воздействий. Такие цифровые устройства нашли широкое распространение и применение в различных областях, начиная со смартфона или фитнес браслета и заканчивая автоматизированными системами контроля и управления в промышленности.

Целью данного проекта стала разработка базового модуля регистрации цифровых сигналов от MEMS-датчика ускорения, GPS-модуля, датчика температуры, атмосферного давления с последующим определением параметров принятого сигнала и принятием соответствующих решений. Разработана архитектура программного средства модуля принятия решений, на базе которого возможно проектирование и построение различных прикладных систем.

Для получения исходных цифровых данных применяются первичные преобразователи, такие как GPS-датчик, преобразователь ускорения, гироскоп. После сбора первичной информации об исследуемом объекте, выполняет её обработка в соответствии с алгоритмами решаемых задач. Полученные такой системой результаты и данные могут быть входными для подсистем последующего более сложного анализа и принятия решений или непосредственно управлять блоками сигнализации или исполнительных действий.

Управляющий центр модуля микроконтроллер STM32 с вычислительным ядром Cortex-M3, первичными преобразователями и устройствами приведен на рис. 1. Для создания экспериментального прототипа модуля была выбрана отладочная плата STM32 Nucleo-F103RB. Максимальная тактовая частота работы микроконтроллера 72МГц. Для энергосбережения и обеспечения продолжительной работы модуля на одном заряде аккумулятора использовались режимы работы микроконтроллера stop и sleep, позволяющие на время остановить выполнение кода программы с сохранением состояния всех выводов, а затем продолжить работу с того места, где она остановилась [1].

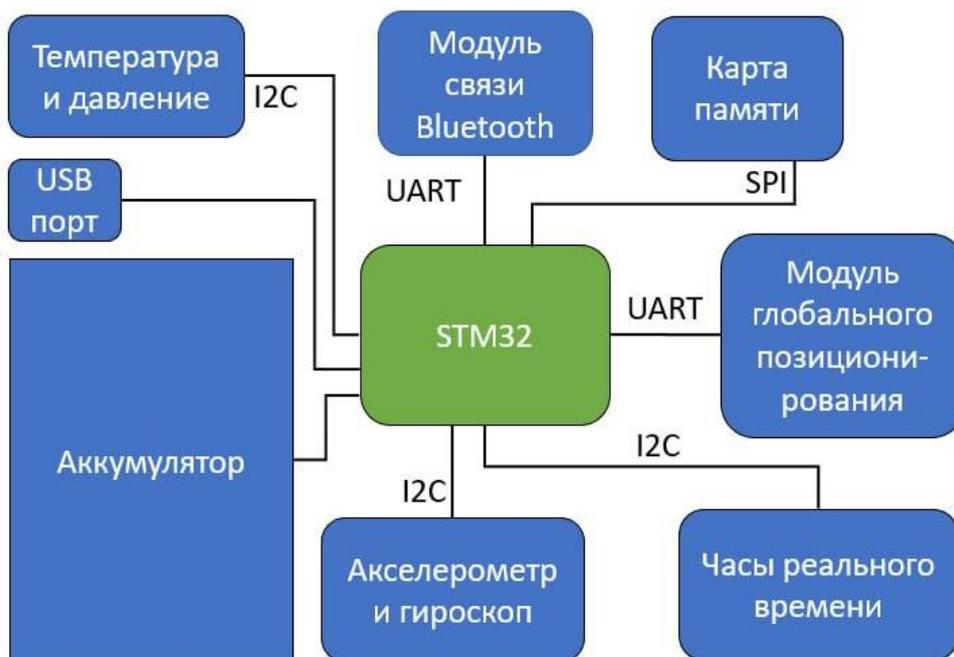


Рисунок 1 – Структурная схема устройства

Датчик температуры и давления (BMP180), модуль акселерометра и гироскопа (MPU6050), а также часы реального времени (DS3231) подключаются к микроконтроллеру по интерфейсу I2C. Данные передаются по двум линиям – линия данных и линия тактов. Каждое устройство распознается по уникальному адресу и может работать как передатчик или приёмник, в зависимости от назначения устройства.

Передача данных от модуля GPS к контроллеру и от контроллера к компьютеру выполняется по интерфейсу UART. Для определения местоположения устройства на карте в проекте используется GPS-модуль на базе чипа Neoway G7. Выбранный модуль принимает сигналы спутников глобального позиционирования – GPS, GLONASS и Galileo. Данные с GPS-модуля передаются на контроллер в формате текстовых сообщений NMEA.

Работа с SD-картой осуществляется по интерфейсу SPI с помощью библиотеки FatFS. Для одновременной работы нескольких устройств по данному интерфейсу была добавлена линия SS (Slave Select), при этом ведущее устройство одно – микроконтроллер STM32.

Эксплуатация устройства автономная, питание от аккумулятора. Для эффективной работы микроконтроллера используются режимы пониженного энергопотребления. При этом удалось сохранять заряд аккумулятора до нескольких дней.

Разработано программное средство, которое получает данные от устройства, обрабатывает их и визуализирует в графике, карте и 3D-модели. Программное средство выступает в качестве типового базового модуля принятия решений, на основе которого можно создавать более сложные системы принятия решений.

Модель базового модуля принятия решений изображена на рисунке 2, где $f(x)$ и $\varphi(\omega)$ – функции обработки параметров и характеристик объекта [2]. Функции могут рассчитывать среднееквадратичное значение, пик фактор, или другие характеристики сигналов. В качестве характеристик и параметров выступают данные, получаемые с устройства. При получении обработанных первичных характеристик, модуль Ψ_k принимает какое-либо решение в зависимости от поставленной задачи. Данное решение y_k может являться входом для следующих модулей принятия решений в другой системе.

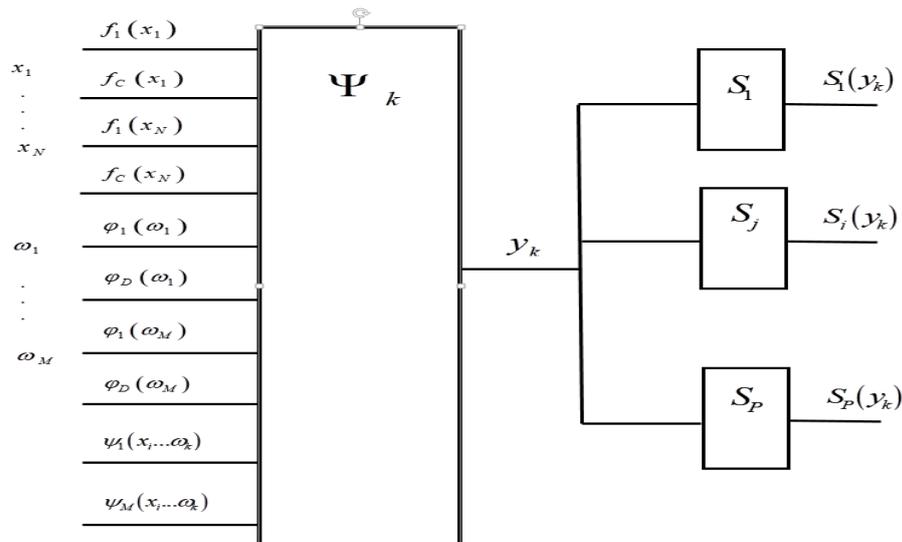


Рисунок 2 - Модель базового модуля принятия решений

Полученный прототип типового модуля принятия решений может применяться в различных автоматических или автоматизированных системах, например, в оценке вибросостояния механизмов и агрегатов, технического состояний автомобилей, качества дорожного покрытия, сохранности перевозимых грузов и других областях хозяйственной деятельности.

Список использованных источников:

1. Авсяник, Е. С. Программно-аппаратный модуль мониторинга перемещения движущихся объектов / Е. С. Авсяник, Д. В. Деменковец // Веб-программирование и интернет-технологии WebConf2021 : материалы 5-й Международной научно-практической конференции, Минск, 18-21 мая 2021 г. / Белорусский государственный университет ; редкол.: И. М. Галкин [и др.] – Минск, 2021. – С. 57–58.
2. Бранцевич, П. Ю. Проектирование сети принятия решений для систем вибрационной диагностики и автоматизации защиты=Decision network for systems vibration diagnosis and protection automation / Бранцевич П. Ю. // Современные методы и приборы контроля качества и диагностики состояния объектов: сборник статей 8-й

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

*Международной научно-технической конференции, Могилев, 29-30 сентября 2022 г. / Белорусско-Российский университет
; редкол.: М. Е. Лустенков [и др.]. – Могилев, 2022. – С. 49-56*

ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗАЦИИ РАСПРЕДЕЛЕНИЯ ЗАДАЧ

Альберт К.Ю.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Куликов С.С. – к.т.н., доцент

В данной статье рассматривается проблематика автоматизации рутинных задач в процессе разработки программного обеспечения, предоставлен алгоритм решения таких задач и архитектура возможного ПО.

Вопросы автоматизации распределения задач всегда являлись актуальными, например, составление расписания в учебном заведении, графика работы станка на предприятии или расписания смен работников общественного питания. Большое количество разработок появилось в середине и в конце 70-х годов прошлого столетия, когда в системах управления стали использовать ЭВМ. В настоящее время вместе со стремительным развитием информационных технологий возникла проблема, как эффективно управлять проектами разработки ПО.

Наиболее распространённая методология управления проектами, которая применяется в настоящий момент – Agile.

Одна из методик Agile — это методика Scrum гибкого управления проектами, помогающая командам структурировать работу и управлять ею на основе набора ценностей, принципов и практик. Методику Scrum чаще всего применяют команды разработчиков приложений, но принципы и опыт её использования можно применить к командной работе любого рода. Это одна из причин такой популярности данной методики.

Главная задача программного средства – оказание помощи разработчикам ПО в автоматизации процесса распределения задач, а т.к. наиболее распространённая методика в разработке ПО – Scrum, то возможность иметь поддержку Atlassian – Jira, которая является наиболее распространённым продуктом для управления задачами, крайне выгодна.

Основополагающая цель в распределении задач – это планирование спринта (спринт – название временного промежутка, за который выполняется определённое количество задач). Компания Atlassian заявляет, что не важна точная планировка задач, так как это не рационально, а нужен примерный план. Составление плана – цель программного средства.

Любая задача оценивается согласно двум основным критериям: приоритетности и сложности, остальные факторы вторичны. На основе этих критериев задачи можно распределять. Если подойти с математической точки зрения, то это классическая задача «NP-полная задача комбинаторной оптимизации», а именно задача о рюкзаке. Однако, задача о рюкзаке не имеет решения за полиномиальное время, а лишь решение перебором (или улучшенным перебором), поэтому вопрос о самом эффективном решении о распределении задач всегда будет открыт.

Для решения данной задачи может быть предложен следующий алгоритм:

- 1) Для входных параметров отсортируем задачи вначале по приоритету;
- 2) Затем отсортируем их по сложности;
- 3) Выберем задачи последовательно и сформируем спринты.

На втором этапе задачи уже отсортированы в том порядке, как если бы один человек выполнял их последовательно без ограничений по времени. Однако, нам нужно распределить задачи по спринтам, и предложенный алгоритм нам полностью подходит. В общем случае задача с большим приоритетом, очевидно, более важна, чем задача с меньшим приоритетом. Задача с одинаковым приоритетом, но большей сложностью, также более важна, чем задача с меньшей сложностью, т.к. при выполнении задачи с большей сложностью может возникнуть много проблемных ситуаций, поэтому её стоит выполнить в первую очередь. Если наша задача не может быть выполнена в течение спринта, то мы её пропускаем и переходим к следующей.

Программное средство следует выполнять на основе микро-сервисной архитектуры. Главным преимуществом является то, что различные модули программного средства для автоматического распределения задач могут разрабатываться и поставляться независимо, что несомненно удобно, поэтому программное средство, которое планируется для автоматизации работы разработчиками, может быть легко усовершенствовано, например, для составления расписания ВУЗа.

Список использованных источников:

1. Collier, Ken W. (2011). *Agile Analytics: A Value-Driven Approach to Business Intelligence and Data Warehousing*. Pearson Education. pp. 121 ff. ISBN 9780321669544. "What is a self-organizing team".
2. "Microservice architectures: more than the sum of their parts?". IONOS Digitalguide. Retrieved 2022-03-29.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*, Second Edition. MIT Press and McGraw-Hill, 2001. ISBN 0-262-03293-7. Section 34.2: Polynomial-time verification, pp. 979–983.

ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ КОНТЕНТОМ САЙТА-ПОРТФОЛИО

Белоусов А.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Болтак С.В. – ассистент

Веб-приложение, выполненное при помощи языков программирования Java и JavaScript в виде трех модулей (модуль сервера, модуль клиента посетителей, модуль клиента администратора). Позволяет располагать в сети Интернет дизайн-проекты, обладает функционалом для их двухуровневой категоризации, а так же позволяет создавать пользовательские страницы сайта.

Портфолио - это список работ, проектов и достижений, которые показывают опыт, стиль и профессионализм автора. Для некоторых профессий (фотограф, дизайнер, модель) портфолио является неотъемлемым дополнением к резюме. Оно позволяет продемонстрировать лучшие умения автора, даже если он имеет минимальный опыт работы. Портфолио дизайнера выступает, своего рода, визиткой или онлайн резюме, которое помогает найти партнеров, заказчиков, создать личный бренд.

При реализации сайта-портфолио есть несколько вариантов: разместить имеющиеся работы на уже существующем и специально предназначенном для этого сайте, создать новый сайт в онлайн-конструкторе либо заказать разработку индивидуального решения у специализированной компании. Все варианты имеют свои достоинства и недостатки, но в данной работе рассматривается последний вариант – реализация решения по индивидуальному заказу.

Сайты-портфолио имеют схожую структуру: позволяют категоризировать имеющиеся проекты, например, по нескольким уровням иерархии либо при помощи меток, тегов, по которым в дальнейшем можно будет легко найти определенный проект. Кроме того, функционал современных приложений такого типа включает в себя возможность удобного редактирования структуры отдельного проекта. Редактор может как предоставлять инструменты для самостоятельного редактирования всей структуры проекта (например, при помощи HTML/CSS), так и упрощать процесс, позволяя пользователю редактировать заранее подготовленный макет. Также, сайты-портфолио дополнительно могут предоставлять возможность пользователю отображать страницы по его собственным макетам.

Настоящее решение позволяет категоризировать проекты в двухуровневой иерархии (секции-категории). Пользователь может переносить проекты между секциями и категориями, изменять последовательность проектов в рамках одной категории или секции, зафиксировать позицию одних проектов и установить случайный порядок отображения при каждой загрузке страницы для других проектов. Каждый проект представляет собой набор блоков, имеющих один из следующих типов: текстовый, изображение, встраиваемый элемент. Пользователь имеет возможность создавать и удалять блоки, переносить их между проектами, изменять последовательность блоков внутри проекта. Дополнительно, пользователю предоставляется функционал для добавления на сайт секций, содержащих вместо набора категорий индивидуальный макет страницы пользователя. Для этой цели в модуле клиента администратора реализован удобный редактор HTML кода.

Приложение выполнено в виде трех независимых модулей: модуля сервера, модуля клиента посетителей и модуля клиента администратора. При сборке проекта, исходный код модулей клиентов преобразуется в набор статических ресурсов. После компиляции модуля сервера, эти ресурсы копируются внутрь, и сборка модуля сервера запаковывается в jar-архив, который далее может быть запущен на сервере. По умолчанию, доступ к модулю клиента посетителей осуществляется по GET HTTP-запросу к серверу по пути /; доступ к модулю клиента администратора осуществляется по GET HTTP-запросу к серверу по пути /admin. Настройка обоих путей доступа вынесена в конфигурационные файлы всех модулей и требует установки одинаковых значений соответствующих переменных как в файле модуля сервера, так и в файле модуля клиента.

Серверный модуль приложения выполнен при помощи языка Java и базирующемся на нём фреймворке Spring. Эта технология имеет всё необходимое для полноценной и удобной реализации сервера веб-приложения.

Оба клиентских модуля приложения реализованы при помощи языка JavaScript и построенной на нём библиотеке ReactJS. Данная библиотека позволяет разрабатывать приложение в виде набора компонентов, которые далее могут быть переиспользованы в различных частях приложения, что значительно сокращает время на разработку.

Список использованных источников:

1. Портфолио дизайнера – советы, шаблоны и примеры – [Электронный ресурс]. – Режим доступа: https://www.canva.com/ru_ru/obuchenie/50-sovetov-po-sozdaniyu-portfolio-dizajnera/ – Дата доступа: 11.04.2023.

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ ПРОГРАММНОГО КОМПЛЕКСА ОРГАНИЗАЦИИ УДАЛЕННОГО ПРОЦЕССА ОБУЧЕНИЯ

Белых Д.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Куликов С.С. – к.т.н., доцент

Обучение через интернет имеет главное преимущество в удобстве, которое создаёт оптимальную среду для активного и эффективного образовательного процесса. Кроме того, онлайн-обучение позволяет студентам более тесно взаимодействовать с преподавателями и получать своевременную оценку своей работы.

Во время образования человек учится не только получать знания, но и взаимодействовать с другими людьми, знакомится с культурой и усваивать ценности общества. Обучение формирует у человека такие качества, как упорство, воспитанность, гибкость взглядов, любознательность, общительность и целеустремлённость.

В современном обществе информационные технологии играют все более важную роль, включая сферу образования. В связи с этим возрастает потребность в более мобильных формах обучения, и дистанционное обучение становится все более популярным, наряду с традиционными формами обучения, такими как очное, заочное и вечернее образование.

Электронные образовательные платформы с интерактивными функциями становятся все более популярными, особенно в свете современных условий. Переход на удалённое обучение подразумевает использование современных методов обучения в процессе обучения.

Программный комплекс организации удалённого процесса обучения позволит решать следующие задачи:

- доступа к обучающим материалам в электронном виде;
- поддержка электронного хранения и анализа учебных достижений студентов;
- организация онлайн-тестирования и проверки знаний студентов;
- обеспечение возможности проведения онлайн-лекций и вебинаров.

Приложение представляет собой клиент-серверную архитектуру. Приложение построено на платформе Java с использованием Spring Framework. Java — это не только универсальный язык программирования, а целая платформа, объединяющая различные технологии, используемые для целого ряда задач, от создания десктопных приложений до написания крупных веб-порталов и сервисов. Spring Framework — универсальный фреймворк с открытым исходным кодом для Java-платформы. Он является простым, надёжным и наиболее популярным для написания серверной части Java-приложений. Таким образом, пользователи смогут использовать веб-приложение с любого устройства, где установлен браузер: компьютер, смартфон, планшет, ноутбук и т.д.

Представленное приложение даёт возможность создавать учебный контент, состоящий из лекций, заданий и тестов, которые формируются из текстовых и графических материалов, загружаемых преподавателем на платформу. Этот контент организуется в модули, которые ведёт наставник и его помощники, а также может быть изменён или удалён. Вся информация, добавленная при создании курса, сохраняется в базе данных и становится доступной для пользователей, которым предоставлен доступ к курсу.

Для улучшения усвоения материала обучающимися преподавателю предоставлена возможность оценки заданий прямо в системе. Также реализована возможность текстовой переписки для коммуникации между преподавателем и студентами.

На платформе существует 4 возможных ролей: гость, студент, преподаватель (с правом редактирования содержимого курса) и администратор. Роли определяют доступ к контенту в системе, управлению другими пользователями и настройке модулей.

В приложении существует встроенный редактор, который позволяет создавать тесты. Цель тестирования — проверка знаний обучающегося в той или иной области. Созданный тест будет доступен для прохождения на странице курса.

Использование программного продукта ожидается приведёт к следующим результатам: улучшение знаний студентов, более эффективный контроль и оценка их знаний, облегчение коммуникации между преподавателем и студентом, а также более удобный доступ к учебным материалам курсов.

Список использованных источников:

- | | | | | | |
|----|--|-----------------------|------|-------------|-------------|
| 1. | Minsk.edu | [Электронный ресурс]. | Дата | обновления: | 28.05.2021. |
| | URL: http://iso.minsk.edu.by/ru/main.aspx?guid=4771 | | | | |

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. *Spring Framework* [Электронный ресурс]. Дата обновления: 15.01.2023. URL: <https://spring.io/>
3. *Java* [Электронный ресурс]. Дата обновления: 21.03.2023. URL: <https://dev.java/>
4. *Client-server architecture* [Электронный ресурс]. Дата обновления: 4.04.2020. URL: <https://habr.com/ru/post/495698/>

ТИПОВЫЕ СХЕМЫ ПОСТРОЕНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ УЧРЕЖДЕНИЙ ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Бобрик А.Ю., Завалюк Д.Н.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Марков А.Н. – магистр техн. наук

Различные подходы к построению компьютерных сетей в учреждениях образования делают возможным обнаружение наиболее стабильной сетевой схемы для конкретной организации. В данной работе продемонстрированы типовые схемы компьютерных сетей трёх учреждений образования Республики Беларусь. Проанализированы применяемые в них сетевые технологии и соответствие их некоторым рекомендациям Министерства образования.

Широкое применение в образовательной сфере современных информационно-коммуникационных технологий формирует необходимость в обеспечении надёжности и безопасности создаваемых аппаратных комплексов. Так, вопрос организации распределённых сетевых ресурсов и балансировки нагрузки в условиях непрерывного роста числа пользователей Интернета создаёт новые преграды на пути к обеспечению быстрого и стабильного доступа к сети, необходимого для проведения качественной образовательной политики [1].

Для изучения применяемых в современных учреждениях образования принципов построения компьютерных сетей были построены схемы сетей двух учреждений общего среднего образования Республики Беларусь (далее – УО-1 и УО-2 соответственно) – ГУО «Гимназия №2 г. Барановичи» и ГУО «Гимназия-колледж искусств г. Молодечно» – и одного учреждения профессионально-технического образования (далее – УО-3) – ГУО «Барановичский государственный колледж машиностроения». В данной работе рассмотрены аппаратные составляющие сетей, и проанализировано их соответствие рекомендациям Министерства образования Республики Беларусь [2].

Для построения схем компьютерных сетей использовался симулятор сети передачи данных Cisco Packet Tracer. В дальнейшем исследовались следующие характеристики сети и её компонентов: интернет-провайдер, модели роутеров, количество занятых и свободных портов коммутаторов, тип широкополосного доступа в интернет, доступ к Wi-Fi.

Схемы компьютерных сетей УО-1 и УО-2 приведены на рисунках 1 и 2 соответственно. При построении схем использовались стандартные компоненты программы Cisco Packet Tracer. Способ подключения устройств определялся автоматически. Согласно рекомендациям Министерства образования, выбор провайдера должен осуществляться с учётом потребностей учреждения образования. Так, во всех трёх УО оператором по предоставлению услуг связи является РУП «Белтелеком». В УО-3 резервный канал обслуживается ООО «Объединённые сети» Unet.by. В УО-1 основной канал обслуживает Wi-Fi-роутер Huawei HG8245H, в УО-2 – Промсвязь МТ-PON-AT-4 (дополнительно используются роутеры TP-Link TL-WR841N и D-Link DIR-615). Основной канал в УО-3 формирует роутер ZTE F609, резервный – TP-Link Archer MR600. Также в ГУО «Барановичский государственный колледж машиностроения» имеется 9 точек доступа Wi-Fi TP-Link TL-WA801N, в качестве шлюза используется HP Prodesk 600 G1.

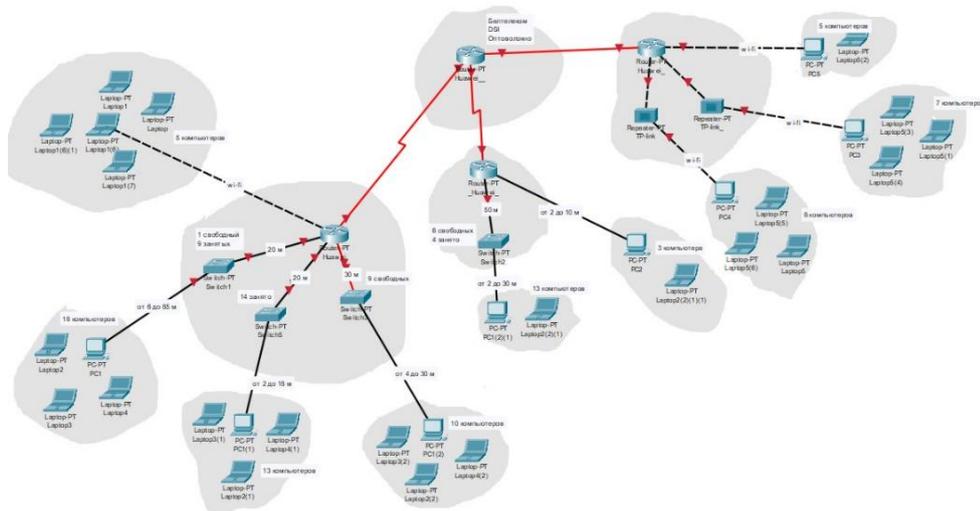


Рисунок 1 – Схема компьютерной сети ГУО «Гимназия №2 г. Барановичи»

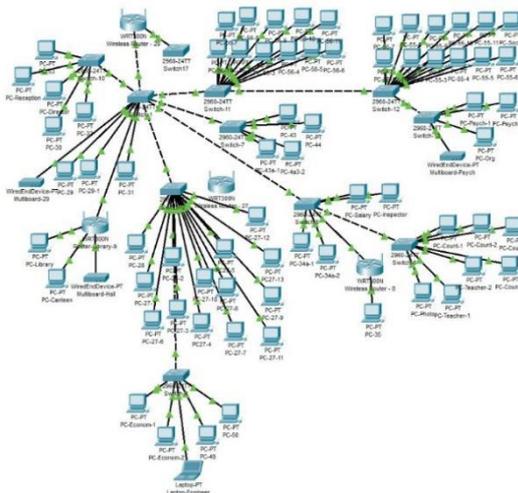


Рисунок 2 – Схема компьютерной сети ГУО «Гимназия-колледж искусств г. Молодечно»

Компьютерные сети во всех трёх изучаемых учреждениях образования построены на преимущественном использовании коммутаторов. Их достоинства, такие как увеличение доступной пропускной способности сети и её производительности, конкурируют с недостатками их применения: более высокая стоимость по сравнению с сетевыми мостами, сложность в обнаружении проблем сетевого подключения. На рисунке 3а, 3б и 3в показаны графики, иллюстрирующие отношение числа свободных и занятых портов коммутаторов в УО-1, УО-2 и УО-3 соответственно. Наличие свободных портов позволяет подключить дополнительные участки сети без дополнительного оборудования. Кроме того, общее количество доступных портов у беспроводных маршрутизаторов соответствует рекомендациям Министерства образования: 4 LAN-порта и 1 порт для беспроводной связи.

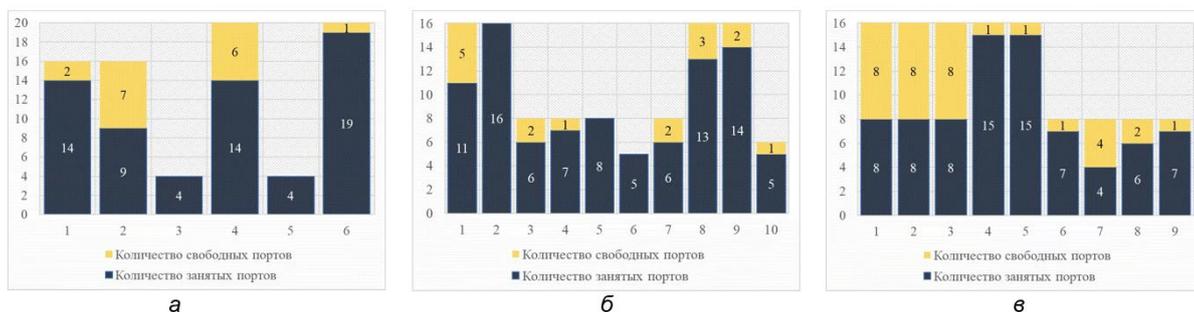


Рисунок 3 – Соотношение количества свободных и занятых портов коммутаторов в учреждениях образования: а) ГУО «Гимназия №2 г. Барановичи»; б) ГУО «Гимназия-колледж искусств г. Молодечно»; в) ГУО «Барановичский государственный колледж машиностроения»

Широкополосный интернет обеспечивает гораздо большую скорость передачи данных в сравнении с коммутируемым соединением. Для реализации широкополосного доступа в учреждениях образования используются разные технологии. Так, в УО-1 применяется технология DSL, в УО-2 – Ethernet, в УО-3 – xPON для основного канала и LTE для резервного.

Доступом к Wi-Fi обладают работники всех трёх учреждений образования. Точки доступа расположены равномерно по всей территории учебных заведений. В УО-1 имеется 2 репитера TP-Link Wi-Fi Range Extender AC750 DUAL BAND для усиления беспроводного сигнала и расширения зоны покрытия беспроводной сети. Для санкционированного доступа к Wi-Fi требуется использование секретного пароля, что необходимо для обеспечения безопасности сети и рекомендовано Министерством образования Республики Беларусь.

В данной работе были построены типовые схемы компьютерных сетей трёх государственных учреждений образования, а также проведён их анализ на соответствие некоторым рекомендациям Министерства образования Республики Беларусь. Созданные схемы могут быть использованы при дальнейшем тестировании реальных сетей, а также в учебных целях при изучении дисциплины «Компьютерные системы и сети».

Список использованных источников:

1. Парамонов, А. И. Проблема балансировки нагрузки при организации распределенных сетевых ресурсов в вузе / А. И. Парамонов, А. Н. Марков, К. М. Северин // Информатика, управляющие системы, математическое и компьютерное

моделирование (ИУСМКМ-2022): материалы XIII Международной научно-технической конференции, Донецк, 25-26 мая 2022 г. / Донецкий национальный технический университет. – Донецк: ДонНТУ, 2022. – С. 396-399.

2. Об использовании современных информационно-коммуникационных технологий в учреждениях общего среднего образования в 2022/2023 учебном году [Электронный ресурс]: инструктивно-методическое письмо Министерства образования Республики Беларусь, 11 окт. 2023 г. // Национальный образовательный портал. – Режим доступа: <https://adu.by/images/2022/10/imp-ИКТ-2022-23.pdf>. – Дата доступа: 20.03.2023.

ПРОБЛЕМЫ ДЕШИФРОВАНИЯ В КРИПТОСИСТЕМЕ РАБИНА

Болтак С. В. , Деренчук В.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Ярмолик В.Н. – д.т.н., профессор

В работе рассматривается алгоритм шифрования Рабина и предлагается способ выбора верного корня из четырёх, полученных при дешифровании.

Шифр Рабина относится к криптосистемам с открытым ключом или как ещё их называют - асимметричным криптосистемам. Алгоритмы получили такое название, так как в отличие от классических, шифрование осуществляется одним ключом, обычно его называют открытым или публичным, а дешифрование другим, секретным или закрытым ключом. Данный алгоритм явился результатом переосмысления криптосистемы RSA и был опубликован в 1979 году Майклом О. Рабином как часть цифровой подписи [1]. В своей работе Рабин впервые привёл доказательство стойкости криптосистем с открытым ключом, которая эквивалентна неразрешимости задачи факторизации числа N . В системе Рабина шифрование происходит быстрее чем в других алгоритмах с открытым ключом, поэтому её можно эффективно использовать во многих прикладных приложениях.

Для генерации пары открытый - закрытый ключ необходимо выполнить следующие шаги:

1. Выбрать два случайных простых числа p и q . При этом $p \approx q$ и $p \equiv q \equiv 3 \pmod{4}$.
2. Вычислить $n = p \cdot q$.
3. Выбрать случайное число $b < n$.

Открытый ключ - n и b , закрытый - числа p и q .

Прежде чем шифровать сообщение M , его необходимо разбить на блоки m_1, m_2, m_3, \dots , ($0 \leq m_i \leq n - 1$). Процедура шифрования имеет следующий вид:

$$c_i = m_i(m_i + b) \pmod{n}. \quad (1)$$

Для того, чтобы расшифровать сообщение, необходимо решить квадратное уравнение вида $m_i^2 + b \cdot m_i - c_i = 0 \pmod{n}$. Для извлечения квадратного корня из дискриминанта используется китайская теорема об остатках. В результате вычисляются четыре результата, что является проблемой криптосистемы Рабина, так как неизвестно, какой из четырёх результатов соответствует исходному сообщению. Особенно, если исходное сообщение - это поток случайных битов.

Один из вариантов решения проблемы - добавление к сообщению известного заголовка перед шифрованием. Ещё один способ выбрать правильный корень - например, продублировать последние биты сообщения. С высокой долей вероятности только один из четырёх корней будет содержать продублированные биты [2]. Однако, добавлять избыточность придётся для каждого шифруемого блока.

Способ выбора корректного корня, предлагаемый в данной работе, подходит для расшифровки файлов с любым содержимым. Это может быть как текстовое сообщение так и любой случайный поток битов. Предлагается рассматривать шифруемое сообщение как поток байтов. Байт может принимать значения от 0 до 255, то есть шифруются числа в этом промежутке. При дешифровании при больших p и q только один из четырёх корней будет лежать в данном диапазоне (таблица 1).

Таблица 1 – Поиск корректного корня в криптосистеме Рабина.

№	p, q, b	Искомые корни
1	$p = 523, q = 3$ $b = 1$	1551, 17 , 540, 1028
2	$p = 5003, q = 5227$ $b = 1234$	26149430, 17 , 495314, 25654133
3	$p = 523, q = 3$ $b = 2$	17 , 1550, 1550, 17

Как видно из таблицы, в нужном диапазоне могут лежать два одинаковых числа. В таком случае выбирается первый найденный. Предлагаемый способ без добавления избыточности обеспечивает выбор верного корня для любого потока случайных битов.

Список использованных источников:

1. Rabin, M. O. *Digitalized signatures and public-key function as intractable as factorization* / M. O. Rabin - Massachusetts institute of technology laboratory for computer science, Cambridge, 1979.
2. Menezes, A. P. *Handbook of applied cryptography* / A. P. Menezes, S. Vanstone - CRC Press, 1996.
3. Ярмолик, С. В. *Криптография и охрана коммерческой информации: методическое пособие* / С. В. Ярмолик, В. Н. Ярмолик - Минск, БГУИР, 2010 - 32 с.

МЕТОД ПОЛУЧЕНИЯ ИЗОБРАЖЕНИЯ ДВИЖУЩЕГОСЯ ОБЪЕКТА В РЕАЛЬНОМ ВРЕМЕНИ НА ПЛАТФОРМЕ RASPBERRY PI CM 4

Борисевич А.В. студент, Мелешко А.С. студент, Деменковец Д.В. аспирант

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Леванцевич В.А. – старший преподаватель

В работе рассматриваются метод и устройство получения изображений движущихся объектов на платформе Raspberry Pi Compute Module 4 используя видео-модуль Pi Camera. Описываются случаи, при которых получение изображения становится трудоемким.

В настоящее время очень востребованным является взвешивание железнодорожных вагонов в режиме динамического взвешивания, т.е. без остановок локомотива с последующей идентификацией каждого вагона без участия оператора [1]. Предлагаемая модель в частности решает одну из проблем получения четкого изображения железнодорожного вагона для определения номера вагона.

Обнаружение и получение изображений объектов является важной задачей в различных приложениях компьютерного зрения, таких как видеонаблюдение, робототехника и самоуправляемые автомобили. Raspberry Pi - популярный и недорогой одноплатный компьютер, который обеспечивает универсальную платформу для разработки приложений компьютерного зрения. Модуль камеры Raspberry Pi, Pi Camera, обеспечивает получение высококачественного изображения при низкой стоимости, что делает его идеальным выбором для разработки систем обнаружения объектов и слежения за ними.

Используемое в данной работе устройство Raspberry Pi Compute Module 4 оснащено четырехъядерным 64-разрядным SoC-процессором Cortex-A72 (ARM v8) что позволяет бесперебойно работать ресурсоемким приложениям. Конструкция устройства и его состав изображено на рис.1.

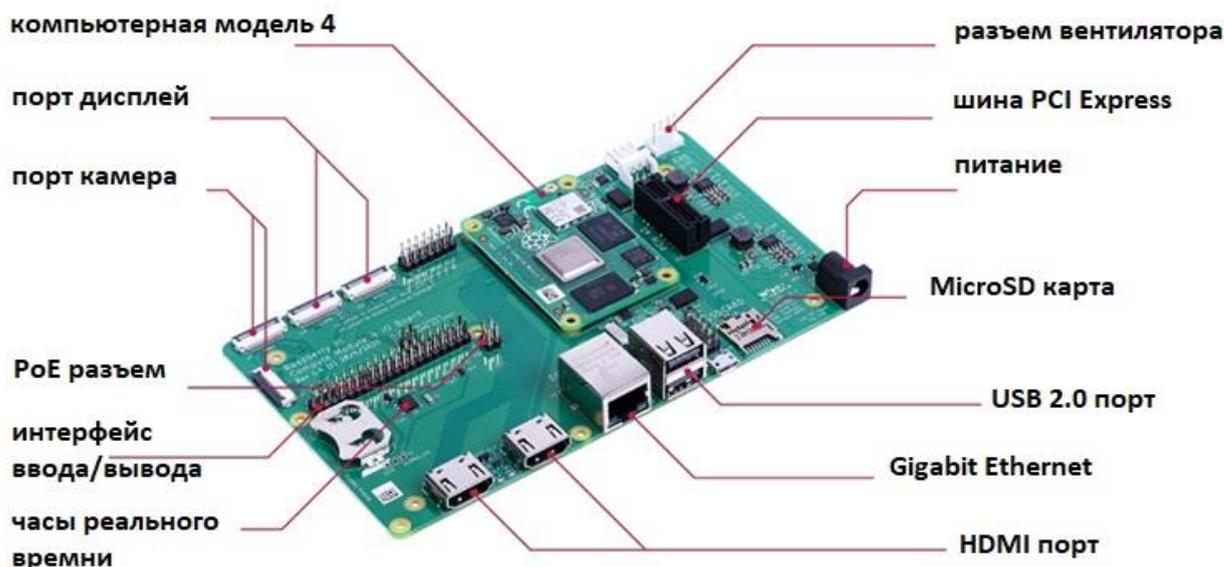


Рисунок 1 – Конструкция устройства Raspberry Pi CM 4 IO Board

Предлагаемое устройство Raspberry Pi и видео модуль Pi Camera захватывает видеокadres с нестандартным разрешением (от 1 до 60 пикселей в ширину и 1080 пикселей в длину) и затем объединяет полученные кадры в единое изображение. Контакты GPIO Raspberry Pi используются для управления устройством, позволяя его запускать и останавливать процесс захвата кадров и указывать на обнаружение движущегося объекта. Операционная система устройства Pi на основе Debian использует библиотеку libcamera для захвата изображения с видео модуля. Данный способ позволяет получить четкое изображение движущегося объекта в режиме реального времени. Но достаточно часто изображение настолько нечеткое, что даже человеку сложно распознать номер

вагона по фотографии [2]. В данном случае применяются алгоритмы обработки изображений такие как вычитание фона, обнаружение больших двоичных объектов и фильтрацию Калмана.

Для обработки изображений применяет ряд методов, которые можно в целом разделить на две категории: низкоуровневые и высокоуровневые методы. Низкоуровневые методы включают в себя базовые операции обработки изображений, такие как улучшение изображения, шумоподавление и регулировка контрастности. Высокоуровневые методы включают в себя более сложные операции, такие как обнаружение и распознавание объектов, сегментация изображений. Raspberry Pi предоставляет ряд инструментов и библиотек как для низкоуровневой, так и для высокоуровневой обработки изображений.

Методы обработки изображений включают в себя операции, такие как обнаружение объектов, распознавание объектов и сегментация изображения. Raspberry Pi предоставляет инструменты и библиотеки для реализации этих методов обработки изображений. При обнаружении объектов используются инструменты OpenCV и TensorFlow. При распознавании объектов OpenCV и Recognition. Классификация изображений выполняется с помощью TensorFlow и scikit-learn.

В процессе реализации были выявлены следующие сложности получения четкого изображения модулем камеры Raspberry Pi. Размытие изображения в движении. Когда транспортное средство движется, его номерной знак может казаться размытым на изображении, снятых камерой. Решением этой проблемы может быть увеличение частоты кадров получения изображений, либо уменьшением области захвата. Другой проблемой получения четкости являются различные условия освещения (например, насыщенность бликами изображения, полученного при восходе или закате солнца в случае фиксации номеров вагонов) [3].

Номерные знаки могут быть трудно различимы при ярком солнечном свете или в условиях низкой освещенности. Решением может быть использование комбинаций камер инфракрасного и видимого света в условиях низкой освещенности.

Еще одной сложностью является ориентация распознаваемого объекта, а различных направлениях. Решение использование камер с широким полем зрения для съемки под различными углами.

Проблема отражения также является важной. Объекты могут отражать солнечный свет или свет фар, что может создавать блики и затрудняет идентификацию. Решением является использование поляризационных фильтров на объективе камеры, для уменьшения бликов и улучшения качество изображения.

В целом, хотя камеры Raspberry Pi могут обеспечить недорогое и доступное решение для захвата изображений, важно знать об ограничениях и проблемах, с которыми можно столкнуться при их использовании. Понимая эти проблемы и используя соответствующие методы и оборудование, можно получать высококачественные изображения с помощью камер Raspberry Pi для различных применений

Конечным результатом применения различных методов обработки изображения полученного с модуля камеры Raspberry Pi может служить изображение на рис.2.



Рисунок 2 – Полученное изображение для последующей идентификации

Полученное изображение в последующем будет использовано для идентификации объекта, в данном случае определения номеров и названий железнодорожных вагонов с системой автоматизированного взвешивания.

Список использованных источников:

1. Частное производственно-торговое унитарное предприятие «Саха-пром» - Автоматизация взвешивания железнодорожных вагонов [Электронный ресурс] ~/ ~--- Режим доступа: <http://sakha-prom.by/smart/avtomatizatsiya-vzveshivaniya.html> --- Дата доступа: 17.03.2020

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. Деменковец, Д. В. Особенности компьютерного распознавания номеров вагонов железнодорожного транспорта / Д. В. Деменковец, А. О. Буйко // Компьютерные системы и сети: материалы 54-й научной конференции аспирантов, магистрантов и студентов, Минск, 23 – 27 апреля 2018 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2018. – С. 62 - 64.

3. Деменковец, Д. В. Программное средство автоматизации процесса динамического взвешивания железнодорожных вагонов с распознаванием номеров на основе искусственной нейронной сети / Деменковец Д. В., Дубицкая К. Ю., Борисевич А. Н. // Информационные технологии и системы 2019 (ИТС 2019) = Information Technologies and Systems 2019 (ITS 2019) : материалы международной научной конференции, Минск, 30 октября 2019 г. / Белорусский государственный университет информатики и радиоэлектроники; редкол. : Л. Ю. Шилин [и др.]. – Минск, 2019. – С. 38 – 39.

ПРОГРАММНОЕ СРЕДСТВО ВИЗУАЛИЗАЦИИ В РЕАЛЬНОМ ВРЕМЕНИ ФИЗИЧЕСКИ КОРРЕКТНЫХ ТРЕХМЕРНЫХ СЦЕН НА БАЗЕ ГРАФИЧЕСКОГО API DIRECTX

Борсуков Е. А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Красковский П. Н. – ст. преподаватель, Каф. ПОИТ

В данной работе предлагается программное средство для визуализации трехмерных сцен, которое использует современные подходы и алгоритмы для отображения трехмерных моделей на экран устройства и просчета физически корректного освещения в реальном времени.

На сегодняшний день задачи в области компьютерной графики связанные с выводом фотореалистичных изображений трехмерных объектов, созданными 3D художниками, остаются одними из самых трудоемких вопросов для вычисления. Самые точные, с точки зрения результатов, подходы визуализации 3D сцен могут работать часами даже на самых производительных конфигурациях вычислительных устройств, что может стать проблемой в тех отраслях, где критически важно время, затраченное на производства продукта. Однако с ростом производительности вычислительных систем стало возможно делать эти просчеты и получать результат быстрее, а то даже на ходу.

Основной идеей для максимальной правильной визуализации трёхмерных сцен в реальном времени лежит в расчете освещения объектов по методу физического корректного рендеринга.

Физически корректный рендеринг (physically based rendering, PBR) – это набор методов по визуализации трехмерных объектов, которая в большей или в меньшей степени основана на физических законах реального мира. Поскольку визуализация данным подходом направлена на имитацию света физически правдоподобным способом, то результат, как правило, получается реалистичней по сравнению с более простыми алгоритмами, такими как затенение по Фонгу и модели Блинна-Фонга. Кроме этого данный подход позволяет художникам создавать поверхностные материалы на основе физических параметров, не прибегая к хитростям по работе с освещением, так как они будут выглядеть правильно, независимо от условия освещения [1].

Физически корректный рендеринг сам по себе сложен для расчета, так как при расчете освещенности каждой точки на сцене нужно высчитывать интеграл, представленный в формуле ниже [2].

$$L_r(x, \omega_r) = \int_{\Omega} f(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i' d\omega_i \quad (1),$$

где $f(x, \omega_i \rightarrow \omega_r)$ – двулучевая функция отражательной способности; $L_i(x, \omega_i)$ – цвет входящего луча света; $\cos \theta_i'$ – угол между вектором входящего луча света и нормалью поверхности.

При условии, что на сцене располагается множество разных объектов с различными источниками света, просчет освещения для каждой модели по формуле (1) может стать затратной с точки зрения производительности. Для уменьшения количества просчетов для таких сцен используется метод отложенного рендеринга.

Отложенный рендеринг работает путем отображения всех объектов сцены в серию двухмерных буферов изображения, которые хранят информацию, необходимую для выполнения расчетов освещения в более поздние этапы отрисовки кадра [3]. Данный подход позволяет уменьшить количество просчетов до числа источников света на сцене. При этом его также применяют для удобства добавления различных эффектов в итоговое изображение на этап постобработки.

Список использованных источников:

1. Ресурс для изучения основ компьютерной графики на базе графического API OpenGL – Learn OpenGL [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://learnopengl.com/PBR/Theory>
2. Технический блог в области компьютерной графики от специалистов из компании NVIDIA – NVIDIA Developer [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://developer.nvidia.com/blog/ray-tracing-essentials-part-6-the-rendering-equation/>
3. Ресурс для изучения основ компьютерной графики – 3Dгер [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.3dger.com/forward-plus/>

ПРОГРАММНОЕ СРЕДСТВО КОНТРОЛЯ ЛОКАЦИИ И СОСТОЯНИЯ ОБЪЕКТА НА БАЗЕ КОНТРОЛЛЕРА STM32

Быковский В.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Леванцевич В.А. –старший преподаватель

Проанализированы методы локации. Рассмотрены системы удалённого сбора данных, методы подготовки данных, архитектурные стили разработки программных средств, варианты моделей для взаимодействия с клиентом. Определен оптимальный способ внедрения программного средства.

Радиолокация - область науки и техники, занимающаяся наблюдением за различными объектами в воздухе, на воде и суше, определение их местоположения и расстояния до них [1]. Радиолокация применяется в различных областях, например, таких как : навигация, разведка, метеорология, геодезия и т.д.

В ходе исследовательской работы, в качестве базы для осуществления контроля локации и состояния объекта, был выбран микроконтроллер STM32 от компании STMicroelectronics. Данный контроллер позволяет быстро собирать и обрабатывать информацию с периферийных устройств. В качестве способа отслеживания их состояния был выбран способ поочерёдного опроса через определённые промежутки времени. Для получения наиболее качественной оценки производится расчёт доверительного интервала и математического ожидания, что позволяет быстро получать готовые, к дальнейшей обработке, данные.

В качестве метода для геолокации были выбраны системы спутниковой навигации GPS [2] и ГЛОНАСС [3]. Для передачи и обработки используются географические координаты объекта в градусах, что унифицирует возможность их использования и позволяет хранить в наиболее компактном формате.

В качестве архитектурного стиля программного средства используется REST API в совокупности с MVC моделью. Данный подход позволяет создавать мультиплатформенные приложения любого масштаба. Все данные, полученные с микроконтроллера, сохраняются в базе данных, а затем динамически отображаются на странице пользователя.

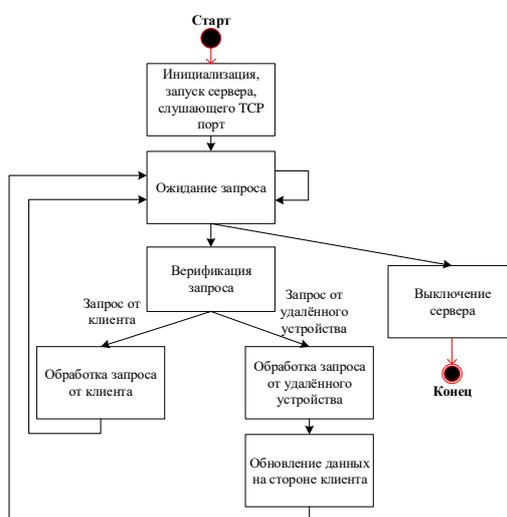


Рисунок 1. – Схема работы программного средства

Выводы: работа с системами спутниковой навигации увеличит скорость определения координат и их достоверность. Получение географических координаты объекта в градусах через фиксированный промежуток времени является наиболее удобным и эффективным подходом. Корректировка данных о состоянии объекта позволит получать информацию с высокой точностью.

Список использованных источников:

1. Гринкевич А., Радиолокация: учеб. пособие / А.В. Гринкевич- Минск: БГУИР, 2015. – 190 с.
2. GPS. Wikipedia [Online]. - <https://ru.wikipedia.org/wiki/GPS>
3. Королёв, Стандарт эксплуатационных характеристик открытого сервиса (СТЭХОС) / Королёв- Москва: ИАЦ ВНО ФГУП ЦНИИмаш, 2019. – 59 с.

СИСТЕМА РАСПОЗНАВАНИЯ ТЕКСТА НА ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЯХ

Воронцова П.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Деменковец Д.В. – старший преподаватель

В данной работе рассматриваются система распознавания текста на графических изображениях. Описываются такие алгоритмы, как предобработка данных и обучение нейронной сети. Представлены примеры работы медианной и монохромной фильтраций, которые используются для предварительной обработки изображения к обучению нейронной сети.

Системы распознавания текста на графических изображениях – это технология, которая позволяет компьютеру распознавать и переводить на понятный для него язык текст, который находится на изображении. Система может работать с различными языками и шрифтами, что делает ее универсальной и применимой в различных областях [1].

При создании системы распознавания текста на графических изображениях была выполнена следующая последовательность действий.

Выполняется сбор данных, который включает в себя накопление графических изображений с текстом, чтобы использовать их для обучения системы. Это достаточно трудоемкий процесс, так как необходимо собрать большой объем изображений, на которых присутствует текст. Для набора данных используются различные источники. В основном источниками являются открытые базы данных и различные интернет-ресурсы. Также при сборе данных важно учитывать разнообразие текстов на изображениях для эффективной работы модели по распознаванию различных видов текста. Некоторые изображения могут быть защищены авторским правом или содержать конфиденциальную информацию. Поэтому необходимо убедиться в том, что использование этих изображений не нарушает законодательство.

После этого выполняется предобработка данных. Этот этап включает в себя предварительную обработку изображений для исключения шумов, улучшения контрастности и приведения их к единому формату. Для обработки используются 2 фильтра: медианный и монохромный. Медианный фильтр применяется для минимизации шума и сглаживания острых краев букв (засечек), искажающих его размер и препятствующий дальнейшему распознаванию. Суть этого фильтра заключается в замене каждого пикселя на изображении на медианное значение яркости его окрестности. Пример работы медианной фильтрации приведен на рисунке 1.

Далее изображение обрабатывается с помощью монохромного фильтра, который выполняет четкую бинаризацию изображения, при этом фиксируя границы символа. При монохромной фильтрации цветные пиксели изображения преобразуются в черно-белые (монохромные) пиксели. Этот фильтр позволяет сделать изображение более четким и тем самым упростить процесс распознавания символов текста. Пример работы монохромной фильтрации приведен на рисунке 2. При этом монохромные изображения занимают значительно меньше места в памяти компьютера, ускоряя обработку данных [2].

Полный алгоритм предобработки массива графических изображения приведен на рисунке 3.

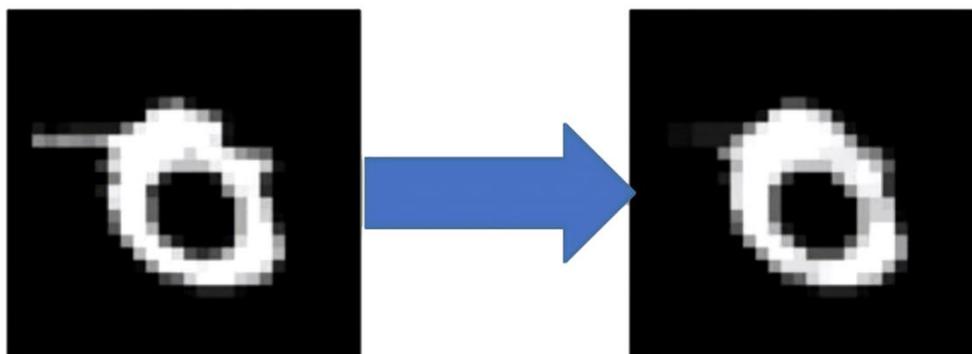


Рисунок 1 – Медианная фильтрация

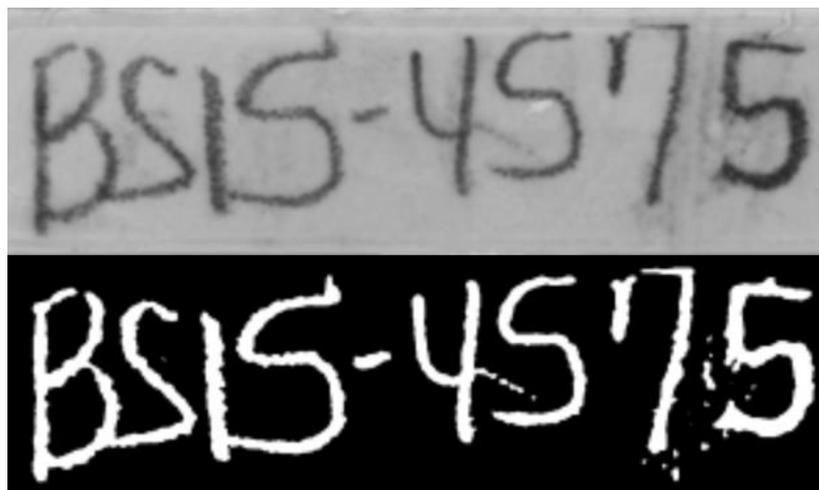


Рисунок 2 – Монохромная фильтрация



Рисунок 3 – Предобработка массива изображений

После предобработки массива изображений следует процесс обучения модели. Для этого используются различные алгоритмы машинного обучения, такие как сверточные нейронные сети или методы опорных векторов [3]. Алгоритм обучения нейронной сети приведен на рисунке 4.



Рисунок 4 – Обучение нейронной сети

Завершающими этапами являются тестирование и оптимизация. Модель тестируется на новых изображениях. Выполняется расчет параметров и оптимизация модели для достижения наилучшей точности распознавания текста в изображении.

В результате была разработана система, позволяющая извлекать текстовую информацию из изображений. Возможен вариант использования данной системы во многих областях. К примеру, распознавание номеров автомобилей, оптическое распознавание символов на чеках и документах, анализ рукописного текста. Это позволит ускорить и автоматизировать процесс обработки текстовых данных с изображений.

Список использованных источников:

1. "OCR and Its Applications in Industry." *International Journal of Computer Science and Mobile Computing*. [Электронный ресурс]. – Режим доступа: <https://www.ijcsmc.com/docs/papers/March2020/V9I3202002.pdf> – Дата доступа: 15.03.2023.
2. "Цифровая обработка сигналов" [Электронный ресурс]. – Режим доступа: [https://dl.booksee.org/genesis/43000/0d06ddd492b9413c9dd6999773695a56/as/\[Glinchenko_A.S.\]_Cifrovaya_obrabotka_signalov\(BookSee.org\).pdf](https://dl.booksee.org/genesis/43000/0d06ddd492b9413c9dd6999773695a56/as/[Glinchenko_A.S.]_Cifrovaya_obrabotka_signalov(BookSee.org).pdf) – Дата доступа: 10.03.2023.
3. "Практический курс по нейронным сетям." [Электронный ресурс]. – Режим доступа: <https://books.ifmo.ru/file/pdf/2394.pdf> – Дата доступа: 12.03.2023.

ВЕБ-ПРИЛОЖЕНИЕ «ОРГАНИЗАЦИЯ ОПРОСОВ» НА БАЗЕ JAVA И SPRING

Горегляд В.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Петюкевич Н.С. – ст. преподаватель кафедры ПОИТ

В настоящее время сбор и анализ данных являются неотъемлемой частью любого бизнеса или исследовательской деятельности. Одним из самых популярных и эффективных способов сбора информации является проведение опросов. Опросы могут использоваться для различных целей: от изучения мнения потребителей о продукте до оценки уровня удовлетворенности сотрудников компании.

Проведение опросов может быть довольно трудоемким процессом, требующим значительных временных и финансовых затрат.

В этом случае приложения для организации опросов могут стать отличным решением для предприятий и исследовательских групп. Они предоставляют возможность создавать и администрировать опросы онлайн. Это позволяет существенно сократить время и затраты на проведение и обработку опросов, а также упростить процесс сбора данных.

Одним из наиболее важных преимуществ приложений для организации опросов является простота их использования. Они не требуют специальных знаний или навыков, что делает их доступными для широкой аудитории пользователей. Все, что нужно сделать, это создать опрос, распространить его среди аудитории и получить результаты.

Кроме того, использование приложений для организации опросов позволяет существенно увеличить количество респондентов. В отличие от традиционных методов проведения опросов, где необходимо обращаться к каждому респонденту лично, сервисы для проведения опросов позволяют отправлять опросы массово через электронную почту или социальные сети.

Пользователи могут получать результаты опросов в режиме реального времени, что делает возможным быстрое принятие решений.

В целом, приложения для организации опросов являются эффективным инструментом для сбора и анализа данных. Они позволяют существенно ускорить процесс организации опросов, сократить затраты на их создание и проведение, а также получить более точные и объективные данные. Поэтому их использование может стать отличным решением для организаций и исследовательских групп, которые заинтересованы в получении качественной и достоверной информации.

Разработанное приложение предназначено для организации опросов. Его архитектура базируется на клиент-серверной модели, где серверная часть построена на основе платформы Java, используя Spring Framework. Java не только является универсальным языком программирования, но и платформой, объединяющей различные технологии, для создания широкого спектра задач, от десктопных приложений до крупных веб-сервисов. Spring Framework — это надежный и популярный фреймворк с открытым исходным кодом для разработки серверной части приложений [1]. Клиентская часть приложения разработана с использованием React. Библиотека React основана на компонентном подходе, где каждый компонент отвечает за выполнение конкретной работы [2]. Архитектура приложения обеспечивает его расширяемость, а единообразие интерфейсов проектируемой системы позволит использовать функции серверной части приложения в других программных средствах (в виде открытых аппаратно-программных интерфейсов). Обмен информацией между клиентской и серверной частями осуществляется по протоколу HTTP в формате JSON. Для хранения данных используется СУБД PostgreSQL — свободная объектно-реляционная система управления базами данных, основанная на языке SQL.

В приложении учтены многие преимущества онлайн-опросов. Процедура создания опросов проста и интуитивно понятна. Она позволяет создать опрос комбинируя различные варианты типов вопросов на любую интересующую тему, предоставляется выбор дат начала и завершения опроса, а также тип доступа к опросу. Сразу же после создания — опрос публикуется, и респонденты могут проходить опрос, который может быть доступен как конкретной группе по паролю, так и любому пользователю. Для приглашения принять участие в опросе достаточно лишь переслать ссылку на опрос. В любой момент времени, после даты начала опроса, создатель опроса может просмотреть детальную статистику проведения выбранного опроса.

Основными пользователями данного приложения могут являться организации, заинтересованные в проведении опросов на различные темы, которые могут помочь им лучше узнать рынок, изучить тенденции развития, а также составить оптимальный план по продвижению

бизнеса.

Список использованных источников:

1. *Spring Framework* [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Spring_Framework
2. *React* [Электронный ресурс]. – Режим доступа: <https://reactjs.org/>
3. *Теория, практика и методология опросов в Интернет* // Федеральный образовательный портал ЭСМ [Электронный ресурс]. – Режим доступа : <http://ecsosman.hse.ru/info/seminar-doc/Wsurvey.doc>
4. *Социология и Интернет: перспективные направления исследования* // Федеральный образовательный портал ЭСМ [Электронный ресурс]. – Режим доступа : <http://ecsosman.hse.ru/conf/1621698041/index.html>.
5. *Онлайновые опросы: обыденность наступившего столетия* // Социо.ру – социологический портал [Электронный ресурс]. – Режим доступа : <http://www.socio.ru/public/doktorov/Edit.doc>

РЕГИСТРАТОР ВИБРАЦИОННЫХ СИГНАЛОВ

Григорьев Д.А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Леванцевич В.А. –старший преподаватель

Показана актуальность мониторинга текущего вибрационного состояния технических объектов на длительном интервале времени. Предложена архитектура регистратора вибрационных сигналов на основе встраиваемой системы с использованием микроконтроллеров семейства STM32. Описаны особенности её аппаратной и программной реализации.

При анализе вибрационных сигналов, полученных с технических объектов, возможно появление редких, кратковременных изменений формы сигнала. Эти изменения могут служить началом развития дефектов. Для выявления таких изменений необходимы системы, способные регистрировать вибросигналы на значительном интервале времени [1].

Для этих целей удобно использовать микроконтроллерные системы на кристалле, например такие как, Atmel, Stm, ESP, и др. Данные системы построены по Гарвардской архитектуре, имеют RISC-процессор, широкий набор специализированных, периферийных устройств и низкое энергопотребление. Все это позволяет использовать микроконтроллеры для решения различных задач. Для разработки модуля регистрации вибрационных сигналов к нему должны быть определены следующие основные технические требования:

1. Частотный диапазон исследуемого вибросигнала: 1 – 10000 Гц;
2. Амплитудный диапазон регистрируемых вибросигналов в единицах виброускорения: 0,1 – 1000 м/с²;
3. Количество каналов – 1;
4. Разрядность цифроаналогового преобразователя: не менее 12 разрядов
5. Максимальная частота дискретизации не менее 25 кГц;
6. Относительная погрешность измерений не должна превышать по абсолютной величине 10%;
7. Промежуточное сохранение результатов измерений на SD-карту;
8. Поддержка файловой системы FAT;
9. Поддерживаемые интерфейсы передачи данных: Serial, USB, Bluetooth, WI-FI;
10. Способы управления: автономный, внешний;
11. Питание: от внешнего бока питания и от аккумулятора.

На основе анализа технических характеристик и стоимости микроконтроллеров различных производителей было принято решение использовать микроконтроллер Stm32F103 компании STMicroelectronics, на базе процессора Cortex M3, установленного на плату прототипирования STM Nucleo board [2].

Микроконтроллер имеет встроенный 12-битный аналого-цифровой преобразователь с управляемой таймером частотой дискретизации до 1МГц. Наличие контроллера прямого доступа к памяти позволяет записывать данные с АЦП сразу в ОЗУ.

В качестве накопителя данных применяется внешний модуль SDCard, подключаемый к SPI- интерфейсу микроконтроллера. Для записи данных на SDCard используется библиотека Fatfs. Необходимый объем карты памяти можно приблизительно оценить исходя из того, что для записи 10 минут сигнала при частоте дискретизации АЦП 25кГц, требуется около 30 мегабайт памяти SDCard. Управлять регистратором можно как с помощью внешнего пульта управления, так и с мобильной или стационарной вычислительной системы. Для дальнейшей обработки оцифрованных сигналов они могут быть переданы на другую вычислительную платформу не только с помощью SDCard, но и с помощью интерфейсов Bluetooth, USB, WI-FI. При использовании USB используется CDC - интерфейс, представляющий собой виртуальный последовательный порт. Микроконтроллеры Stm32 имеют режимы работы с оптимизированным энергопотреблением, что позволяет использовать регистратор автономно, с элементами питания батарейного типа.

Использование регистратора позволяет регистрировать вибрационные сигналы на длительном промежутке времени и фиксировать малейшие изменения вибрационного состояния технического объекта.

Список использованных источников:

1 Бранцевич, П. Ю. Оценка технического состояния механизмов на основе анализа длительных вибрационных сигналов / Бранцевич П. Ю., Базылев Е. Н., Костюк С. Ф. // Современные методы и приборы контроля качества и диагностики состояния объектов: сборник статей 6-й Международной научно-технической конференции, Могилев, 19-20 сентября 2017 г. / Белорусско-Российский университет ; редкол.: И. С. Сазонов [и др.]. – Могилев, 2017. – С. 418-423.

2. STMicroelectronics. [Электронный ресурс]. – Электронные данные. – Режим доступа: [URL:https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html](https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html)

УДК 004.032.26

КЛАССИФИКАЦИЯ ДАННЫХ ПРЕЦИЗИОННОЙ МОДЕЛЬЮ НЕЙРОННОЙ СЕТИ

Деренчук В.И., Болтак С.В., Рябинкин Г.М., магистранты

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Смолякова О.Г. – канд. техн. наук

Аннотация. Классические нейронные сети современности имеют ряд недостатков. Они сказываются на точности результатов их работы. В настоящей работе будет охарактеризована модель и описаны принципы работы нейронной сети, базирующейся на структуре биологических нейронов головного мозга человека, которая нивелирует классических нейронных сетей и успешно подходит для решения задач классификации, распознавания образов и других.

Ключевые слова. Нейронная сеть, прецизионная модель нейронной сети, искусственный интеллект, машинное обучение, классификация данных, распознавание изображений, синапс, рецептор.

При помощи передовых технологий машинного обучения в современности решается огромное количество задач, в том числе экономических, социально значимых, касающихся информационной безопасности [1, 2] и иных. Именно нейронные сети считаются универсальными моделями в машинном обучении, поскольку позволяют решать данный широкий спектр задач. Несомненными их достоинствами являются: отказоустойчивость (при выходе из строя части нейронов остальные нейроны остаются работоспособными, и несмотря на снижение точности работы, ответы поврежденной ИНС остаются логичными и правильными), эффективная фильтрация шумов (после обучения ИНС способны обрабатывать только нужную им информацию, игнорируя посторонние шумы), адаптация (возможность адаптации к изменениям во входных данных позволяет работать в правильном режиме все время, при этом непрерывное самообучение – самое важное свойство ИНС), скорость работы (каждый нейрон, по сути, является микропроцессором, а так как в ИНС их тысячи, решение происходит намного быстрее, чем при использовании обычных алгоритмов решения).

Вместе с тем, на практике использование классических моделей нейронных сетей вызывает множество трудностей. Так, на этапе проектирования нейронной сети возникают вопросы, ответить на которые зачастую можно только эмпирическим путём, а использование готовых решений не всегда подходит по тем или иным причинам. Одной из доминирующих проблем применения моделей ИНС является заранее неизвестная архитектура проектируемой нейронной сети и ее степень сложности, которых будет достаточно для достоверности получаемого результата. Так, выбор количества слоёв, нейронов на них, настройка параметров обучения становятся нетривиальными задачами [3]. Более того, при их обучении требуется большой объем данных с большим количеством параметров, процесс изменения которых зачастую необозрим, а работа с такими данными в целом занимает большое количество времени. Совокупность указанных недостатков обычных ИНС (многослойный перцептрон, сверточные нейронные сети и иные) усложняет реализацию нейронных сетей, к тому же, зачастую невозможно их дополнительное обучение [4, 5].

Архитектура модели прецизионной нейронной сети существенно отличается от классических моделей. Главными особенностями предлагаемой модели являются:

1. процесс высвобождения нейромедиатора синапсами и его влияния на рецепторы;
2. свойства движения рецепторов;
3. условия генерации выходного импульса нейрона (спайк),
4. критерий несоответствия хранимой информации поступающей в нейрон после обучения.

Приведем графическое описание процессов обучения, а также распознавания сигналов согласно предлагаемой модели. На рисунке 1 изображен процесс обучения нейрона, состоящего из 3 синапсов (С1 и С2) и 1 рецептора (Р), для удобства на примере сигнала из 3 импульсов (нейрон прецизионной модели ИНС может содержать множество таких структурных единиц). На представленной схеме С1, С2 и С3 высвобождают нейромедиатор, а Р – рецептор, который на него реагирует. Когда в первом цикле Ц1 на вход нейрона поступает импульс, синапсы С1, С2 и С3 высвобождают некоторое количество нейромедиатора. Указанный процесс вызывает сдвиг рецептора Р из исходного положения П0 в положение П1. При приходе второго импульса на нейрон рецептор также смещается в некоторое финальное положение П2, после чего фиксируется в нем, при этом координаты сохраняются в связи с окончанием обучения. В последующем данные координаты будут использоваться для расчета критерия дивергенции и распознавания иных

поступающих на нейрон по окончании процесса обучения сигналов. В этой связи целесообразно назвать его образцовым.

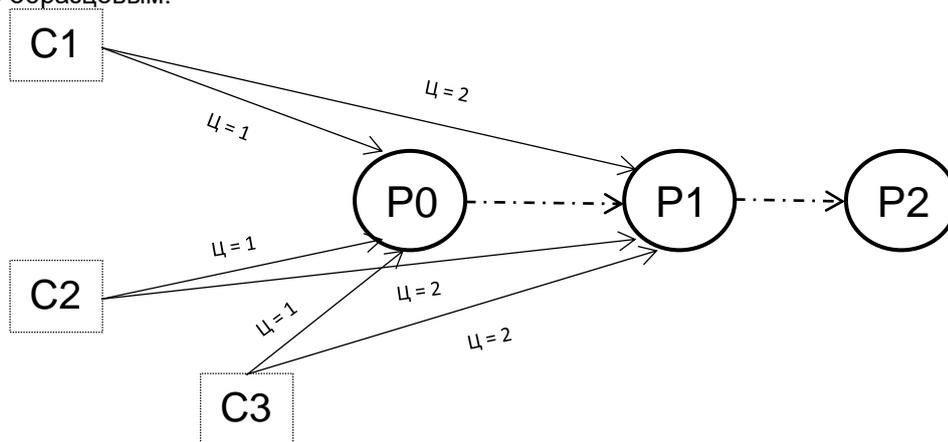


Рисунок 1 – Процесс обучения нейрона

На рисунке 2 изображен процесс распознавания одинакового отличного от обучающего сигнала из 3 импульсов.

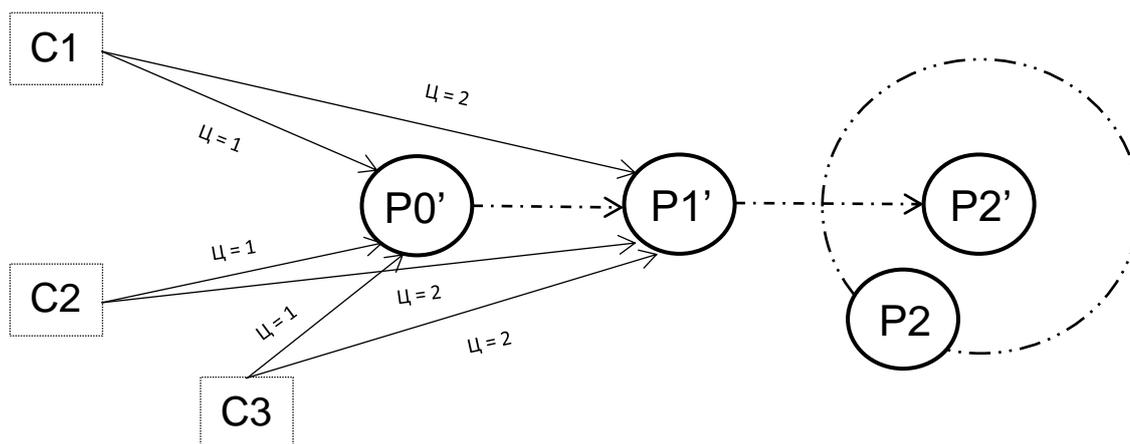


Рисунок 2 – Процесс распознавания сигнала

В данном случае при распознавании сигнала применяются виртуальные копии рецепторов – фантомные рецепторы (их координаты используются исключительно для сравнения сигналов в конечный момент времени, при этом не сохраняются). Среднее расстояние от образцовых рецепторов до фантомных следует принять за критерий расхождения. $P0'$ – рецептор в начальный момент времени, его координаты совпадают с координатами $P0$. На первом и последующих циклах происходят те же процессы, что и при обучении нейрона – к синапсам поступает импульс, они выделяют нейромедиатор и рецептор смещается. Так как сигнал отличается от используемого для обучения нейрона, изменилась траектория движения рецептора, а в итоговый момент времени (импульс на втором цикле) фантомный рецептор занимает положение $P2'$, которое отличается от положения $P2$ (образцовый рецептор). Для определения величины таких различий целесообразно рассчитать значение критерия расхождения. Так, если установить некоторое пороговое значение – допустимую степень отклонения, то можно решить, относится ли распознанный сигнал к тому же классу, которому был обучен нейрон. При объединении нейронов в сеть, один изолированный нейрон или их группа могут отвечать за один класс сигналов. Тогда классификация будет осуществляться путем подачи распознаваемого сигнала на обученные нейроны или группы нейронов и вычисления для них критерия расхождения. Классу распознанного сигнала

соответствует класс, для нейрона (группы нейронов) которого критерий расхождения имеет наименьшее значение.

Прецизионная модель нейронной сети может быть реализована в виде библиотеки языка программирования (на рисунке 3 изображена схема, которая ляжет в основу работы такой библиотеки), которая позволит создавать, обучать и применять ИНС при разрешении практических задач и ускорить параллельные вычисления, что форсирует процессы при использовании многоядерных систем.

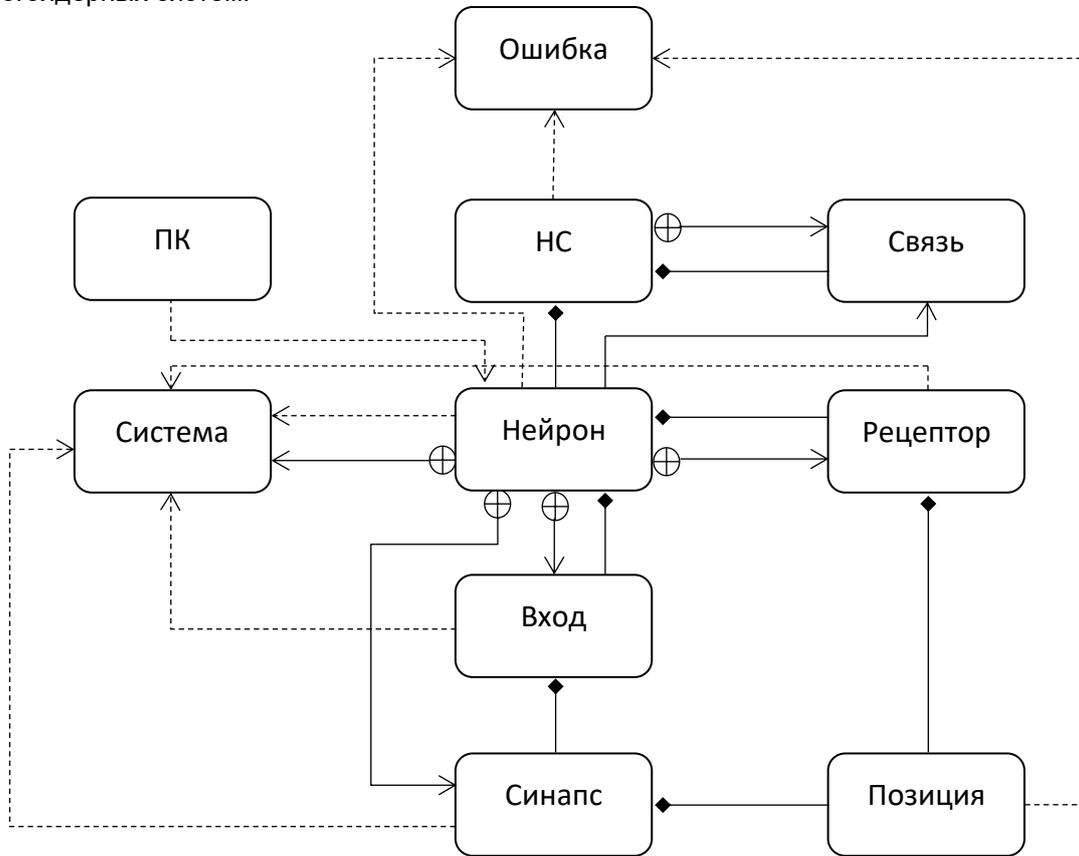


Рисунок 3 – Схема работы библиотеки прецизионной ИНС

Резюмируя, следует сказать, что прецизионная модель искусственной нейронной сети может успешно использоваться для решения задач классификации данных в связи с тем, что она имеет ряд преимуществ в сравнении с классическими ИНС. Во-первых, ей необходимо меньшее количество данных для обучения, вплоть до одной выборки на класс, чтобы сеть могла распознать их. Во-вторых, эта модель имеет меньшее количество изменяемых параметров и их изменения видны. Также она может быть дополнительно обучена и имеет архитектурную возможность распараллеливать вычисления применительно к нейронам и рецепторам. Описанная выше модель ИНС способна заменить сверточные нейронные сети, так как с ее помощью будет возможность распознавать образы, что дает новые возможности для реализации практических задач, к примеру, диагностика аномалий в организме человека [6] или динамический анализ данных в системе реального времени [7]. Так, решение указанных задач с использованием оптимизированных под них классических нейронных сетей дает точность на уровне 80-90%, что недостаточно, например, для клинического применения в медицине, при этом нет возможности дополнительно обучать модель в процессе использования и повышать точность распознавания. Повысить точность результатов в таких случаях сможет применение прецизионной ИНС.

Список использованных источников:

1. Kumar, S. Are machine learning based intrusion detection system always secure? An insight into tampered learning / Kumar S., Hemanta K., Biju I. – *Journal of Intelligent & Fuzzy Systems*, 2018. – С. 35-38.
2. Wu Songyang, Wu Effective detection of android malware based on the usage of data flow APIs and machine learning / Wu Songyang Wu., Pan W., Kumar S., Hemanta K., Biju I., Zhang Yong X. – *Information and Software Technology*, 2016. – С. 75.
3. Shifei, D. Extreme learning machine and its applications / Shifei D., Xinzheng X., Ru N. – *Neural Computing and Applications*, 2014. – С. 25.

4. Indolia, S. *Conceptual Understanding of Convolutional Neural Network – A Deep Learning Approach* / Indolia S., Goswami A.K., Mishra S.P., Asopa P. – *Procedia Computer Science*, 2018. – С. 679-688.

5. Хайкин, С. *Нейронные сети: полный курс* / С.Хайкин – 2-е издание, 2019. – С. 1104.

6. Лукас, А. *Диагностика патологии ЭЭГ на основе машинного обучения* / Лукас А.В. Гемейн, Робин Т. Ширрмейстер, Патрик Храбонц, Дэниел Уилсон, Йошка Бедкер, Андреас Шульце-Бонхаге, Франк Хаттер, Тонио Болл. – *NeuroImage*, 2020.

7. Останин, М. *Распознавание объектов на изображениях высокого разрешения с помощью нейронной сети* / Молодёжь. Техника. Космос: Материалы XI Всероссийской молодежной научно-технической конференции, БГТУ «Военмех», 2019. – С. 485.

UDC 004.032.26

DATA CLASSIFICATION WITH A PRECISION NEURAL NETWORK MODEL

Derenchuk V.I., Boltak S.V., Rabinkin H.M.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Smolyakova O.G. – PhD in Technical Sciences

Abstract. Classical neural networks of our time have a number of disadvantages. They affect the accuracy of the results of their work. In this paper, we will characterize a model and describe the principles of operation of a neural network based on the structure of biological neurons of the human brain, which levels out classical neural networks and is successfully suitable for solving problems of classification, pattern recognition, and others.

Keywords. Neural network, precision neural network model, artificial intelligence, machine learning, data classification, image recognition, synapse, receptor.

ПРОГРАММНОЕ СРЕДСТВО МОНИТОРИНГА ИНДИВИДУАЛЬНЫХ ЗАНЯТИЙ ПО ИНОСТРАННОМУ ЯЗЫКУ

Ермакович К. В.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Деменковец Д.В. – ст. преподаватель

В докладе рассматривается тематика автоматизации мониторинга в области организации индивидуальных занятий по иностранному языку. Предлагаются возможные варианты использования и улучшения разработанного программного средства.

Изучение иностранных языков является одним из основополагающих факторов развития человека, особенно занятого в IT-сфере.

Одним из вариантов изучения иностранного языка являются индивидуальные занятия с преподавателем, на которых учащийся вправе сам выбирать темы для разговоров для большего вовлечения в учебный процесс [1]. Также учащийся имеет возможность согласовывать каждое занятие по времени, по дате или отменить его в случае каких-либо обстоятельств. При этом подавляющее большинство языковых школ предоставляют групповые занятия строго по расписанию и без возможности отмены.

Все большее количество людей как преподавателей, так и учащихся, детей и взрослых, стали выбирать формат удаленных занятий. Такой формат позволяет заниматься в любых локациях, где есть доступ к интернету. Также существенным плюсом является возросшая конкуренция среди преподавателей. Каждый может проанализировать своих конкурентов, изучив их профили и достижения, что мотивирует преподавателя развиваться в качестве носителя языка.

Данная работа подразумевает создание программного средства, в котором учтены недостатки существующих аналогов на белорусском рынке. Основная цель проекта заключается в том, чтобы обеспечить контроль над расписанием занятий как для преподавателя, так и для учащегося.

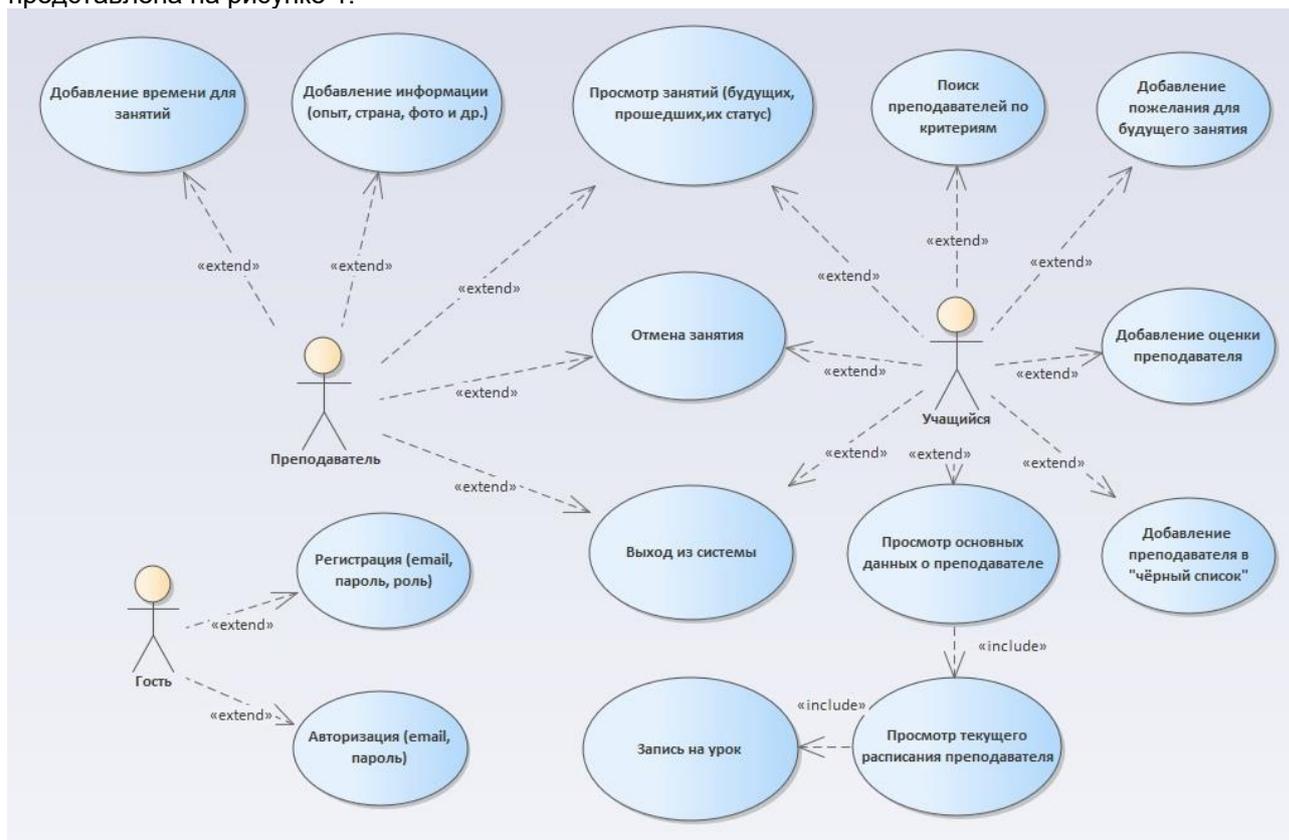
Основные функциональные возможности программного средства мониторинга индивидуальных занятий по иностранному языку представлены ниже:

- Регистрация пользователя как в качестве преподавателя, так и в качестве учащегося. Первым это позволяет иметь доход, вторым – широкий выбор преподавателей.
- Поиск преподавателей по различным критериям, которые учащийся может выбрать для себя. Например, его интересуют преподаватели только из Беларуси, которые предоставляют время для занятий в понедельник в определенное время.
- Гибкий график работы преподавателей. Возможность самим определять время для занятий по мере своего желания.
- Мониторинг занятий преподавателя и учащегося, прошедшие, будущие занятия, их статус.
- Возможность оставлять пожелания и рекомендации к проведению будущего урока учащимся. Данный комментарий отобразится в уведомлениях преподавателя с минимальной задержкой при условии настройки push-уведомлений для почтового сервиса. Преподаватель сможет скорректировать занятие с учётом пожеланий учащегося.
- Уведомления, приходящие на электронную почту, указанную при регистрации. При отмене занятия учащимся, преподаватель получает соответствующее письмо.
- Поддержка «чёрных списков». Любой учащийся может внести преподавателя в специальный список, и при последующем поиске занятий данный преподаватель не будет виден в результате поиска, даже если он подходит по выбранным параметрам.
- Поддержка фильтра интересов преподавателей. При регистрации преподаватель указывает темы, которые ему наиболее интересны для обсуждения на занятиях. Таким образом, учащийся сразу может проанализировать и оценить свою вовлеченность в учебный процесс.

Разработанное программное средство разворачивается с помощью системы контейнеризации Docker [2]. Производится настройка трех контейнеров: первый – с SQL базой данных, второй – с серверной частью приложения на основе языка программирования Java, третий – с клиентской частью на основе фреймворка Angular [3, 4]. С использованием Docker система становится изолированной, гибкой, безопасной и масштабируемой.

Применение программного средства обеспечит внутренний рынок удобным средством для изучения иностранных языков, а также позволит взаимодействовать учащимся и преподавателям без участия третьих лиц. Специализация на иностранные языки позволит обратить внимание на

детали, связанные конкретно с изучением языков. Диаграмма вариантов использования представлена на рисунке 1.



Список использованных источников:

1. 10 причин начать изучать иностранный язык [Электронный ресурс]. – Режим доступа: <https://mti.edu.ru/ob-institute/poleznye-stati/10-prichin-nachat-izuchat-inostrannyj-yazyik>. – Дата доступа: 15.03.2023.
2. Docker(докер): что это такое и для чего нужны контейнеры [Электронный ресурс]. – Режим доступа: <https://blog.skillfactory.ru/glossary/docker/#:~:text=Docker>. – Дата доступа: 18.03.2023.
3. Java [Электронный ресурс]. – Режим доступа: <https://www.oracle.com/cis/java/>. – Дата доступа: 18.03.2023.
4. Angular. The web development framework for building the future [Электронный ресурс]. – Режим доступа: <https://angular.io/>. – Дата доступа: 17.03.2023.

РЕАЛИЗАЦИЯ АЛГОРИТМА ОПРЕДЕЛЕНИЯ МЕРЫ РАЗЛИЧИЯ ДВОИЧНЫХ ТЕСТОВЫХ НАБОРОВ

Ильина А.Ю., магистрант

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Ярмолик В.Н. – д-р. техн. наук, профессор

Аннотация. Рассматривается алгоритм определения меры различия $AD(T_i, T_k)$ двоичных тестовых наборов равной длины. Описывается реализация данного алгоритма с использованием битовых масок.

Ключевые слова. Мера различия, расстояние, тестовый набор, управляемое вероятностное тестирование.

Вероятностное тестирование (random testing) – одна из распространенных технологий тестирования по методу черного ящика [1, 2]. При вероятностном тестировании на исследуемый объект подаются случайные или псевдослучайные входные последовательности [1]. Таким образом, тестовые случаи выбираются, генерируются случайным образом из набора входных данных [3, 4, 5]. Вероятностное тестирование широко применяется на практике [1, 6].

Для повышения эффективности вероятностных тестов используется управляемое вероятностное тестирование (controlled random testing) [2, 5]. Под управляемым вероятностным тестированием понимается случайная тестовая последовательность, в которой очередной тестовый набор формируется с учетом ранее сгенерированных предыдущих наборов [1]. Ключевая особенность контролируемого генерирования случайных тестовых наборов – информация, которая извлекается в виде определенных характеристик (метрик) из ранее сгенерированных тестовых наборов. Эта информация используется для формирования очередного тестового набора [1, 5, 7, 8].

Главный недостаток управляемого вероятностного тестирования заключается в сложности определения очередного тестового набора по отношению к предыдущим наборам теста [1, 2]. В управляемом вероятностном тестировании очередной тестовый набор формируется максимально отличным от всех наборов, сгенерированных ранее [1].

При контролируемом генерировании случайных тестовых наборов в качестве меры различия (distance measure) чаще всего используются расстояние Хэмминга и декартово (евклидово) расстояние [1]. Предложена новая мера различия $AD(T_i, T_k)$ двух символьных тестовых наборов на основе интервала, используемого в теории строки цепи последовательных событий [7, 8]. В качестве меры различия рассчитывается расстояние $AD(T_i, T_k)$ между тестовыми наборами T_i и T_k , использующее характеристику интервала и основанное на определении независимых пар одинаковых (тождественных) символов, принадлежащих двум наборам, и вычислении интервалов между ними. Предложенная мера различия расширяет возможности генерирования тестовых последовательностей при генерировании управляемых вероятностных тестов [8].

Для двоичных тестовых наборов задача вычисления метрики $AD(T_i, T_k)$ сводится к задаче о назначении. Данная задача решается с помощью венгерского алгоритма, вычислительная сложность которого оценивается соотношением $O(n^4)$. Как альтернатива венгерскому алгоритму предложен алгоритм определения $AD(T_i, T_k)$ – алгоритм вычисления рассматриваемой меры, вычислительная сложность которого существенно меньше и имеет оценку $O(n^2)$ [8].

Для случая двоичных данных равной длины исходными являются тестовые наборы T_i и T_k , состоящие из n данных $t_{i,j}, t_{k,r} \in \{0, 1\}$, $j, r \in \{0, 1, \dots, n-1\}$.

Сущность предложенного алгоритма вычисления меры различия $AD(T_i, T_k)$ заключается в определении количества тождественных пар данных $t_{i,j}$ и $t_{k,r}$ тестовых наборов T_i и T_k , имеющих определенное значение расстояния $D(t_{i,j}, t_{k,r})$. Учитывая, что длина тестовых наборов T_i и T_k одинакова и равна n , значение $D(t_{i,j}, t_{k,r})$ равняется минимальному значению из двух величин $|j-r|$ и $n-|j-r|$. Тогда необходимо рассмотреть только $\lfloor n/2 \rfloor + 1$ возможных значений $D(t_{i,j}, t_{k,r}) = 0 \div \lfloor n/2 \rfloor$, где квадратные скобки обозначают целую часть числа [8].

Генерируются циклические сдвиги T_i влево и вправо с последующим определением количества совпадающих данных, имеющих одинаковые индексы $j=r$. В результате операций циклического сдвига T_i индекс j для каждого данного $t_{i,j}$ набора T_i принимает все возможные значения $j \in \{0, 1, \dots, n-1\}$. Таким образом, каждому данному $t_{k,r}$ набора T_k последовательно сопоставляются все данные $t_{i,j}$ и анализируются на предмет совпадения. Совпадение данных $t_{i,j}$ и $t_{k,r}$ свидетельствует о наличии пары тождественных данных с расстоянием $D(t_{i,j}, t_{k,r})$, равным количеству циклических сдвигов набора T_i . Независимость пар тождественных данных обеспечивается исключением данных, входящих в выявленную пару, из дальнейшего рассмотрения [8].

Алгоритм определения меры различия $AD(T_i, T_k)$ двоичных данных равной длины состоит из следующих шагов [7, 8]:

1 В исходных тестовых наборах T_i и T_k определяются совпадающие данные $t_{i,j}$ и $t_{k,r}$, имеющие расстояние $D(t_{i,j}, t_{k,r}) = 0$ и число совпадающих данных $q(0)$. Выделенные данные исключаются из дальнейшего рассмотрения. В случае, когда $q(0) = n$, т. е. при полном совпадении данных набора T_i с данными набора T_k , задается $q(t) = 0$, и осуществляется переход к пункту 5.

2 Если количество данных n (длина тестовых наборов) является нечетным числом, то последовательно для $v = 1, 2, \dots, [n/2]$ формируются циклические сдвиги тестового набора T_i относительно набора T_k на v позиций влево и вправо. Следует отметить, что операция циклического сдвига эквивалентна уменьшению и увеличению значения индекса j на величину v по модулю n . Далее определяется количество $q(v)$ тождественных пар данных с совпадающими индексами, а именно с модифицированным в результате сдвига $j = (j \pm v) \bmod n$ и индексом r . Величина $q(v)$ определяет количество пар данных $t_{i,j}$ и $t_{k,r}$, имеющих расстояние $D(t_{i,j}, t_{k,r}) = v$. При рассмотрении очередного значения сдвига v набора T_i выделенные тождественные данные исключаются из дальнейшего рассмотрения. Выполнение данного шага прекращается в случае, если количество пар эквивалентных данных, выявленных в наборах T_i и T_k , достигло величины n (т. е. все данные в наборах совпали) и после задания $q(t) = 0$ осуществляется переход к пункту 5.

3 Если n четное число, выполняются действия шага 2, однако вместо пары циклических сдвигов тестового набора T_i на $v = [n/2]$ влево и вправо формируется единственный циклический сдвиг на $v = [n/2]$ позиций влево либо вправо.

4 Определяется количество $q(t)$ пар данных $t_{i,j}$ и $t_{k,r}$ наборов T_i и T_k , которые не участвовали в формировании пар тождественных данных на предыдущих шагах алгоритма. При равенстве числа данных $t_{i,j}$ и $t_{k,r}$ в наборах T_i и T_k для каждого из них существует пара либо тождественных данных, либо несовпадающих [8].

5 Вычисляется значение меры различия $AD(T_i, T_k)$. Для двоичных тестовых наборов равной длины n в случае несовпадения данных расстояние между ними считается равным значению $[n/2]$, поэтому значение меры различия $AD(T_i, T_k)$ вычисляется согласно выражению [8]

$$AD(T_i, T_k) = q(t) [n/2] + \sum_{v=1}^{[n/2]} v \cdot q(v). \quad (1)$$

Полученное значение $AD(T_i, T_k)$ является искомой величиной меры различия наборов T_i и T_k .

В таблице 1 представлено применение алгоритма определения $AD(T_i, T_k)$ для случая восьмибитных двоичных тестовых наборов $T_i = 00010111$ и $T_k = 11101010$.

Таблица 1 – Применение алгоритма определения меры различия $AD(T_i, T_k)$ для восьмибитных двоичных тестовых наборов $T_i = 00010111$ и $T_k = 11101010$.

Номер шага и величина сдвига влево или вправо v	Данные тестового набора	Данные тестового набора с индексами позиций j для t_i и r для t_k								Количество совпадений на данном шаге $q(v)$ и после выполнения данного шага $ov(v)$
		0	1	2	3	4	5	6	7	
$j = (j - v) \bmod 8 = r$		0	1	2	3	4	5	6	7	
Шаг 0, $v_0 = 0$	$t_{i,j}$	0 ₀	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	$q(v_0) = 1,$ $ov(v_0) = 1$
	$t_{k,r}$	1 ₀	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
Шаг 1, $v_{left,1} = 1$	$t_{i,j (left,1)}$	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	0 ₀	$q(v_{left,1}) = 4,$ $ov(v_{left,1}) = 5$
	$t_{k,r}$	1 ₀	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
Шаг 2, $v_{right,1} = 1$	$t_{i,j (right,1)}$	1 ₇	0 ₆	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	$q(v_{right,1}) = 1,$ $ov(v_{right,1}) = 6$
	$t_{k,r}$	1 ₀	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
Шаг 3, $v_{left,2} = 2$	$t_{i,j (left,2)}$	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	0 ₀	0 ₁	$q(v_{left,2}) = 0,$ $ov(v_{left,2}) = 6,$
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
Шаг 4, $v_{right,2} = 2$	$t_{i,j (right,2)}$	1 ₆	1 ₇	0 ₀	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	$q(v_{right,2}) = 0,$ $ov(v_{right,2}) = 6$
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
Шаг 5, $v_{left,3} = 3$	$t_{i,j (left,3)}$	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	0 ₀	0 ₁	0 ₂	$q(v_{left,3}) = 0,$ $ov(v_{left,3}) = 6$
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
Шаг 6, $v_{right,3} = 3$	$t_{i,j (right,3)}$	1 ₅	1 ₆	1 ₇	0 ₀	0 ₁	0 ₂	1 ₃	0 ₄	$q(v_{right,3}) = 1,$ $ov(v_{right,3}) = 7$
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
Шаг 7, $v_{left,4} = 4$ ($v_{right,4} = 6$)	$t_{i,j (left,4)}$ ($t_{i,j (right,4)}$)	0 ₄	1 ₅	1 ₆	1 ₇	0 ₀	0 ₁	0 ₂	1 ₃	$q(v_{left,4}) = q(v_{right,4}) = 0,$ $ov(v_{left,4}) = ov(v_{right,4}) = 7$
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	

В таблице 1 подчеркнуты и выделены полужирным шрифтом совпадающие на текущем шаге данные тестовых наборов, неиспользованные ранее. Зачеркнуты ячейки, содержащие данные, совпавшие на более ранних шагах алгоритма.

Алгоритм определения меры различия $AD(T_i, T_k)$ может быть реализован с использованием битовых масок. Битовая маска – последовательность бит, применяемая для маскирования, т. е. выбора отдельных бит из двоичных данных.

В описываемой реализации алгоритма определения $AD(T_i, T_k)$ используются две битовые маски M_i и M_k для выполнения операций над тестовыми наборами T_i и T_k . Длины масок равны длине тестовых наборов и составляют n . Тестовые наборы состоят из данных $t_{i,j}, t_{k,r} \in \{0, 1\}$ маски – из данных $m_{i,j}, m_{k,r} \in \{0, 1\}$, причем $j, r \in \{0, 1, \dots, n - 1\}$.

Шаги алгоритма определения меры различия $AD(T_i, T_k)$ представляются следующим образом:

1 Искомая мера различия AD инициализируется нулем, а данные маски M_i , соответствующей тестовому набору T_i – единицами. Все единицы в маске M_i означают, что данные набора T_i не были использованы в тождественных парах ранее.

2 Вычисляется вспомогательный набор $result$. Значение $result$ рассчитывается по формуле $result = NOT(T_i XOR T_k)$. Данное выражение означает следующее. Операция исключающее ИЛИ применяется к тестовым наборам T_i и T_k для выделения тождественных пар данных. Таким образом получается набор данных длины n , где совпадающие пары данных тестовых наборов T_i и T_k обозначены нулями, а несовпадающие – единицами на соответствующих позициях. К полученному результату применяется инверсия (операция логическое НЕ), и совпадающие пары данных тестовых наборов T_i и T_k в конечном наборе обозначены единицами, а несовпадающие – нулями.

3 Количество совпавших пар данных $overlap$ инициализируется значением, равным количеству единиц в найденном на шаге 2 наборе $result$.

4 Находится новая маска M_i по формуле $M_i = M_i XOR result$. Операция исключающее ИЛИ обнуляет значения единиц, обозначающих найденные пары совпадающих данных в маске, что позволяет отметить их как исключенные из рассмотрения. Маска M_k , соответствующая тестовому набору T_k , инициализируется найденным новым значением маски M_i , поскольку до текущего шага сдвигов не производилось, и значения масок M_i и M_k совпадают.

5 Если длина тестовых наборов n – нечетное число, последовательно для $v = 1, 2, \dots, [n / 2]$ и направлений влево и вправо выполняется процедура нахождения количества пар тождественных данных, находящихся на определенном расстоянии друг от друга. Выполнение данного шага прекращается, если маски M_i и M_k полностью обнулились, т. е. все данные наборов были использованы. Следует отметить, что обнуление обеих масок происходит одновременно.

6 Если длина тестовых наборов n – четное число, выполняются действия шага 5, однако последовательно для $v = 1, 2, \dots, [n / 2] - 1$ влево и вправо и для $v = [n / 2]$ влево либо вправо.

7 Находится разность между длиной набора n и количеством совпавших пар данных $overlap$. Значение меры AD увеличивается на произведение найденной разности и значение $[n / 2]$.

Выполнение шагов 5 и 6 алгоритма основано на процедуре нахождения количества пар тождественных данных, находящихся на определенном расстоянии друг от друга. На входе процедура получает тестовые наборы T_i и T_k , соответствующие им маски M_i и M_k , направление сдвига $direction$ (влево или вправо) и количество позиций v , на которые осуществляется сдвиг.

Алгоритм нахождения количества пар тождественных данных, находящихся на определенном расстоянии друг от друга состоит из следующих шагов:

1 Формируются циклические сдвиги $T_{i(direction,v)}$ и $M_{i(direction,v)}$ тестового набора T_i и маски M_i в заданном направлении $direction$ (влево или вправо) на заданное количество позиций v .

2 Вычисляется вспомогательный набор $result$. Значение $result$ рассчитывается по формуле $result = (NOT(T_{i(direction,v)} XOR T_k)) AND M_{i(direction,v)} AND M_k$. Данное выражение, по сути, означает следующее. Операция исключающее ИЛИ применяется к тестовым наборам $T_{i(direction,v)}$ и T_k с целью выделения тождественных пар данных. В результате выполнения операции $T_{i(direction,v)} XOR T_k$ получается набор данных длины n , совпадающие пары данных тестовых наборов $T_{i(direction,v)}$ и T_k обозначены нулями, а несовпадающие – единицами. К полученному результату применяется инверсия. Результатом выполнения операций $NOT(T_{i(direction,v)} XOR T_k)$ является набор данных длины n , где совпадающие пары данных тестовых наборов $T_{i(direction,v)}$ и T_k обозначены уже единицами, а несовпадающие – нулями. Далее к полученному набору применяется операция логического И с масками $M_{i(direction,v)}$ и M_k . Это действие необходимо для исключения из конечного результата тождественных пар данных, уже учтенных на более ранних шагах алгоритма. Поскольку в масках единицами обозначены еще не использованные данные, а нулями – уже учтенные и потому исключенные из рассмотрения, после выполнения операций логического И в значении $result$ единицами обозначаются только пары совпадающих данных, которые не были использованы ранее. Все остальные пары данных в $result$ отмечены нулями. Таким образом, в наборе $result$ единицами обозначаются обнаруженные пары эквивалентных данных, которые не были исключены из рассмотрения, а нулями – все прочие пары данных.

3 Формируются новые маски $M_i^{(direction,v)}$ и M_k с учетом найденных на текущем шаге пар совпадающих данных, которые не были учтены ранее. Новые маски вычисляются следующим образом: $M_i^{(direction,v)} = M_i^{(direction,v)} \text{ XOR result}$; $M_k = M_k \text{ XOR result}$. Операция исключающее ИЛИ обнуляет значения единиц, обозначающих найденные пары тождественных данных в масках, что позволяет отметить использованные на текущем шаге данные как исключенные из рассмотрения.

4 Находится количество единиц во вспомогательном наборе result, т. е. количество найденных на текущем шаге пар совпадающих данных, которые не использовались на более ранних шагах.

5 Рассчитывается слагаемое метрики AD, полученное на текущем шаге алгоритма. Для этого количество единиц в наборе result умножается на заданное количество позиций сдвига v, равное расстоянию между тождественными данными. Т. е. количество пар совпадающих данных, найденных на текущем шаге, не исключенных ранее, умножается на расстояние между совпадающими данными.

С помощью описанной процедуры находятся значения, на которые необходимо увеличить метрику AD и количество совпавших пар данных overlap. Процедура вычисляет маски M_i и M_k , необходимые для следующих шагов алгоритма. Следует учесть, что в процедуре находится $M_i^{(direction,v)}$, т. е. маска M_i , циклически сдвинутая на v позиций влево или вправо.

В таблице 2 приведено применение реализации алгоритма определения $AD(T_i, T_k)$ с использованием битовых масок для случая, рассмотренного в таблице 1 – восьмибитные двоичные тестовые наборы $T_i = 00010111$ и $T_k = 11101010$.

Таблица 2 – Применение реализации алгоритма определения меры различия $AD(T_i, T_k)$ с использованием битовых масок для восьмибитных двоичных тестовых наборов $T_i = 00010111$ и $T_k = 11101010$.

Номер шага и величина сдвига влево или вправо v	Данные тестового набора, маски или вспомогательного набора result	Данные тестового набора, маски или вспомогательного набора result с индексами позиций j для t_i и r для t_k								Количество совпадений на данном шаге $q(v)$ и после выполнения данного шага $ov(v)$
		0	1	2	3	4	5	6	7	
$j = (j - v) \text{ mod } 8 = r$		0	1	2	3	4	5	6	7	
Шаг 0, $v_0 = 0$	$t_{i,j}$	0 ₀	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	$q(v_0) = 1,$ $ov(v_0) = 1$
	$t_{k,r}$	1 ₀	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
	$m_{i,j(k,r)}$	1 ₀	1 ₁	1 ₂	1 ₃	1 ₄	1 ₅	1 ₆	1 ₇	
	result _{r(0)}	0 ₀	0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	1 ₆	0 ₇	
Шаг 1, $v_{left,1} = 1$	$t_{i,j(left,1)}$	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	0 ₀	$q(v_{left,1}) = 4,$ $ov(v_{left,1}) = 5$
	$m_{i,j(left,1)}$	1 ₁	1 ₂	1 ₃	1 ₄	1 ₅	0 ₆	1 ₇	1 ₀	
	$t_{k,r}$	1 ₀	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
	$m_{k,r}$	1 ₀	1 ₁	1 ₂	1 ₃	1 ₄	1 ₅	0 ₆	1 ₇	
Шаг 2, $v_{right,1} = 1$	$t_{i,j(right,1)}$	1 ₇	0 ₆	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	$q(v_{right,1}) = 1,$ $ov(v_{right,1}) = 6$
	$m_{i,j(right,1)}$	1 ₇	0 ₀	1 ₁	1 ₂	0 ₃	0 ₄	0 ₅	0 ₆	
	$t_{k,r}$	1 ₀	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
	$m_{k,r}$	1 ₀	1 ₁	0 ₂	0 ₃	0 ₄	1 ₅	0 ₆	0 ₇	
Шаг 3, $v_{left,2} = 2$	$t_{i,j(left,2)}$	0 ₂	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	0 ₀	0 ₁	$q(v_{left,2}) = 0,$ $ov(v_{left,2}) = 6,$
	$m_{i,j(left,2)}$	1 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇	0 ₀	1 ₁	
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
	$m_{k,r}$	0 ₀	1 ₁	0 ₂	0 ₃	0 ₄	1 ₅	0 ₆	0 ₇	
Шаг 4, $v_{right,2} = 2$	$t_{i,j(right,2)}$	1 ₆	1 ₇	0 ₀	0 ₁	0 ₂	1 ₃	0 ₄	1 ₅	$q(v_{right,2}) = 0,$ $ov(v_{right,2}) = 6$
	$m_{i,j(right,2)}$	0 ₆	0 ₇	0 ₀	1 ₁	1 ₂	0 ₃	0 ₄	0 ₅	
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
	$m_{k,r}$	0 ₀	1 ₁	0 ₂	0 ₃	0 ₄	1 ₅	0 ₆	0 ₇	
Шаг 5, $v_{left,3} = 3$	$t_{i,j(left,3)}$	1 ₃	0 ₄	1 ₅	1 ₆	1 ₇	0 ₀	0 ₁	0 ₂	$q(v_{left,3}) = 0,$ $ov(v_{left,3}) = 6$
	$m_{i,j(left,3)}$	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇	0 ₀	1 ₁	1 ₂	
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
	$m_{k,r}$	0 ₀	1 ₁	0 ₂	0 ₃	0 ₄	1 ₅	0 ₆	0 ₇	
Шаг 6, $v_{right,3} = 3$	$t_{i,j(right,3)}$	1 ₅	1 ₆	1 ₇	0 ₀	0 ₁	0 ₂	1 ₃	0 ₄	$q(v_{right,3}) = 1,$ $ov(v_{right,3}) = 7$
	$m_{i,j(right,3)}$	0 ₅	0 ₆	0 ₇	0 ₀	1 ₁	1 ₂	0 ₃	0 ₄	
	$t_{k,r}$	1 ₆	1 ₁	1 ₂	0 ₃	1 ₄	0 ₅	1 ₆	0 ₇	
	$m_{k,r}$	0 ₀	1 ₁	0 ₂	0 ₃	0 ₄	1 ₅	0 ₆	0 ₇	
Шаг 6, $v_{right,3} = 3$	result _{r(right,3)}	0 ₀	0 ₁	0 ₂	0 ₃	0 ₄	1 ₅	0 ₆	0 ₇	

Шаг 7, $V_{\text{left},4} = 4$ ($V_{\text{right},4} = 6$)	$t_{i,j} (\text{left},4)$ ($t_{i,j} (\text{right},4)$)	0₄	1₅	1₆	1₇	0₆	0 ₁	0₂	1₃	$q(V_{\text{left},4}) =$ $= q(V_{\text{right},4}) = 0,$ $ov(V_{\text{left},4}) =$ $= ov(V_{\text{right},4}) = 7$
	$m_{i,j} (\text{left},4)$ ($m_{i,j} (\text{right},4)$)	0 ₄	0 ₅	0 ₆	0 ₇	0 ₀	1 ₁	0 ₂	0 ₃	
	$t_{k,r}$	1₆	1 ₁	1₂	0₃	1₄	0₅	1₆	0₇	
	$m_{k,r}$	0 ₀	1 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇	
	$result_r (\text{left},4)$ ($result_r (\text{right},4)$)	0 ₀	0 ₁	0 ₂	0 ₃	0 ₄	0 ₅	0 ₆	0 ₇	

По сравнению с таблицей 1, в таблице 2 помимо тестовых наборов T_i и T_k рассмотрены также маски M_i и M_k и вспомогательный набор $result$.

Как и в таблице 1, в таблице 2 совпадающие на текущем шаге данные тестовых наборов, неиспользованные ранее, подчеркнуты и выделены полужирным шрифтом. Таким же образом отмечены и соответствующие данные масок и результатов выполнения шагов. Зачеркнуты ячейки, содержащие совпавшие на более ранних шагах алгоритма данные. Следует отметить, что нулевые данные в масках соответствуют зачеркнутым ячейкам, содержащим исключенные из дальнейшего рассмотрения данные тестовых наборов $t_{i,j}$ и $t_{k,r}$. Единичные данные в масках соответствуют не зачеркнутым ячейкам, содержащим доступные для рассмотрения данные.

Рассмотрена реализация алгоритма определения меры различия $AD(T_i, T_k)$ для случая двоичных тестовых наборов равной длины с использованием битовых масок.

Совпадающие данные идентифицируются с помощью логических операций исключающее ИЛИ (XOR), НЕ (NOT), И (AND) над тестовыми наборами T_i , T_k и масками M_i , M_k . Над тестовым набором T_i и соответствующей ему маской M_i выполняются операции циклического сдвига.

При определении совпадающих данных в тестовых наборах T_i и T_k единичное значение в маске указывает на доступность данного под тем же номером в соответствующем тестовом наборе. Нулевое значение в маске сигнализирует об исключении данного соответствующего набора из рассмотрения по причине его использования в тождественной паре. Маски M_i и M_k – индикаторы использования данных тестовых наборов в совпадающих парах. Когда все данные задействованы в тождественных парах, обе маски одновременно обнуляются. Последовательное формирование циклических сдвигов тестового набора T_i относительно набора T_k останавливается при равенстве значения любой из масок (обеих масок) нулю.

Значение меры различия AD и количества использованных в тождественных парах данных $overlap$, необходимое для вычисления итогового значения меры различия AD , последовательно накапливаются в шагах алгоритма. Отдельное значение $overlap$ на определенном шаге вычисляется как количество единиц в наборе бит, что может быть реализовано на основе сдвигов и/или логической операции И.

Двоичные тестовые наборы данных равной длины – распространенный случай требуемых на практике тестовых данных. В рассмотренной реализации алгоритма определения меры различия $AD(T_i, T_k)$ используются сдвиги чисел и логические операции XOR, NOT, AND над числами. Вычислительная сложность выполнения сдвигов и логических операций в сравнении, например, с реализацией алгоритма на массивах, крайне низкая. Следует отметить, что в большинстве современных вычислительных систем используемые в алгоритме битовые сдвиги и побитовые логические операции реализованы на аппаратном уровне.

Список использованных источников:

1. Ярмолик, В.Н. Контроль и диагностика вычислительных систем / В.Н. Ярмолик. – Минск : Бестпринт, 2019. – 387 с.
2. Леванцевич, В.А. Многократное управляемое вероятностное тестирование / В.А. Леванцевич, В.Н. Ярмолик // Доклады БГУИР. – 2019. – № 3 (121). – С. 65-69.
3. A cost-effective software testing strategy employing online feedback information / Z.Q. Zhou [et al.] // Information Sciences. – 2018. – Vol. 422. – P. 318-335.
4. An empirical comparison of combinatorial testing, random testing and adaptive random testing / H. Wu [et al.] // IEEE Transactions on Software Engineering. – 2020. – Vol. 46, № 3. – P. 302-320.
5. Ильина, А.Ю. Управляемое вероятностное тестирование / А.Ю. Ильина // Современная наука: актуальные вопросы, достижения и инновации: сборник статей XXVIII Международной научно-практической конференции, Пенза, 10 декабря 2022 г. : в 2 ч. / МЦНС «Наука и Просвещение»; редкол.: Г.Ю. Гуляев [и др.]. – Пенза, 2022. – Ч. 1. – С. 28-30.
6. A Survey on Adaptive Random Testing / R. Huang [et al.] // IEEE Transactions on Software Engineering. – 2021. – Vol. 47, № 10. – P. 2052-2083.
7. Ярмолик, В.Н. Мера отличия для управляемых вероятностных тестов / В.Н. Ярмолик, Н.А. Шевченко, В.В. Петровская // Доклады БГУИР. – 2022. – Т. 20, № 6. – С. 52-60.
8. Ярмолик, В.Н. Мера различия для тестовых наборов при генерировании управляемых вероятностных тестов / В.Н. Ярмолик, В.В. Петровская, И. Мрозек // Информатика. – 2022. – № 19 (4). – С. 7-26.

UDC 004.415.533

IMPLEMENTATION OF THE ALGORITHM FOR DETERMINING THE DISTANCE MEASURE OF BINARY TEST SETS

Ilyina A. Y.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Yarmolik V.N. – D. Sc. (Eng.), Professor

Annotation. An algorithm for determining the distance measure $AD(T_i, T_k)$ of binary test sets of equal length is considered. The implementation of this algorithm using bitmasks is described.

Keywords. Distance measure, distance, test set, controlled random testing.

ПРОГРАММНОЕ СРЕДСТВО «МЕНЕДЖЕР РАБОЧЕГО ВРЕМЕНИ»

Занько М.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Леванцевич В.А. – старший преподаватель

Рассмотрена актуальность проблемы учета рабочего времени сотрудников предприятия. Предложен подход, который позволит автоматизировать процесс учета рабочего времени и таким образом повысить эффективность работы предприятия. Обосновано использование мобильной версии программного средства, как инструмента контроля рабочего времени

Учет рабочего времени может быть использован для решения различных задач, таких как [1]:

- расчет заработной платы сотрудников на основе отработанных часов;
- планирование графиков работы сотрудников;
- оптимизация рабочих процессов;
- контроль и управление переработками и отсутствием сотрудников;
- улучшение управления персоналом.

Кроме того, учет рабочего времени позволяет организации соблюдать законодательные требования по рабочему времени и отдыху. Например, многие страны имеют законы, регулирующие максимальное количество рабочих часов в неделю и требующие определенных перерывов и отпусков для сотрудников. Учет рабочего времени может помочь организации соблюдать эти требования и избежать штрафов и неприятностей.

В целом, учет рабочего времени является важным инструментом, который может помочь повысить эффективность работы и управление персоналом и сократить затраты на заработную плату.

С помощью программного средства учета рабочего времени можно автоматизировать этот процесс, сделать его более точным и эффективным и обеспечить лучший контроль над рабочим временем сотрудников, а также предоставлять аналитические отчеты и статистику о продуктивности сотрудников, что может быть полезно для принятия управленческих решений. Например, руководство может использовать эти данные для определения наиболее эффективных сотрудников, выявления проблем в рабочих процессах и принятия мер для улучшения производительности.

Программное средство может быть полезно для управления персоналом, особенно в организациях с большим количеством сотрудников, где сложно вести учет рабочего времени вручную, что позволяет снизить вероятность ошибок.

Существует несколько способов внедрения программного обеспечения для учета рабочего времени [1]:

- локальная установка на компьютеры организации;
- облачное хранение и управление на удаленных серверах;
- использование мобильных приложений для смартфонов и планшетов.

Использование мобильных приложений для учета рабочего времени может быть особенно удобным для организаций, у которых значительное количество сотрудников работает удаленно или не постоянно находится в офисе. Мобильное приложение позволяет сотрудникам легко отслеживать свое рабочее время с помощью своих мобильных устройств в любом месте и в любое время. Они могут легко отмечать начало и конец рабочего дня, а также перерывы и отсутствия.

Кроме того, данный способ внедрения позволяет сотрудникам предоставлять отчеты о проделанной работе в режиме реального времени. Это особенно полезно как для менеджеров проектов, так и для заказчиков, которые должны следить за выполнением работ и сроками проекта [2].

Важным преимуществом использования приложения на платформе Android, для учета рабочего времени является их легкость развертывания. Для использования такого приложения, сотрудники могут просто загрузить его на свои мобильные устройства и начать использовать без необходимости установки и настройки сложного программного обеспечения на компьютерах в офисе. Это позволяет сократить время и затраты на развертывание программы и обучение сотрудников ее использованию.

Важно помнить, что при выборе программного обеспечения учета рабочего времени организации должны учитывать свои конкретные потребности и бизнес-цели, а также уделять внимание безопасности данных и соблюдению соответствующих законодательных требований.

Список использованных источников:

Конференция «Компьютерные системы и сети»

1. Searchinform [Электронный ресурс]. Дата обновления: 20.02.2023. URL: <https://searchinform.ru/kontrol-sotrudnikov/uchet-rabochego-vremeni/osobennosti-ucheta-rabochego-vremeni>
2. Habr [Электронный ресурс]. Дата обновления: 11.10.2020. URL: <https://habr.com/ru/post/527614>

ЭТАПЫ И МЕТОДЫ РАСПОЗНАВАНИЯ ЖЕСТОВ БЕЛОРУССКОГО ЖЕСТОВОГО ЯЗЫКА

Ковалева-Евсюченя А.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Бранцевич П.Ю. – канд. техн. наук, доцент

В данной работе рассмотрены этапы и современные методы распознавания жестов. Выбраны технологии для реализации программного средства для распознавания жестов белорусского жестового языка.

Речь – это неотъемлемая часть жизни любого человека. При помощи речи мы каждый день общаемся со своими близкими, выражаем свое мнение и взаимодействуем с окружающим миром. Однако, более 450 миллионов людей в мире страдают от нарушений слуха и полной глухоты и большинство из них не может использовать речь как основное средство коммуникации [1]. Для этой цели они используют жестовые языки. Так как слышащие люди редко владеют жестовыми языками, для коммуникации с ними глухие люди вынуждены прибегать к услугам сурдопереводчиков.

В Беларуси на законодательном уровне закрепили понятие «белорусский жестовый язык» в 2022 году [2]. Государство способствует подготовке сурдопереводчиков и повышению их квалификации, но сейчас существует нехватка специалистов – в реестре переводчиков жестового языка Республики Беларусь зарегистрирован 21 человек [3]. Поэтому, сейчас есть потребность в технологии распознавания белорусского жестового языка, которая сможет помочь в обучении новых специалистов и в будущем дополнить живой сурдоперевод.

Жестовое общение использует верхнюю часть туловища, включая жесты руками, мимику, чтение по губам и положение тела в пространстве. Распознавание жестового языка, основанного на образах, включает в себя пять этапов: получение изображений для составления обучающей выборки; предварительную обработку полученных изображений; сегментацию изображений; извлечение признаков; классификацию образа.

Для получения изображений для распознавания используются различные устройства. Самое широко используемое устройство - веб-камера. Веб-камеру легко использовать, и она встроена в многие устройства. Самым точным устройством считается специальная перчатка, но она обладает высокой стоимостью и может быть некомфортна в использовании. Контроллер Kinect позволяет получать изображение и одновременно траектории движения рук, но он также обладает высокой стоимостью [4].

Предварительная обработка изображений осуществляется с целью убрать посторонние шумы и улучшить качество изображения. Применяемые подходы для улучшения качества изображения включают в себя выравнивание гистограммы (HE), адаптивное выравнивание гистограммы (AHE), контрастно-ограниченное выравнивание гистограммы (CLAHE) и логарифмическое преобразование. Сравнительный анализ методов улучшения качества изображений приведен в таблице 1. Используемые алгоритмы шумоподавления включают в себя медианный фильтр, фильтр Гаусса, фильтр Винера, адаптивный фильтр и фильтр скользящего среднего. Правильный выбор методов обработки изображений является ключевым для корректного распознавания жеста [5].

Таблица 1 – Сравнительный анализ методов улучшения качества изображений.

Метод улучшения качества изображений	Преимущества	Недостатки
Выравнивание гистограммы	Простота реализации. Эффективность при использовании на черно-белых изображениях	Изменение яркости изображений. Повышение контраста посторонних шумов
Адаптивное выравнивание гистограммы	Усиление локального контраста во всех областях изображений	Потеря исходного уровня яркости изображений. Усиление посторонних шумов
Контрастно-ограниченное выравнивание гистограммы	Низкий уровень шумов в изображениях. Сохранение исходной яркости изображений	Потеря информации при использовании изображений с повышенной яркостью
Логарифмическое преобразование	Сжатие динамического диапазона изображений, имеющих большие вариации в значениях пикселей	Потеря информации при использовании изображений, на которых большая часть

		деталей сосредоточена в светлых областях
--	--	--

Сегментация изображений – это разделение изображений на значимые сегменты. Для сегментации изображений используются два подхода: контекстуальный и неконтекстуальный [6]. Контекстуальная сегментация использует отношения между признаками изображения, такими как расстояние между признаками на изображении, контрастность сегментов изображения, границы признаков и другие. Неконтекстуальная сегментация игнорирует признаки отдельных областей изображения и осуществляет разделение, используя глобальные атрибуты изображения. Подробная классификация методов сегментации изображения показана на рисунке 1.

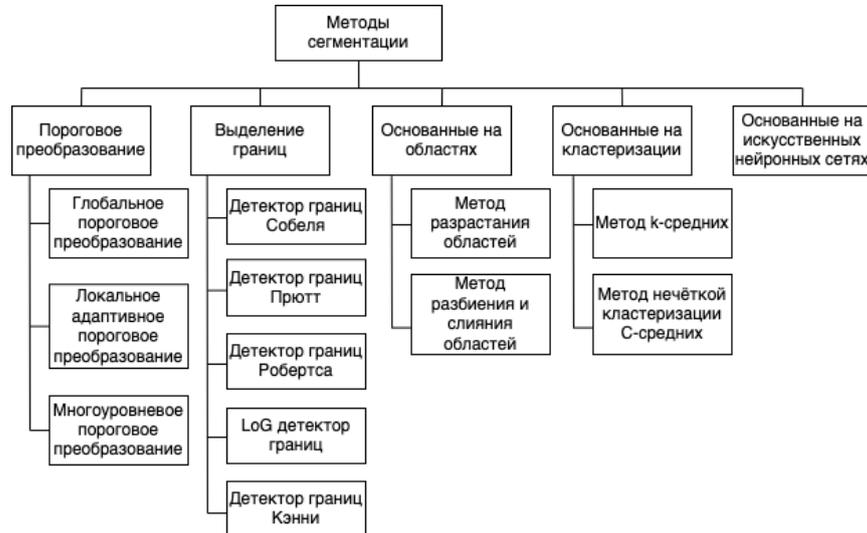


Рисунок 1 – Классификация методов сегментации изображения

На этапе извлечения признаков, для увеличения точности распознавания, выделяются самые релевантные признаки изображения, которые представляют собой отличия между классами распознавания программного средства [7]. Результатом выполнения данного этапа является вектор признаков, который в дальнейшем используется этапом классификации. Важные методы извлечения признаков включают в себя метод главных компонент (PCA), Фурье-дескрипторы (FD), гистограмму направленных градиентов (HOG), масштабно-инвариантную трансформацию признаков (SIFT), ускоренные устойчивые функции (SURF).

При классификации образа, изображение, не входящее в обучающую выборку, поступает в алгоритм классификации. Задачей алгоритма классификации является указание наименования класса на основе имеющегося вектора признаков. Среди существующих классификаторов, алгоритм k-ближайших соседей (KNN), искусственная нейронная сеть (ANN), метод опорных векторов (SVM), скрытая марковская модель (HMM) и сверточная нейронная сеть (CNN) являются основными [8].

Согласно рассмотренным этапам распознавания, оптимальным выбором классификатора для создания программного средства для распознавания жестов белорусского жестового языка является сверточная нейронная сеть в сочетании с масштабно-инвариантной трансформацией признаков. SIFT-CNN автоматически распознает релевантные признаки образа без вмешательства программиста и выполняет классификацию жеста с высокой точностью. Сегментация на основе искусственной нейронной сети позволяет получить качественное разделение на сегменты с низким уровнем шума, несмотря на большую вычислительную сложность. Для получения изображения для распознавания используется веб-камера как самое доступное устройство, с применением фильтра Гаусса для шумоподавления.

Список использованных источников:

1. World report on hearing process / Department of Noncommunicable Diseases ; ed.: Jackie Clark [et al.]. – Geneva : WHO, 2021. – 252 p.
2. О правах инвалидов и их социальной интеграции [Электронный ресурс] : Закон Республики Беларусь, 30 июня 2022 г., № 183-З // Национальный правовой Интернет-портал Республики Беларусь. – Режим доступа: <https://pravo.by/document/?guid=12551&p0=H12200183&p1=1>. – Дата доступа: 10.04.2023.
3. Реестр переводчиков жестового языка Республики Беларусь [Электронный ресурс]. – Режим доступа: <http://belog.org/reestr-perevodchikov-belorusskogo-zhestovogo-yazyka/reestr-perevodchikov-bzhya>. – Дата доступа: 10.04.2023.
4. Kamal, S. M. Technical Approaches to Chinese Sign Language Processing: A Review / Kamal, S. M. [et al.] // IEEE Access. – 2019. – Vol. 7. – P. 96926–96935.
5. Majeed, S. H. Adaptive Entropy Index Histogram Equalization for Poor Contrast Images / Majeed, S. H., Isa, N. A. M. // IEEE Access. – 2021. – Vol. 9. – P. 2169–3536.

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

6. Xu, W. *A novel star image thresholding method for effective segmentation and centroid statistics* / Xu, W. [et al.] // *Optik*. – 2013. – Vol. 124. – Iss. 20. – P. 4673–4677.
7. Patel, J. M. *A review on feature extraction techniques in Content Based Image Retrieval* / Patel, J. M., Gamit, N. C. // *Proc. IEEE Intern. Conf. on Wireless Communications, Signal Processing and Networking*. – Chennai, India, 2016. – P. 2259–2263.
8. Gopika, P. *Single-layer convolution neural network for cardiac disease classification using electrocardiogram signals* / Gopika, P. [et al.] // *Deep Learning for Data Analytics* / Gopika, P. [et al.]. – Cambridge : Academic Press, 2020. – Chap. 2. – P. 21–35.

МОБИЛЬНЫЙ ДЕЦЕНТРАЛИЗОВАННЫЙ НЕКАСТОДИАЛЬНЫЙ КРИПТОВАЛЮТНЫЙ КОШЕЛЕК НА ПЛАТФОРМЕ IOS

Козко Р.С.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Фадеева Е.Е. – ассистент

Данная статья посвящена мобильным криптовалютным кошелькам как удобному, быстрому и надежному инструменту для хранения и использования криптовалют на мобильных устройствах. В ней рассмотрены основные принципы работы мобильных криптовалютных кошельков и предоставлен путь реализации децентрализованного некастодиального кошелька на платформе iOS.

В последние годы в связи с развитием технологий блокчейн криптовалюта стала одной из наиболее обсуждаемых и быстро развивающихся технологий в мире. Она представляет собой цифровую валюту, которая может быть использована для относительно быстрых и безопасных транзакций в любой точке мира, обходя при этом традиционные финансовые институты. Криптовалюта уже получила широкое распространение и привлекает внимание не только инвесторов и обычных пользователей, но и крупных корпораций и правительств. В связи с быстрым развитием технологий и ростом интереса к криптовалюте, ее актуальность в наше время неоспорима, а роль криптовалютных кошельков в обеспечении безопасности и удобства хранения цифровых активов становится все более важной.

Принцип работы криптовалютных кошельков и криптовалюты в целом основан на использовании публичных и приватных ключей. Приватный ключ используется для создания публичного ключа и отправки криптовалюты, тогда как публичный ключ используется для получения криптовалюты и является уникальным адресом. Когда пользователь отправляет криптовалюту, он подписывает транзакцию приватным ключом и она отправляется в сеть блокчейн, где она подтверждается валидаторами и записывается в блок.

Для разработки мобильного криптовалютного кошелька необходимо решить следующие задачи:

Выбрать или написать собственную библиотеку для управления криптовалютой. Написание собственной библиотеки занимает огромное количество времени и ресурсов, в то время как существует достаточное количество открытых и популярных решений.

Выбрать способ хранения мнемонической фразы. По способу управления приватными ключами криптовалютные кошельки делятся на кастодиальные и некастодиальные [1]. В зависимости от типа мнемонической фразы может храниться на некоем сервере или непосредственно на устройстве. Так как рассматривается создание некастодиального кошелька на платформе iOS, то мнемоника должна храниться непосредственно на мобильном устройстве в защищенном хранилище Keuchain, доступ к которому осуществляется через соответствующий интерфейс;

Обеспечить доступ к блокчейну и как следствие отправку подписанной транзакции, синхронизацию балансов и других данных. Доступ к блокчейну осуществляется через специализированные узлы. Доступ к ним может быть как через серверную, так и напрямую из мобильного приложения в основном по протоколу RPC [2]. Так как рассматривается создание децентрализованного кошелька, то необходимо найти и реализовать взаимодействие с узлами для всех поддерживаемых блокчейнов непосредственно в мобильном приложении;

Обеспечить подсчет наиболее удачных комиссий. Необходимо реализовать алгоритмы подсчета комиссий на основе данных с узлов и из других открытых API для каждого из поддерживаемых блокчейнов;

Синхронизировать список транзакций. Для каждого из поддерживаемых блокчейнов реализовать синхронизацию транзакций пользователя из эксплореров;

Предоставить информацию о текущих курсах криптовалют. Необходимо периодически синхронизировать данные о курсах;

Выбрать базу данных для хранения и удобного использования синхронизированных данных;
Создать наиболее удобный и минимизирующий количество возможных ошибок пользовательский интерфейс.

Таким образом, можно отметить, что мобильные криптовалютные кошельки являются неотъемлемой частью криптовалютной экосистемы, а их реализация требует учета многих аспектов.

Список использованных источников:

1. Custodial vs. Non-Custodial Wallets [Электронный ресурс]. – Режим доступа: <https://academy.binance.com/en/articles/custodial-vs-non-custodial-wallets-what-s-the-difference> – Дата доступа: 06.04.2023.

2. *Remote Procedure Call* [Электронный ресурс]. – Режим доступа:
<https://www.techtarget.com/searcharchitecture/definition/Remote-Procedure-Call-RPC> – Дата доступа: 06.04.2023.

ВЕБ-ПРИЛОЖЕНИЕ ОНЛАЙН-КИНОТЕАТР С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА JAVA И ФРЕЙМВОРКА SPRING BOOT

Колодко Н.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Болтак С.В. – ассистент

В данной работе описывается разработка программного средства для просмотра фильмов онлайн с использованием протокола HLS, а также принцип работы этого протокола.

На сегодняшний день просмотр фильмов и сериалов является одним из самых популярных хобби среди подростков Беларуси [1]. Всё из-за того, что фильмы и сериалы позволяют «переключиться на другую волну» и отдохнуть от обилия мыслей, которые беспокоят на протяжении всего дня. Талантливый фильм или сериал может перевернуть взгляд на мир или на отдельные его аспекты.

Для потоковой передачи фильмов, как и остального видео-контента, часто используется протокол HLS (HTTP Live Streaming) – протокол потоковой передачи данных с адаптивным битрейтом. Он использует файлы с расширением M3U8 для описания медиа потоков и протокол HTTP для связи между сервером и клиентом.

В протоколе HLS существует два типа файлов M3U8:

1. Медиа плейлист: содержит URL-адреса файлов, необходимых для потоковой передачи, то есть фрагменты исходного видео для воспроизведения. Фрагментами являются файлы с расширением TS, которые содержат исходный видео-контент, разбитый на десятисекундные отрывки. Последняя версия HLS позволяет использовать фрагментированные .MP4 файлы. Каждый фрагмент также может содержать в себе субтитры и озвучки, если видео-контент их поддерживает.

2. Основной плейлист: содержит URL-адреса медиа плейлистов, которые, в свою очередь, содержат варианты одного и того же видео, подготовленные для разной пропускной способности [2].

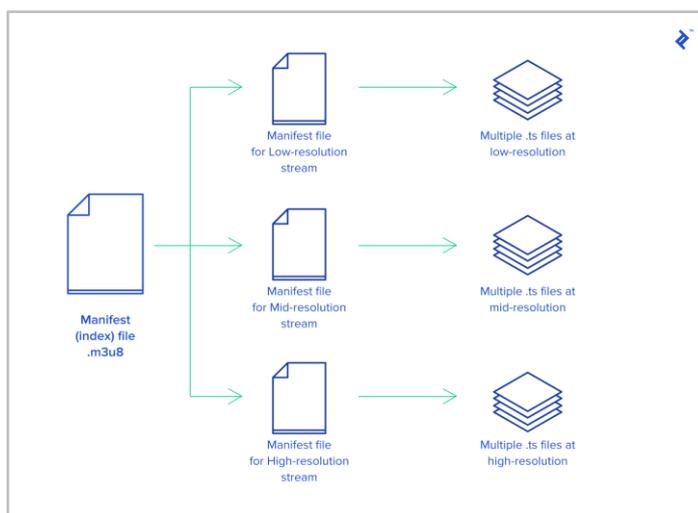


Рисунок 1 – Структура ОСНОВНОГО ПЛЕЙЛИСТА

В разработанном веб-приложении видеоплеер начинает воспроизведение с первого варианта потока. Порядок остальных вариантов не важен.

Если первый файл .TS загружается слишком долго, вызывая ожидание следующего фрагмента, видеоплеер переключается на поток с меньшим битрейтом. И если новый поток загружается достаточно быстро, значит, он может переключиться на более качественный вариант, но только если это имеет смысл для разрешения дисплея. Если первый поток в списке плейлиста .M3U8 не самый лучший, клиенту потребуется один или два цикла, пока он не остановится на подходящем варианте.

Список использованных источников:

1. Какие хобби популярны у белорусских подростков [Электронный ресурс] – Режим доступа: <https://adukar.com/by/news/abiturientu/kakie-hobbi-populyarny-u-belorusskih-podrostkov>. – Дата доступа: 06.04.2023.
2. HTTP Live Streaming [Электронный ресурс] – Режим доступа: <https://www.toptal.com/apple/introduction-to-http-live-streaming-hls>. – Дата доступа: 06.04.202

ПРОГРАММНОЕ СРЕДСТВО УЧЁТА ПЕРСОНАЛА ОРГАНИЗАЦИИ НА ОСНОВЕ ТЕХНОЛОГИИ WOOCEL И POSTGRESQL

Кондрацкий А.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сурков Д.А. – ст. преподаватель

В работе рассматривается проект для управления рабочим персоналом организации. Рассматриваются принципы построения аналогичных продуктов и их особенностей.

В настоящее время информационные технологии занимают огромное количество отраслей жизни. Иногда они используются просто для развлечения, но зачастую они находят применение в науке и даже в бизнесе.

Любая компания располагает рабочим персоналом, для которого определены различные бонусы и поощрения, установленное количество дней отпуска и многое другое. На рынке существуют программные решения, направленные на крупный и средний бизнес. Такие решения направлены на автоматизацию многих отделов производства. Это может быть как и управление финансовой деятельностью, так и управление закупками, ведение бухгалтерской отчётности и многое другое. Программы такого масштаба громоздки, они требуют установки на рабочие компьютеры организации.

Но есть еще один немаловажный аспект, который заключается в том, что корпоративные решения направлены, например, на работу самого отдела кадров или бухгалтера, нежели на сотрудника. Но, рассматривая использование программного средства для малого бизнеса, важно ориентировать сотрудника на работу с системой.

Учитывая все перечисленные аспекты, становится понятно, что это должно быть легковесное, веб-ориентированное приложение. Приложение должно быть простым и понятным. Этого можно добиться, используя табличное представление информации. Для визуализации данных удобно использовать отдельные карточки, которые вместе будут составлять таблицу. Каждая карточка должна иметь одинаковую структуру, но разные данные.

Пользовательский интерфейс программы должен поддерживать работу с этими таблицами, позволять изменять, добавлять, обновлять информацию. Для возможности использования приложения разными предприятиями необходима гибкость. Структура каждой карточки должна настраиваться отдельно, отвечая всем требованиям и запросам компании. Пример такой карточки можно видеть на рисунке 1.

	Vitali Griffin Senior Software Engineer { Tags }	Грифин	Виталий	Игоревич	Vitali Hrifin 1966493245
		{ Projects }	+375291242341	vitalii.gri@gmail.com	
			г. Минск, 220062, ул. Мясникова д. 16 кв. 105		
			г. Минск, 220062, ул. Мясникова д. 16 кв. 105		
	Herman Parker Senior Software Engineer { Tags }	Паркер	Герман	Дмитриевич	Herman Parker 2144806606
		{ Projects }	+375291242341	parker.h.d@gmail.com	
			г. Минск, 220015, ул. Одоевского д.1, кв. 15		
			г. Минск, 220015, ул. Одоевского д.1, кв. 15		
	Daniil Robinson Software Engineer { Tags }	Розбинсон	Даниил	Григорьевич	Daniil Robinsan 2071906426
		{ Projects }	+375441242341	donila@gmail.com	
			Минский р-н д. Лесковка ул. Центральная 1 к. 17		
			Минский р-н д. Лесковка ул. Центральная 1 к. 17		

Рисунок 1 – Таблица данных о работниках компании

Для создания программного средства были использованы существующие инструменты, такие как Woocel и PostgreSQL. Woocel – разработка компании Nezaboodka. Это приложение для работы с большими данными в виде таблиц. Конфигурация этих таблиц ведётся с помощью спецификаций, которые задают разметку для таблиц и карточек [1]. PostgreSQL – бесплатная свободная объектно-реляционная система управления базами данных [2].

В результате работы было создано программное средство, позволяющие визуализировать огромное количество данных. Оно имеет возможность гибко конфигурировать отдельные карточки с данными. Также приложение отличается веб-ориентированностью, что позволяет сотрудникам, используя мобильный телефон или компьютер, получить доступ к приложению в любой удобный момент времени.

Список использованных источников:

1. Woocel spreadsheets [Электронный ресурс]. – Режим доступа: <https://woocel.com/>. – Дата доступа: 08.04.2023.
2. Obe, Regina O. PostgreSQL: Up and Running / Regina O. Obe, Leo S. Hsu. – Sebastopol: O'Reilly Media, 2018. – 322 p.

ПРОГРАММНОЕ СРЕДСТВО РАСПОЗНАВАНИЯ ЯЗЫКА ЖЕСТОВ

Кравченко П.Д.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Красковский П.Н. – ст.преп.

Предложена структура программного средства распознавания языка жестов для классификации American Sign Language (ASL) жестов. Описана модель, обученная на размеченных точках, полученных с помощью MediaPipe Holistic Solution.

Так как обучение модели лучше всего воспроизводить на одном языке жестов, то в качестве используемого был выбран American Sign Language (ASL) по нескольким причинам. Во-первых, он является одним из самых популярных языков жестов на данный момент времени. Во-вторых, ASL может послужить как один из тех языков, что применяются на международном уровне, чтобы носители из разных уголков планеты могли понимать друг друга. Для обучения был использован датасет, который предоставляет 250 различных жестов от 21 человека разной расы, пола, возраста, интерьера съемки [1]. Их жесты могут содержать в себе переходы с прошлых жестов или иметь какие-нибудь недочеты. Некоторые носители использовали левую руку, другие правую. Кто-то переключался с одной на другую.

Обработка входных данных будет осуществляться с помощью MediaPipe Holistic Solution [2], идея которого заключается в объединении различных моделей для рук, позы, лица человека. Использовать эти модели отдельно является не очень хорошей идеей, так как распознавание позы принимает на вход видеокادر с фиксированным разрешением (256x256), а если обрезать руки и лицо для двух других моделей, то разрешение получается слишком низким для точной артикуляции. Holistic Solution справляется с этой проблемой. Также нет необходимости использовать все 468 точек лица, поэтому остаются только те, что отвечают за область губ. Это связано с тем, что при жестикуляции человек может также проговаривать слова.

Основой программного средства является модель, которая на вход принимает данные, а на выходе возвращает значение жеста. Для решения этой задачи использованы трансформеры [3], которые были представлены в 2017 году специалистами из Google Brain с целью решения задач обработки естественного языка. Одно из основных отличий от существующих методов обработки данных заключается в том, что входная последовательность может передаваться параллельно, чтобы можно было эффективно использовать графический процессор, а также увеличивать скорость обучения.

Основными компонентами трансформеров являются энкодер и декодер. Энкодер берет на вход данные и проецирует их на пространство большей размерности (N-мерный вектор). Этот абстрактный вектор подается в декодер, который превращает его в выходную последовательность, и в данном случае обозначает значение жеста. Другие инновации, лежащие в основе трансформеров, сводятся к трем основным концепциям: позиционные энкодеры, внимание и самовнимание.

Позиционные энкодеры позволяют распараллелить процесс. Для этого используются позиционные кодировки, что помогают перенести бремя понимания порядка со структуры нейронной сети на сами данные. Сначала, прежде чем трансформеры обучатся на какой-либо информации, они не знают, как интерпретировать эти позиционные кодировки. Но по мере того, как модель видит все больше и больше примеров и их кодировок, она учится эффективно их использовать.

Механизм внимания одновременно просматривает несколько частей входной последовательности и решает, какие из них важны, приписывая им разные веса. Декодер же помимо вектора принимает и эти веса, предоставленные механизмом, что делает его работу намного проще, потому что теперь есть понимание того, что больше всего влияет на выбор выходного значения.

Последняя часть трансформеров — это поворот внимания, называемый самовниманием. Если механизм внимания определяет, какие значения вектора являются самыми важными, то самовнимание модифицирует каждое значение вектора подмешивая к нему другие близкие значения из контекста с некоторыми весами.

Список использованных источников:

1. Kaggle ASL-signs [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.kaggle.com/competitions/asl-signs/data>.
2. MediaPipe Holistic Solution [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://github.com/google/mediapipe/blob/master/docs/solutions/holistic.md>.
3. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Polosukhin I., Kaiser Ł. Attention is All you Need (англ.) // Advances in Neural Information Processing Systems 30 / I. Guyon, U. v. Luxburg, S. Bengio, H. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett — 2017. — arXiv:1706.03762.

ПРОГРАММНОЕ СРЕДСТВО ОТСЛЕЖИВАНИЯ ЦЕЛОСТНОСТИ КОНТЕЙНЕРОВ

Логвинов Д.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Леванцевич В.А. –старший преподаватель

Проанализированы методы отслеживания целостности файлов. Рассмотрены алгоритмы вычисления хэш-суммы, методы хранения состояния файлов, виды нотификаций об изменении отслеживаемых файлов, варианты внедрения программного средства внутрь пода с отслеживаемым контейнером. Определен оптимальный способ внедрения программного средства.

Контейнеризация – это способ упаковки программного средства и всех необходимых для его работы зависимостей в один образ, который запускается в изолированной среде в виде отдельного контейнера. Данный метод позволяет удобно доставлять и развёртывать программные средства. Проверка целостности контейнеров, имеет широкий спектр применений начиная от правильности доставки данных при развёртывании, заканчивая защитой файлов конфигурации от злоумышленников.

В ходе исследования проблемы целостности контейнеров в качестве способа отслеживания их состояния был выбран метод, основанный на сверке хэш-суммы файлов через определённые промежутки времени, настраиваемые пользователем в файле конфигурации. Метод сверки хэш-суммы позволяет достичь максимальной скорости проверки целостности файлов внутри отслеживаемых контейнеров [1].

В качестве предустановленных криптографических алгоритмов вычисления хэш-суммы выступают SHA-256, SHA-512 и MD5 [2]. Также был реализован унифицированный интерфейс для подключения иных алгоритмов хэширования, опробовать данный интерфейс можно на основе библиотеки Bce2, реализовывающий алгоритм вычисления хэш-суммы белорусского стандарта.

Для интеграции программного средства внутрь пода с отслеживаемым контейнером был выбран способ инъекции внутрь пода типа sidecar-контейнер, посредством получения webhook нотификации при развёртывании отслеживаемого контейнера в кластер. Данный способ позволяет упростить конфигурацию программного средства посредством добавления метки внутрь аннотации развёртывания. На рисунке 1 представлена схема интеграции программного средства.

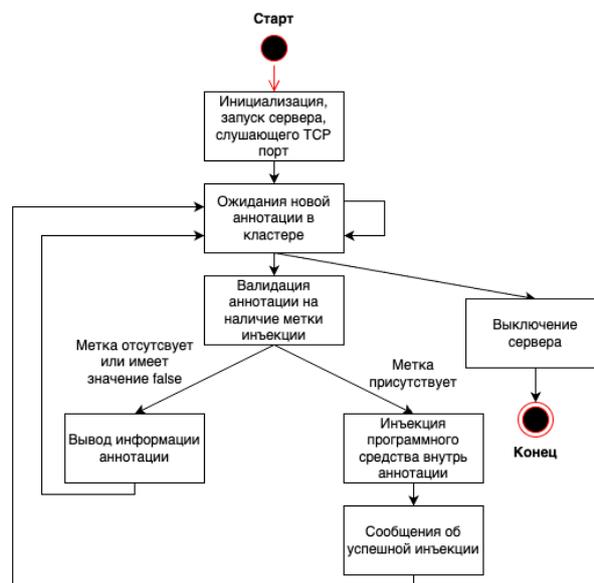


Рисунок 1. – Схема интеграции программного средства

Выводы: сверка хэш-суммы файлов является наиболее удобным и эффективным способом отслеживания целостности файлов и позволяет гибко настраивать программное средство для его использования со стандартами любых стран.

Список использованных источников:

1. Маркелов А., Введение в технологии контейнеров и Kubernetes. - М.: ДМК.Пресс, 2019. – 194 с.
2. Чмора А., Современная прикладная криптография. - М.: Гелиос АРВ, 2001. - 218

УДК 004.021:004.4

МЕТОДЫ И АЛГОРИТМЫ КОНТРОЛЯ СООТВЕТСТВИЯ ВЕРСИЙ И ЦЕЛОСТНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМ ПЕРЕМЕЩЕНИЙ

Марко А.Ф., аспирант

*Белорусский государственный университет информатики и радиоэлектроник
г. Минск, Республика Беларусь*

Карпович С.Е. – доктор техн.наук, профессор

Аннотация. Представлены методы и алгоритмы контроля за соответствием версий и целостности программного обеспечения для управления системами перемещений реального времени с использованием стандарта передачи данных EtherCAT.

Ключевые слова. Программное обеспечение, соответствие версий, контроль целостности, системы перемещений, стандарт EtherCAT.

Объединение узлов точной механики с электронными, электрическими и компьютерными компонентами позволило осуществлять проектирование и производство качественно новых модулей, систем и машин с их интеллектуальным управлением. С развитием электрических приводов и возможностей их применения в индустриально-производственных и транспортных системах, стала очевидна необходимость полной интеграции составляющих элементов электропривода: механики, электрических машин, силовой электроники, микропроцессорной техники и программного обеспечения для наиболее полного использования возможностей современного электропривода, и построения на его основе мехатронных систем перемещения [1].

Проведённый анализ современных программно-аппаратных средств показал, что наиболее эффективной технологией для реализации управления системами многокоординатных перемещений в режиме реального времени является технология EtherCAT, внедрение которой требует разработки дополнительного программного обеспечения. Разработка такого программного обеспечения выполняется с применением специальных инструментов, которые повышают эффективность разработки за счёт снижения трудоёмкости выполняемых операций. К таким инструментам относятся различные среды разработки программного обеспечения и системы контроля версий. В данных инструментах существует проблема отсутствия универсального решения для версионирования dll-библиотек и исполняемых exe-файлов программного обеспечения с целью установления связи между данными файлами и их исходным кодом. Также недостаточно проработан вопрос контроля целостности программного обеспечения, особенно объектов базы данных [1].

В рамках настоящей работы рассматривается программное обеспечение для контроля целостности и соответствия версий при управлении системами многокоординатных перемещений в режиме реального времени.

Для управления в реальном времени всё большее распространение получает технология EtherCAT. Типовая схема пересылки данных в EtherCAT-сети представлена на рисунке 1.

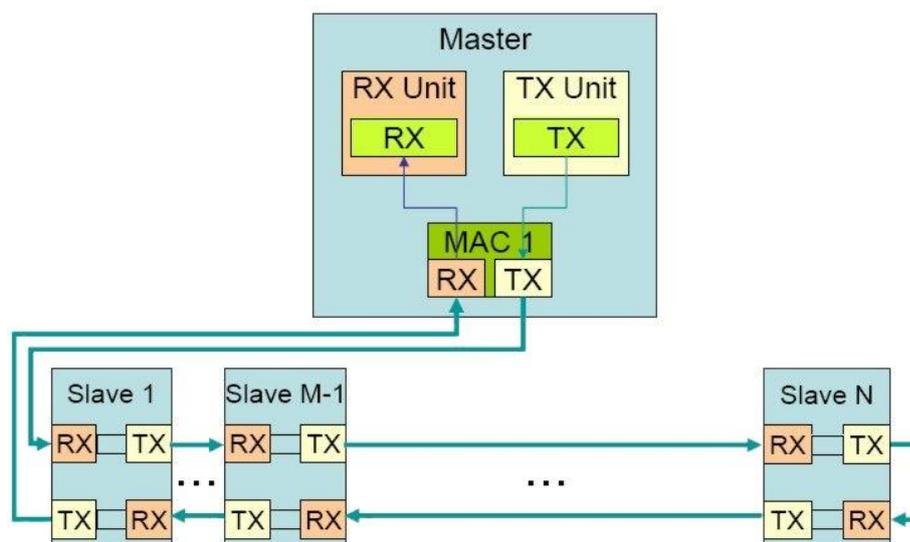


Рисунок 1 – Типовая схема пересылки данных в EtherCAT-сети

Из всех устройств, подключённых к шине EtherCAT, только мастер может быть инициатором телеграмм. Все остальные устройства модифицируют проходящую через них телеграмму, читая и записывая в неё данные технологического процесса. Аппаратная задержка на прохождение телеграммы через одно slave-устройство составляет всего несколько наносекунд [1].

Рассматриваемая в работе EtherCAT-сеть (рисунок 2) содержит один управляющий компьютер master и шесть локальных систем управления, каждая из которых работает в режиме slave и обеспечивает реализацию прецизионных перемещений соответствующего планарного позиционера по двум координатам.

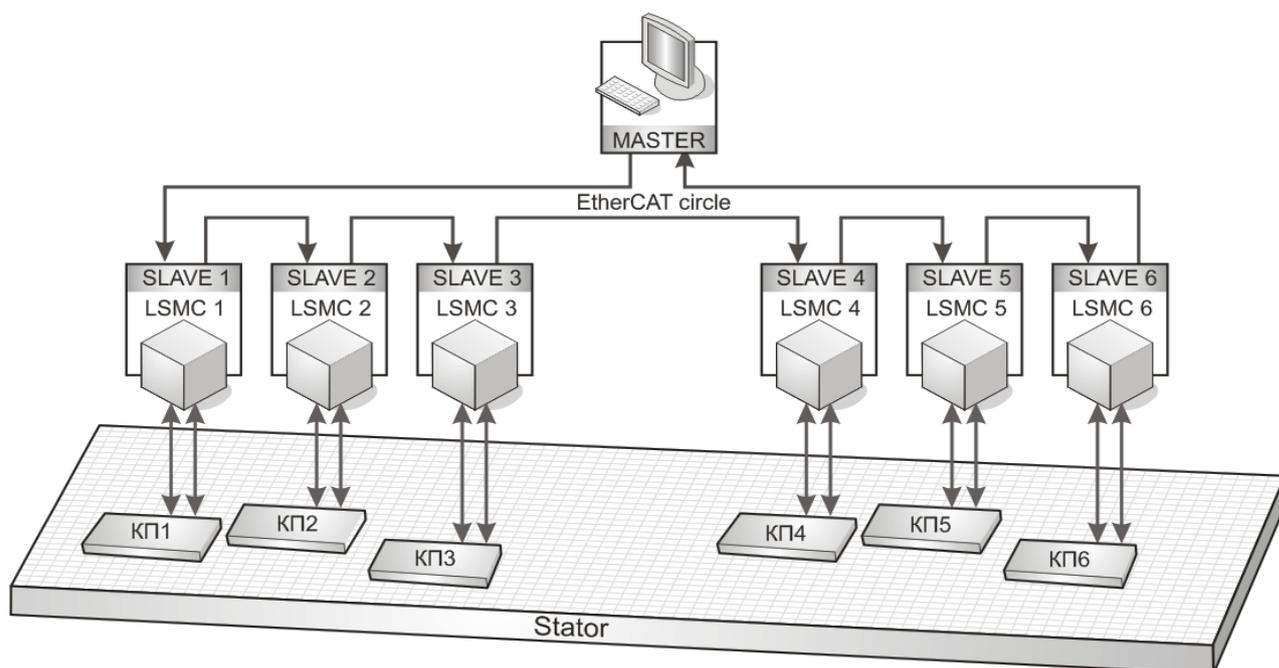


Рисунок 2 – Схема EtherCAT-сети для управления шестью позиционерами

Таким образом, EtherCAT-технология предоставляет разработчикам систем управления технологическими процессами и сложным оборудованием полностью интегрированное решение, обеспечивающее стандартную и надёжную сеть обмена управляющей информацией. При этом количество задействованных полевых шин и интерфейсов уменьшается, обеспечивая тем самым унификацию всех процессов управления, гибкость структуры при практически неограниченном количестве устройств и малое время реакции на события, а также обеспечивается возможность переконфигурирования системы управления без необходимости её полного отключения [1].

В связи с намечившимся внедрением технологии EtherCAT в прецизионное технологическое оборудование актуальной и важной является разработка специальных инструментов, позволяющих разрабатывать программное обеспечение системы управления в множестве версий и тем самым с

постоянным изменением кода. Поэтому актуальной и важной является задача автоматизации контроля за соответствием версий компонентов такого программного обеспечения в процессе его разработки и контроля целостности в процессе эксплуатации.

Контроль за соответствием версий позволяет решить задачу обновления версий сборок с расширениями dll и exe при изменении их исходного кода, который компилируется в данные сборки при помощи среды Visual Studio (VS). Алгоритмы обновления версий реализованы в виде расширения для среды VS, которая в свою очередь может взаимодействовать как с централизованной системой управления версиями Team Foundation Server (TFS), так и с децентрализованной системой Git [2]. Разработанный с применением технологии WPF (Windows Presentation Foundation) интерфейс пользователя расширения интегрирован в среду VS, что позволяет осуществлять версионирование и разработку программного обеспечения в одном окружении. Данное расширение определяет какие компоненты версионизируемого программного обеспечения подверглись изменениям по отношению к последней версии в системе TFS или Git, формирует новую версию, присваивает данную версию компонентам и сохраняет изменения в систему TFS или Git [2, 3]. В случае программного обеспечения, разрабатываемого на языке C#, базовыми компонентами являются проекты [2].

Версия каждого проекта хранится в специальном файле AssemblyInfo и изменяется, данная версия отображается в свойствах исполняемых exe-файлов и dll-библиотек, которые соответствуют проектам и формируются в результате их сборки. Для получения или изменения версии в расширении используются регулярные выражения. Новая версия для изменённого проекта формируется следующим образом: старшие разряды берутся из версии основного проекта родительского решения, которое определяется с помощью структурного анализа файла проекта, в качестве младшего разряда версии выступает инкрементированный номер последнего сохранения в системе TFS или Git. Старшие разряды обновляются только при сборке релиза программного обеспечения и только в случае внесения существенных изменений.

Алгоритмы формирования и сравнения контрольных сумм в процессе эксплуатации встроены непосредственно в программное обеспечение системы управления и предназначены для определения незапланированных изменений [3, 4]. Программное обеспечение для системы управления состоит из множества различных объектов, таких как исполняемые файлы, файлы данных и объекты баз данных, формирование контрольных сумм выполняется для каждого типа по-разному. Также принимается во внимание, что некоторые объекты, такие как таблица пользователей или регистрационные файлы изменяются в процессе эксплуатации, следовательно, контрольные суммы для них не формируются. На рисунке 3 приведена последовательность формирования контрольных сумм для выделяемых типов объектов ПО.

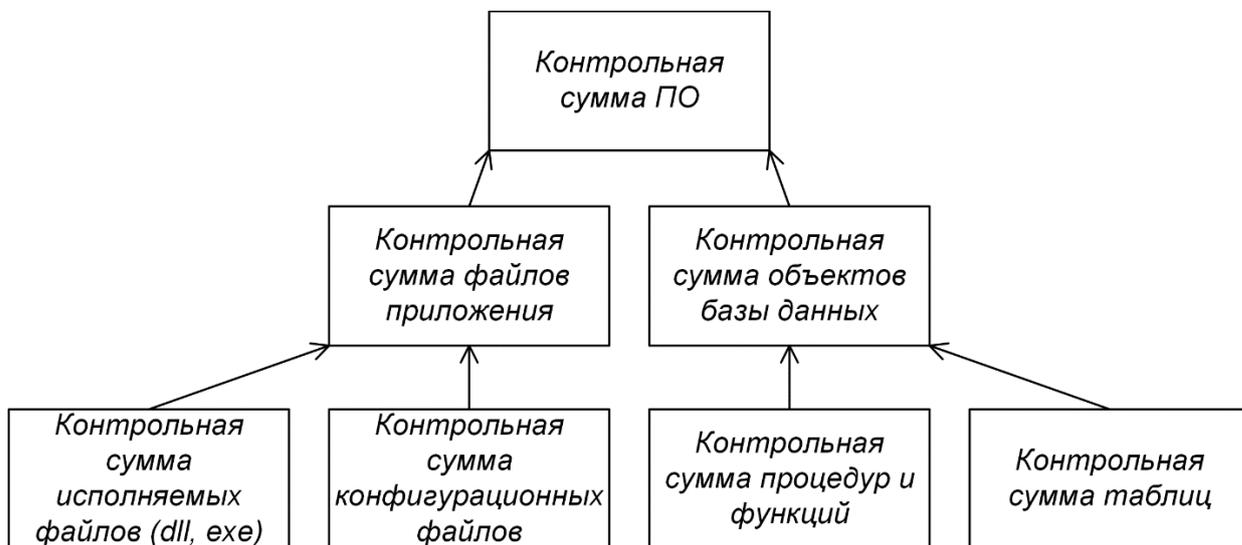


Рисунок 3 – Последовательность формирования контрольных сумм для выделяемых типов объектов ПО

Иерархичность процесса формирования и хранения контрольных сумм заключается в удобстве представления информации о состоянии каждой подгруппы объектов и всей системы в целом, также такой подход позволяет сократить времени, необходимое для нахождения модифицированных объектов программного обеспечения.

Для формирования контрольных сумм объектов используется современный алгоритм SHA-2, который создаёт контрольные суммы от 224 до 512 бит, что обеспечивает высокую степень

надёжности и защиты от подделки или изменения файла. Кроме того, SHA-2 является стандартом безопасности для многих приложений и операционных систем [5].

Таким образом были разработаны: метод и алгоритмы контроля за соответствием версий компонентов программного обеспечения, заключающиеся в автоматизированном обновлении версий dll-библиотек и исполняемых exe-файлов при внесении изменений в их исходный код, а также метод и алгоритмы контроля целостности программного обеспечения, заключающиеся в формировании эталонных контрольных сумм с использованием алгоритма SHA-2 для объектов программирования и объектов баз данных, сравнении их с текущими контрольными суммами, и позволяющие детектировать любые изменения указанных объектов и тем самым уменьшить вероятность использования программного обеспечения с незапланированными изменениями. Данные методы и алгоритмы оказались весьма востребованными для систем управления реального времени на многокоординатных приводах прямого действия, таких как тестеры печатных плат, сборочное и оптико-механическое оборудование микроэлектроники.

Список использованных источников:

1. Карпович, С.Е. Системы многокоординатных перемещений на механизмах параллельной кинематики: монография / С.Е. Карпович [и др.]; под ред. д-ра техн. наук, проф. С.Е. Карповича. – Минск : Бестпринт, 2017. – 254 с.
2. Шарп, Дж. Microsoft Visual C#. Подробное руководство. 8-е изд. – СПб.: Питер, 2017. – 848 с.
3. Марко, А.Ф. Контроль целостности и соответствия версий программного обеспечения для управления системами перемещений в режиме реального времени / А.Ф. Марко // Информационные системы и технологии = Informatic Systems and Technologies [Электронный ресурс] : материалы междунар. науч. конгресса по информатике. В 3 ч. Ч. 1, Респ. Беларусь, Минск, 27–28 окт. 2022 г. / Белорус. гос. ун-т; редкол.: С.В. Абламейко (гл. ред.) [и др.]. – Минск : БГУ, 2022. – С. 89–94.
4. Марко, А.Ф. Методы соответствия версий и контроля целостности программного обеспечения для систем перемещений в режиме реального времени / А.Ф. Марко // Информационные технологии и системы 2022 (ИТС 2022) = I74 Information Tehnologies and Systems 2022 (ITS 2022) : материалы международной научной конференции, Минск, Беларусь, 23 ноября / Л. Ю. Шилин [и др.]. – Минск : БГУИР, 2022. – С. 63–64.
5. Нильс Фергюсон, Брюс Шнайер. Практическая криптография = Practical Cryptography: Designing and Implementing Secure Cryptographic Systems. – М. : Диалектика, 2004. – 432 с.

UDC 004.021:004.4

METHODS AND ALGORITHMS FOR CONFORMITY OF VERSIONS AND INTEGRITY OF SOFTWARE OF MOVEMENT SYSTEMS

Marko A.F

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Karpovich S.E. – Ph.D., Full Professor

Annotation. Methods and algorithms for monitoring the integrity and compliance of software versions for managing real-time motion systems using EtherCAT data transfer standard are presented.

Keywords. Software, Version Compliance, Integrity Control, Motion Systems, EtherCAT standard.

ОНЛАЙН-СЕРВИС КОНТРОЛЯ ОСНОВНЫХ ПРОЦЕССОВ УПРАВЛЕНИЯ ПЕРСОНАЛОМ (HRM) ИЗ ЕДИНОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ФРЕЙМВОРКА ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP – LARAVEL

Михальков М.Д.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Фадеева Е.Е. – ассистент

В результате работы была спроектирована и разработана база данных, а также веб-приложение для онлайн-сервиса контроля основных процессов управления персоналом (HRM) из единой системы. Отличительной особенностью программного средства стала система мотивации сотрудников. Ее идея состоит в следующем: за какие-то отличительные качества сотрудник премируется с помощью условной единицы, после набора определенного количества таких условных единиц, сотрудник имеет право на прохождение собеседования на повышение заработной платы, должности и так далее; за какие-то проступки сотрудник будет лишаться условных единиц, что в последующем может привести к штрафам и выговорам.

Во всем мире организации накапливают или уже накопили в процессе своей административно-хозяйственной деятельности большие объемы данных, в том числе и в электронном виде. Эти коллекции данных хранят в себе большие потенциальные возможности по извлечению новой аналитической информации, на основе которой можно и необходимо строить стратегию организации, выявлять тенденции развития рынка, находить новые решения, обуславливающие успешное развитие в условиях конкурентной борьбы. Для некоторых организаций такой анализ является неотъемлемой частью их повседневной деятельности, другие только начинают активно приступать к нему [1].

Идея проекта состояла в реализации онлайн-сервиса контроля основных процессов управления персоналом (HRM) из единой системы.

Human Resource Management (HRM) или Human Resources Information System (HRIS) – это CRM-система, которая помогает привлекать и контролировать собственный персонал. Это комплексная система автоматизации найма, отбора, привлечения персонала, обеспечения ориентации, обучения и развития, оценки их работы, определения вознаграждения и предоставления льгот, мотивации, поддержания надлежащих отношений с каждым членом команды и обеспечения безопасности работы за счет применения программ для HR и HR-аналитики.

Выбор программы для HR-менеджеров и правильной системы для управления персоналом – важный процесс для компании. В любой отрасли существует немного успешных компаний, у которых нет какой-либо программы для автоматизации HR на данном этапе. Уже недостаточно иметь списки сотрудников и их зарплат, а также excel-таблицы с отпусками и больничными, требуется более сложная система, в которой можно видеть информацию по каждому кандидату, сотруднику, отделу или подразделению, оценивать вовлеченность, собирать обратную связь от сотрудников для анализа удовлетворенности, вести учет по рабочим часам, отпускам и больничных в рамках одной системы. Современные технологии изменили ситуацию в управлении рутинными HR процессами, предоставляя удобное программное обеспечение для HR менеджеров и сотрудников компаний. HRM система помогла эффективно разрушить бюрократию и «сгладить» многие организационные моменты, чтобы улучшить HR бренд [2].

В процессе работы был произведен патентный поиск аналогов разрабатываемого программного средства с помощью поисковой системы Google Patents [3]. В результате было выявлено три основных сайта-аналога: Zoho People, Asana, Microsoft Azure. Все они имели свои преимущества и недостатки. Их анализ позволил выделить сильные стороны, применить их в разрабатываемом программном средстве, устранить недостатки сайтов-аналогов, а также внедрить систему мотивации сотрудников.

На основе анализа результата патентного поиска были сформированы следующие основные функциональные требования, предъявляемые к программному средству:

- вывод информации о пользователях;
- система авторизации пользователей;
- отображение организационной структуры;
- оценка эффективности работы сотрудников;
- расчет расходов на отпуска сотрудников;
- управление отпуском сотрудников;
- подача запросов на отпуск;
- вывод календаря компании;

- зарплатный инструмент;
- вывод упорядоченного списка проектов;
- вывод упорядоченного списка задач;
- автоматизированная система уведомлений;
- автоматизированная система штрафов и поощрений;
- загрузка и хранение файлов организации;
- система целей и задач сотрудников;
- экспорт контрольных списков основных бизнес-процессов организации.

Разработанное программное средство обладает порталом самообслуживания, где сотрудники и сами менеджеры могут участвовать в обработке данных. Это увеличивает точность и уменьшает административные промахи.

Отличительной особенностью стала система мотивации сотрудников. Ее идея состоит в следующем: за какие-то отличительные качества сотрудник премируется с помощью условной единицы, после набора определенного количества таких условных единиц, сотрудник имеет право на прохождение собеседования на повышение заработной платы, должности и так далее; за какие-то проступки сотрудник будет лишаться условных единиц, что в последующем может привести к штрафам и выговорам.

Эти функциональные требования были полностью реализованы в ходе разработки программного средства.

Необходимость разработки данного программного средства обусловлена тем, что с каждым годом все большее количество компаний сталкивается с трудностями, связанными с контролем основных процессов управления персоналом.

Данные трудности связаны не только с огромным объемом данных, подлежащим обработке и анализу, но и с ошибками, вызванными из-за человеческого фактора.

Разработанное программное средство предназначено для автоматизации основных процессов управления персоналом предприятия и предоставления аналитики по результатам деятельности предприятия. Тем самым онлайн-сервис позволяет устранить трудности, описанные выше.

Общая тенденция в сторону автоматизации основных процессов управления персоналом позволяет прогнозировать более высокий спрос на подобные решения в ближайшие годы и коммерческий успех решения.

Веб-приложение разрабатывалось на базе фреймворка языка программирования PHP – Laravel для серверной части, а также языка программирования JavaScript и библиотеки этого языка jQuery на стороне клиента. В роли СУБД выступила MySQL.

Интерфейс главной страницы сотрудника представлен на рисунке 1:

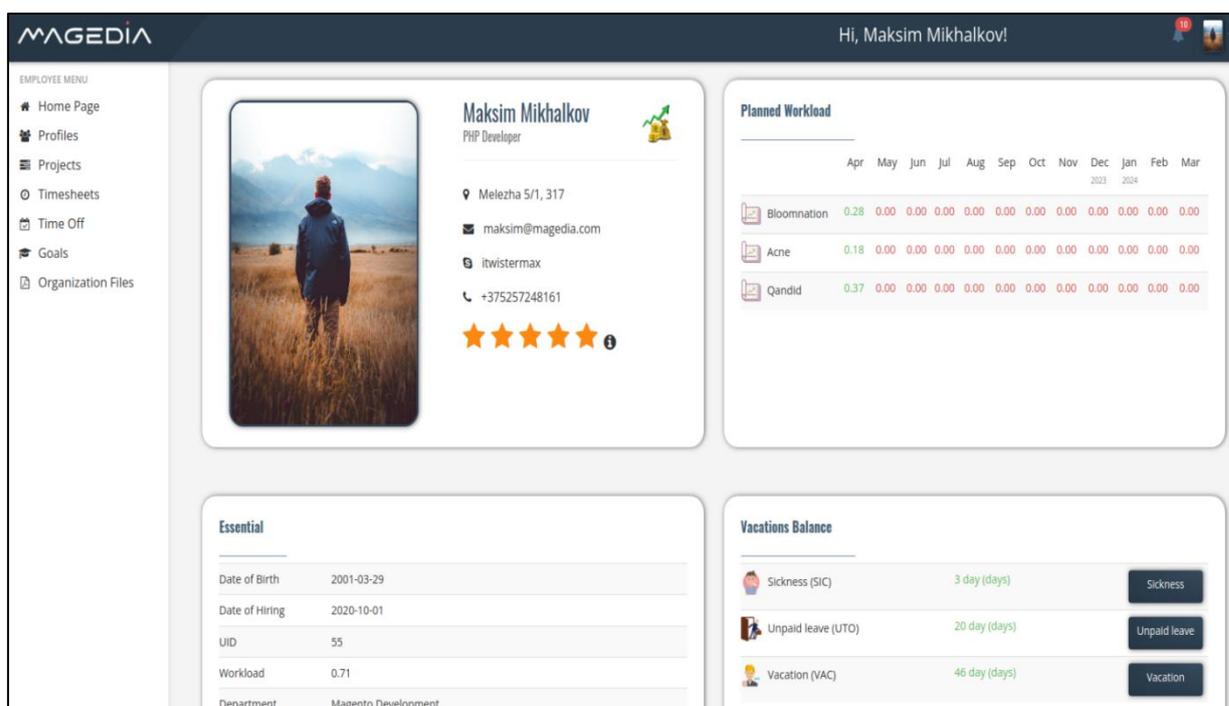


Рисунок 1 – Интерфейс главной страницы сотрудника

Список использованных источников:

1. Технологии хранения данных [Электронный ресурс] – Режим доступа: http://mf.grsu.by/UchProc/lib/olap/bd_wh/doc18.htm/ – Дата доступа: 07.04.2023.
2. Что такое HRM, и как оптимизировать HR процессы? [Электронный ресурс] – Режим доступа: <https://peopleforce.io/ru/blog/cto-takoe-hrm-i-kak-legko-avtomatizirovat-hr-protsessy/> – Дата доступа: 07.04.2023.
3. Google Patents [Электронный ресурс] – Режим доступа: <https://patents.google.com/> – Дата доступа: 07.04.2023.

ПРОГРАММНОЕ СРЕДСТВО СБОРА И ВИЗУАЛИЗАЦИИ ТЕХНОЛОГИЧЕСКИХ ПАРАМЕТРОВ

Насанович Г.Б.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Леванцевич В.А. –старший преподаватель

Рассмотрены подходы к построению систем сбора параметров на основе программируемых контроллеров. Проанализированы протоколы передачи данных. Разработано программное средство сбора, обработки и визуализации технологических параметров.

Автоматизация современных производств требует оперативного контроля разнообразных технологических параметров. Для сбора и первичной обработки данных используются контроллеры с программируемой логикой (PLC), среди которых можно выделить контроллеры компании Omron corporation. Эта компания выпускает большой перечень контроллеров различного назначения и характеристик.

Omron PLC содержит ряд аппаратных особенностей, которые накладывают определенные ограничения на разработку программ для них. Контроллеры имеют ограниченный объем оперативной памяти, фиксированное время выполнения команд и особый цикл их выполнения. Программы для Omron PLC состоят из подпрограмм, которые называются задачами. Задачи выполняются циклично, входные и выходные значения обновляются только после выполнения полного цикла задач (рисунок 1)[1].

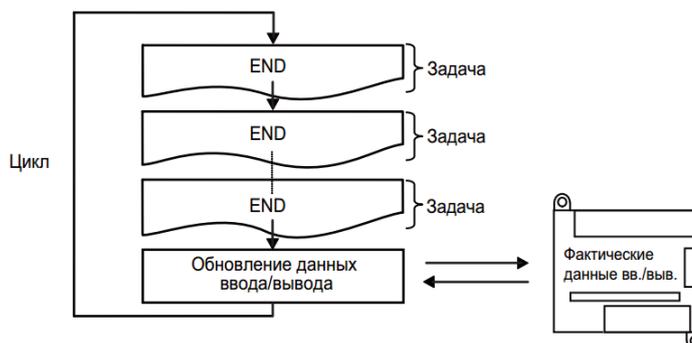


Рисунок 1. – Циклическое выполнение задач

Для подключения контроллеров к сети передачи данных компания Omron разработала специальный протокол связи, который называется FINS (Factory Interface Network System). Обработчик FINS-команд позволяет вставлять задачи работы с памятью в общий цикл обработки задач после обновления входных и выходных значений.

FINS-команды могут быть переданы как через интерфейс Ethernet, так и через COM порт. Обработчик FINS-команд может принимать TCP и UDP запросы. Для работы с TCP необходимо отправлять специальный заголовок, в котором будет указано количество передаваемых байт. FINS-команда состоит из заголовка команды, кода команды и текста команды. Заголовок команды состоит из 10 байт, которые необходимо заполнить, для точной адресации контроллера. Код команды состоит из 2 байт. Текст команды может иметь размер от 0 байт до 512 байт [2].

Для разработки программного средства контроля и визуализации технологических параметров на основе данных, полученных с контроллеров Omron, выбрана клиент-серверная архитектура. Для клиентской части была выбрана библиотека React, а для серверной части – ASP .Net 6.

Выводы: программное средство позволяет оперативно отслеживать технологические параметры по которым можно принимать решение о состоянии контролируемого оборудования или техпроцесса.

Список использованных источников:

1. FINS commands reference manual [Электронный ресурс] - Режим доступа: https://www.myomron.com/downloads/1.Manuals/Networks/W227E12_FINS_Commands_Reference_Manual.pdf - дата доступа 2.04.2023
2. CP1H/CP1L CPU Unit PROGRAMMING MANUAL [Электронный ресурс] - Режим доступа: https://assets.omron.eu/downloads/manual/en/v1/w451_cp1_cpu_unit_programming_manual_en.pdf - дата доступа 2.04.2023

СВЯЗНЫЕ НЕИСПРАВНОСТИ ВЗАИМНОГО ВЛИЯНИЯ

Деменковец Д.В., Петровская В.В.

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Ярмолик В.Н. – профессор, д.т.н.

Рассматриваются модели связанных неисправностей взаимного влияния, состоящих из одиночных неисправностей. Представлен пример визуализации массива ячеек памяти после применения многократного маршевого теста MATS++.

Оперативная память является основным компонентом вычислительных систем. Актуальной задачей является исследование связанных неисправностей запоминающих устройств в силу сложности их обнаружения однократными маршевыми тестами (March tests) [1].

Связанные неисправности взаимного влияния (linked coupling faults – LCF) представляют собой множество одиночных неисправностей взаимного влияния, которые имеют общие ячейки памяти. Общие ячейки памяти могут одновременно участвовать в нескольких одиночных неисправностях и выполнять различные роли. Например, на рис. 1 представлены модели связанных неисправностей взаимного влияния, включающие три ячейки памяти с адресами $i < j < k$, в которых общая ячейка a_j выступает как в роли агрессора, так и в роли жертвы. В некоторых случаях одна и та же ячейка памяти может иметь обе роли по отношению к другой ячейке.



Рисунок 1 – Примеры связанных неисправностей взаимного влияния

В качестве эксперимента было сгенерировано 53 различных типа связанных неисправностей взаимного влияния, включающие в себя три ячейки памяти. Моделируемые связанные неисправности состояли из одиночных инверсных неисправностей (CFin). Для памяти размером 4096 ячеек многократно применялся маршевый тест MATS++ с изменением адресной последовательности. Адресные последовательности были сгенерированы с помощью регистра сдвига с линейной обратной связью. Визуализация итераций теста представлена на рисунке 2 [2]. На рис. 2, а отображен исходный массив ячеек, выполняющих роль жертв в смоделированных неисправностях. На рис. 2, b, c, d – состояния памяти после применения одной, двух и трех итераций теста соответственно.

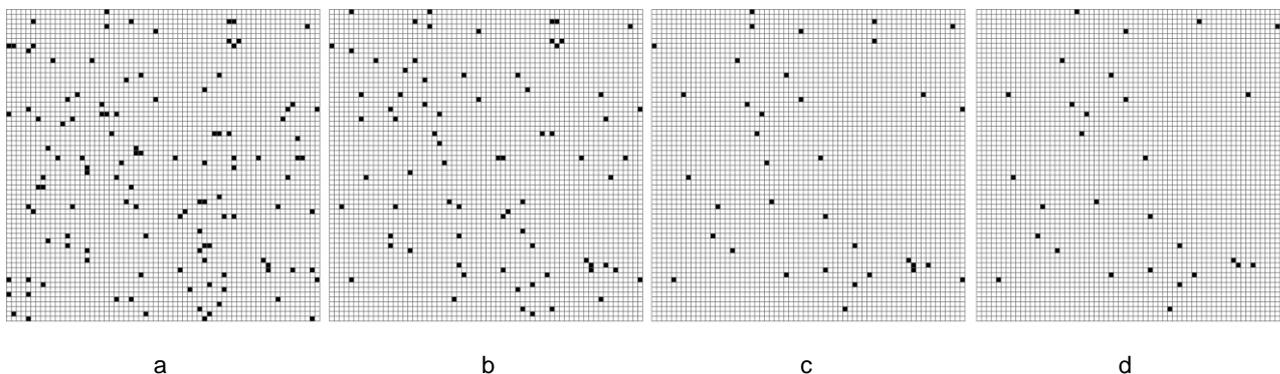


Рисунок 2 – Пример визуализации массива ячеек памяти

Результат эксперимента показывает, что многократное применение теста MATS++ приводит к увеличению полноты покрытия связанных неисправностей. Среди необнаруженных неисправностей сложные связанные неисправности, характеризующиеся лавинообразным (рекурсивным) поведением,

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

которое определяется физическими характеристиками памяти либо возможным эффектом автогенерации, когда ячейки многократно меняют свои состояния на противоположные.

Список использованных источников:

1. Ярмолик, В. Н. *Контроль и диагностика вычислительных систем: [монография] / В. Н. Ярмолик.* – Минск: Бестпринт, 2019. – 387 с: ил. 75.

2. Деменковец, Д. В. *Программное средство моделирования и поиска неисправностей запоминающих устройств / Деменковец Д. В. // Компьютерные системы и сети : сборник тезисов докладов 56-й научной конференции аспирантов, магистрантов и студентов, Минск, апрель-май 2020 года / Белорусский государственный университет информатики и радиоэлектроники. - Минск : БГУИР, 2020. - С. 58-60.*

УДК 621.3.049.77–048.24:537.2

КРИПТОГРАФИЧЕСКОЕ ОБОСНОВАНИЕ СТОЙКОСТИ NFT НА БАЗЕ СМАРТ-КОНТРАКТОВ ETHEREUM

Плетинский И.В., студент гр.951007

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Нестеренков С.Н. – канд. техн. наук

Аннотация. Был изучен стандарт ERC 721 для невзаимозаменяемых токенов на базе смарт-контрактов Ethereum и техническая документация. Были изучены используемые криптографические алгоритмы, структуры данных и алгоритмы подтверждения транзакций в блокчейне. Было дано обоснование криптографической стойкости NFT на базе смарт-контрактов Ethereum. Была разработана методика аудита кода смарт-контракта NFT.

Ключевые слова. NFT, ERC 721, смарт-контракт, SHA-256, эллиптические кривые, доказательство доли владения

Невзаимозаменяемый токен (NFT) — вид криптографических токенов, каждый экземпляр которого уникален, не может быть отредактирован или замещен. Токены могут иметь любое содержание, например, текстовые данные или медиаконтент. Они используются для подтверждения авторства, для электронной цифровой подписи, для подтверждения права собственности на цифровые активы в рамках блокчейна.

В данной статье рассматриваются используемые криптографические алгоритмы и алгоритм консенсуса “Доказательство доли владения”, обеспечивающие уникальность, неподделываемость, необратимость и возможность определения владельца NFT.

Стандарт ERC 721 [1] описывает требования к смарт-контрактам Ethereum для удовлетворения критериям невзаимозаменяемых токенов и для развёртывания смарт-контрактов в любых Ethereum сетях.

Потенциальные вектора атак злоумышленников на смарт-контракты невзаимозаменяемых токенов могут быть направлены на [2]:

- подмена содержимого блоков в блокчейне;
- редактирование содержимого токенов;
- непосредственно кража NFT, т.е. изменения владельца.

Для обеспечения криптографической стойкости смарт-контрактов к перечисленным выше атакам используются следующие алгоритмы:

- алгоритм хэширования SHA256;
- алгоритм цифровой подписи на основе эллиптических кривых;
- алгоритм консенсуса “Доказательство доли владения”.

Алгоритм SHA-256 является криптографической хеш-функцией, которая используется для создания уникального идентификатора фиксированной длины из произвольных данных. Алгоритм обеспечивает целостность данных блока в блокчейне. На рисунке 1 изображен принцип использования алгоритма SHA-256 при записи данных в блокчейн:

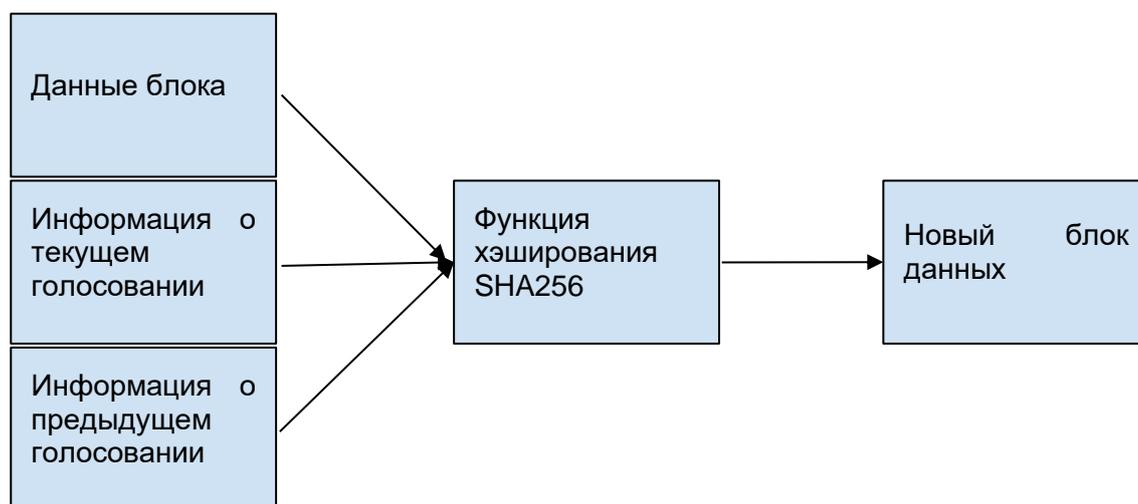


Рисунок 1 – Принцип использования алгоритма SHA-256

Алгоритм цифровой подписи на основе эллиптических кривых (ECDSA) используется для подписи транзакций и обеспечения их целостности. Он основывается на математической теории эллиптических кривых и использует дискретный логарифм для создания ключей и подписей. Для генерации ключей используется приватный ключ, который является случайным числом, и публичный ключ, который является точкой на эллиптической кривой. Эллиптическая кривая в ECDSA — это линия на плоскости, задаваемая уравнением $y^2=x^3+a \cdot x+b$.

Чтобы создать подпись, сообщение сначала хэшируется и затем используется приватный ключ для создания эллиптической кривой. Затем используется публичный ключ для расшифровки кривой и проверки подписи.

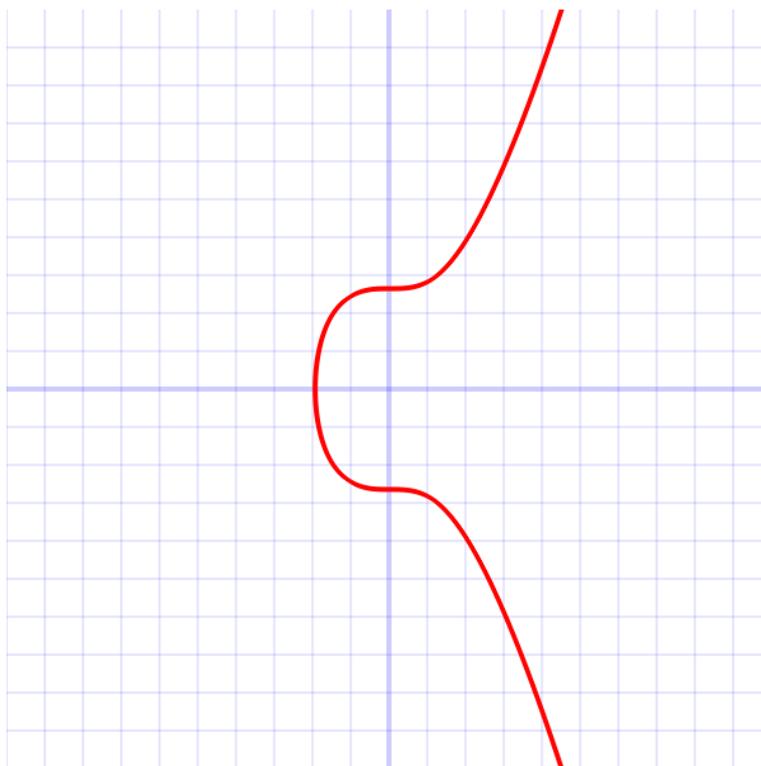
Принцип стойкости ECDSA основывается на “односторонних функциях”, потому что дискретный логарифм на эллиптических кривых очень труден для вычисления. Кроме того, алгоритм имеет высокую скорость работы и меньшее потребление ресурсов, что делает его идеальным для использования в мобильных устройствах и других устройствах с ограниченными ресурсами.

Сети Bitcoin и Ethereum используют эллиптическую кривую secp256k1 [3], уравнение которой $y^2=x^3+7$, определенную в стандарте эффективной криптографии и точку $G(x,y)$, такую, что:

$$x = 79BE667EF9DCBBAC55A06295CE870B07029BFCDB2DCE28D959F2815B16F81798$$

$$y = 483ADA7726A3C4655DA4FBFC0E1108A8FD17B448A68554199C47D08FFB10D4B8$$

На рисунке 2 изображена эллиптическая кривая $y^2=x^3+7$:

Рисунок 2 – Эллиптическая кривая $y^2=x^3+7$

Алгоритм доказательства доли владения (Proof-of-Stake) [4] используется в сети Ethereum в качестве механизма консенсуса участников сети для подтверждения транзакций, т. е. создания и передачи невзаимозаменяемых токенов. Суть алгоритма доказательства доли владения заключается в том, что владельцы криптовалюты могут поставить свою криптовалюту в залог в обмен на возможность быть выбранными для создания новых блоков и получения вознаграждения. Чем больше криптовалюты у участника сети в залоге, тем больше шансов у него быть выбранным. Ставки используются при подписи транзакций, и владелец криптовалюты гарантирует ценой своей ставки целостность и подлинность транзакции.

Изначально сеть Ethereum работала используя механизм доказательство выполнения работы (Proof-of-work). При таком подходе участники сети решают задачи на нахождение простых чисел на больших диапазонах для подписи транзакций используя вычислительные мощности и получают за это вознаграждение в виде криптовалюты. При PoS нет необходимости решать сложные математические задачи, что снижает затраты на электроэнергию и вычислительную мощность. Кроме того, PoS считается более безопасным, так как для атаки на сеть нужно контролировать более 50% криптовалюты, что значительно сложнее, чем контролировать более 50% вычислительной мощности.

Однако, PoS не лишен недостатков. Например, стейкеры могут быть не заинтересованы в обеспечении безопасности сети, так как они получают вознаграждение, даже если они не участвуют в создании блоков. Также, у PoS есть потенциальные проблемы с распределением криптовалюты, так как владельцы большого количества криптовалюты могут получать еще больше криптовалюты в виде вознаграждения.

При алгоритме доказательства доли владения, в отличие от варианта доказательства майнингом, нет необходимости решать сложные математические задачи, что снижает затраты на электроэнергию и вычислительную мощность. Схема алгоритма доказательства доли владения изображена на рисунке 2:

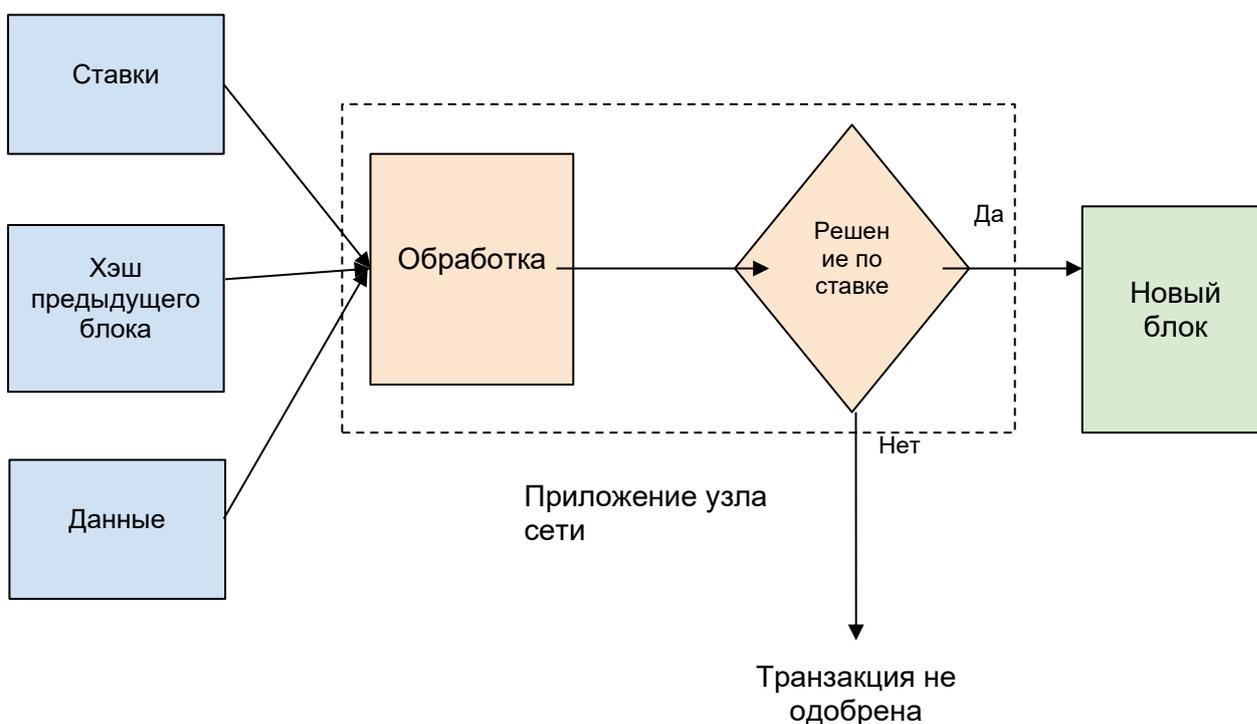


Рисунок 2 – Алгоритм доказательства доли владения

Таким образом, для проверки криптографической стойкости NFT необходимо провести аудит кода смарт-контракта на предмет использования надёжных алгоритмов хэширования SHA и эллиптические кривых для цифровых подписей, а также рассмотреть механизм консенсуса сети.

Были исследованы используемые криптографические алгоритмы в децентрализованных приложениях с использованием смарт-контрактов Ethereum, проанализирован стандарт ERC 721 с точки зрения криптографической стойкости. Было дано обоснование криптографической стойкости NFT на базе смарт-контрактов Ethereum — использование надежных алгоритмов хэширования, цифровой подписи и алгоритма доказательства доли владения делает смарт-контракт стойким к атакам на подмену содержимого блоков, редактированию токенов и краже.

Разработана методика проверки надежности кода смарт-контракта NFT. Исходный код необходимо проверить на факт соответствия стандарту ERC 721, использования алгоритмов хэширования семейства SHA, использования алгоритма цифровой подписи на основе эллиптических кривых и использования алгоритма доказательства доли владения в целевой сети развертываемого приложения.

Список использованных источников:

1. ERC 721: Non-fungible Token standard [Electronic resource] / Ethereum foundation, 2018. – Mode of access: <https://eips.ethereum.org/EIPS/eip-721>. – Date of access : 20.03.2023.
2. Виды атак на блокчейн и умные контракты / Трубач Г. Г. // 75-я научная конференция студентов и аспирантов Белорусского государственного университета: материалы конф. В 3 ч. Ч. 2, Минск, 14–23 мая 2018 г. Белорус. гос. ун-т, Гл. упр. науки – 2018. – С. 278-281.
3. Arithmetic of Koblitz Curve Secp256k1 Used in Bitcoin Cryptocurrency Based on One Variable Polynomial Division / Satoshi Pote, Virendra Sule, B.K. Lande // 2nd International Conference on Advances in Science & Technology (ICAST) 2019 on 8th, 9th April 2019 by K J Somaiya Institute of Engineering & Information Technology, Mumbai, India. – 2019. – p. 333-338.
4. Proof-of-Stake consensus mechanism [Electronic resource] / Ethereum foundation, 2022. – Mode of access: <https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/>. – Date of access : 20.03.2023.

UDC 621.3.049.77–048.24:537.2

JUSTIFICATION OF CRYPTOGRAPHIC STRENGTH OF NFT BASED ON ETHEREUM SMART CONTRACTS

Pletinskij I.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Nesterenkov S.N. – PhD in echnical Sciences

Annotation. The ERC 721 standard for non-interchangeable tokens based on Ethereum smart contracts and technical documentation were studied. Cryptographic algorithms used, data structures and transaction validation algorithms in the blockchain were studied. The cryptographic robustness of NFT based on Ethereum smart contracts was justified. A methodology for auditing NFT smart contract code was developed.

Keywords. NFT, ERC 721, smart contract, SHA-256, elliptic curves, proof of stake

ПРОЕКТИРОВАНИЕ ПАТТЕРНОВ СБОРКИ МОДЕЛЕЙ ОРИГАМИ И АЛГОРИТМ ОЦЕНКИ ИХ ПЛОСКОСКЛАДЫВАЕМОСТИ

Путято М.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Парамонов А. И. – канд. техн. наук, доцент

Рассмотрены известные решения и предложена методика построения паттернов сборки моделей оригами. Рассмотрены известные методы оценки плоскоскладываемости паттерна сборки моделей оригами. Предложен алгоритм оценки плоскоскладываемости паттерна сборки.

Оригами – искусство и наука складывания моделей из бумаги. Помимо широко известных способов сложения моделей оригами при помощи пошаговых схем складывания, существуют альтернативный способ сложить модель – использовать паттерн сборки. Паттерн сборки – один из видов диаграмм оригами, представляющий собой чертёж, на котором изображены все складки базовой формы модели. Хотя результат проектирования паттерна более компактен, процедура его формирования не менее сложна, чем проектирование сотен шагов схемы.

В данной статье предложена методика проектирования паттерна сборки моделей оригами и алгоритм оценки плоскоскладываемости паттерна на основе теорем математики складывания бумаги. Обзор источников по теме показал, что строгого стандарта описания паттерна на данный момент не существует, но можно точно выделить два основных вида складок: горы и долины. Процедура нанесения на чертеж линий сгибов не представляет особой сложности, так как тип сгиба (гора или долина) известен автору заранее. Существуют правила, которые определяют набор операций, которые можно произвести над листом бумаги. Эти правила представляют собой 7 аксиом, более известных как правила Фудзиты [1]:

- пусть заданы две точки p_1 и p_2 , тогда лист можно сложить так, что данные две точки будут лежать на складке;
- пусть заданы две точки p_1 и p_2 , тогда лист можно сложить так, что одна точка перейдёт в другую;
- пусть заданы две прямые l_1 и l_2 , тогда лист можно сложить так, что одна прямая перейдёт в другую.;
- пусть заданы прямая l_1 и точка p_2 , тогда лист можно сложить так, что точка попадёт на складку, а прямая перейдёт сама в себя (то есть линия складки будет ей перпендикулярна);
- пусть заданы прямая l_1 и две точки p_1 и p_2 , тогда лист можно сложить так, что точка p_2 попадёт на складку, а p_1 – на прямую l_1 ;
- пусть заданы две прямые l_1 и l_2 и две точки p_1 и p_2 , тогда лист можно сложить так, что точка p_1 попадёт на прямую l_1 , а точка p_2 попадёт на прямую l_2 ;
- пусть заданы две прямые l_1 и l_2 и точка p_1 , тогда лист можно сложить так, что точка p_1 попадёт на прямую l_1 , а прямая l_2 перейдёт сама в себя (то есть линия складки будет ей перпендикулярна).

Эти правила могут быть использованы в качестве возможных вариантов нанесения складок на паттерн и реализованы в программном средстве.

Когда складывание оригами позиционируется как геометрическая задача, предполагается, что модель может быть сложена в плоском виде. Плоское складывание – складывание модели таким образом, чтобы она была плоской и при этом ее целостность не была нарушена путем разрезания, склеивания и других подобных операций. В основе классического складывания всегда стоит единственный лист бумаги (в большинстве случаев квадратный).

Существуют несколько необходимых, но не достаточных условий плоскоскладываемости паттерна. Эти правила описывают углы между сгибами и присваивание сгибу типов (гор или долин). Задача о плоскоскладываемости одной вершины поддается решению с помощью нескольких важнейших теорем. Среди них можно отметить теорему Мазкавы [2], а также теорему Кавасаки [3].

Теорема Кавасаки гласит, что сумма чередующихся углов, образованных складками вокруг вершины, должна быть равна 180 градусам для того чтобы паттерн мог быть сложен в плоской форме. Рассмотрим пример на рисунке 1.

Согласно данному рисунку, имеет место следующее выражение:

$$a + c + e = b + d + f \quad (1)$$

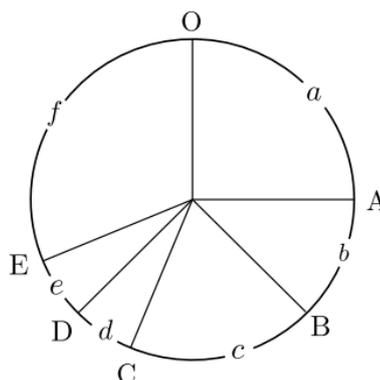


Рисунок 1 – Паттерн одной вершины со складками

Однако стоит заметить, что данная теорема не учитывает тип складок, которые используются. Правило для типов складок описывается в теореме Маэкавы. Теорема Маэкавы гласит, что количество складок-гор и количество складок-долин одной вершины должны отличаться на 2, чтобы паттерн мог быть сложен в плоской форме.

На рисунке 2 представлены два паттерна: первый паттерн удовлетворяет теореме Маэкавы, а второй нет. Если попробовать сложить данные паттерны, можно убедиться в корректности вышеописанной теоремы.

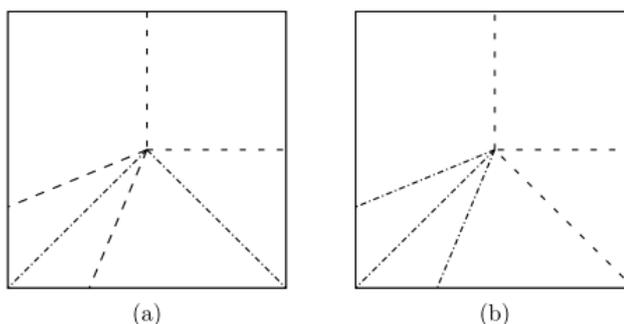


Рисунок 2 – Паттерн сборки, удовлетворяющий теореме Маэкавы, (слева) и паттерн, не удовлетворяющий ей (справа)

Если паттерн состоит из единственной вершины, то можно сказать, что он плоскоскладываемый, если удовлетворяет двум вышеописанным теоремам. Однако было доказано, что определение, является ли определенный паттерн сборки складываемым в плоском виде, является NP-полной задачей [4]. Ситуация, когда случайным образом сформированный паттерн является плоскоскладываемым редка. Однако зачастую в определении глобальной плоскоскладываемости паттерна нет необходимости, так как почти всегда автор проектирует паттерн на основе уже сложившейся модели или на основе хорошо известных базовых форм. Тем не менее, корректность паттерна может быть оценена одним из необходимых условий глобальной плоскоскладываемости, а именно условием локальной плоскоскладываемости. Примем такое определение, что паттерн является локально плоскоскладываемым, если каждая его вершина является плоскоскладываемой. Таким образом, используя теорему Кавасаки и теорему Маэкавы, можно однозначно утверждать является ли паттерн локально плоскоскладываемым. Этого вполне достаточно, если идет проектирование паттерна уже сложившейся в плоском виде модели, поскольку проектирование паттерна происходит уже после того, как автор собрал модель.

В рамках исследований по теме был проведен анализ существующих теорем в области математики складывания бумаги. На основе анализа подходов предложено использование авторской методики проектирования паттерна сборки моделей оригами на основе правил Фудзиты. Для дальнейшей программной реализации сборки был составлен алгоритм оценки плоскоскладываемости паттерна, на основе определения его локальной плоскоскладываемости с помощью теорем Кавасаки и Маэкавы.

Список использованных источников:

1. Robert J. Lang. *Huzita-justin axioms. Robert J. Lang Origami. [Электронный ресурс]. – Режим доступа: <http://www.langorigami.com/article/huzita-justin-axioms> – Дата доступа: 01.04.2023.*

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. Thomas C. Hull. *The combinatorics of flat folds: a survey*. In *Origami*. AK Peters, 2002. [Электронный ресурс]. – Режим доступа: <http://arxiv.org/abs/1307.1065>. – Дата доступа: 01.04.2023.
3. *Roses, Origami & Math* / Toshikazu Kawasaki // *Japan Publications Trading Company*. – 1998. – Vol. 19. – P. 136–173.
4. *The complexity of flat origami* / M. Bern, B. Hayes // *In Proceedings of the 7th ACM/SIAM Symposium on Discrete Algorithms*. – 1996. – P. 175–183.

ПРОГРАММНОЕ СРЕДСТВО «НАВЫК ДЛЯ ГОЛОСОВОГО ПОМОЩНИКА АЛИСА. ИЗУЧЕНИЕ АНГЛИЙСКОГО ЯЗЫКА» С ИСПОЛЬЗОВАНИЕМ СЕРВИСА ЯНДЕКС. РАЗРАБОТЧИК

Радько А. А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Видничук В. Н – м.т.н.

Изучение иностранных языков всегда происходит с большими трудностями, но на данный момент мы сталкиваемся с тем, что очень важно знание английского языка. В данной работе предлагается программное средство «НАВЫК ДЛЯ ГОЛОСОВОГО ПОМОЩНИКА АЛИСА. ИЗУЧЕНИЕ АНГЛИЙСКОГО ЯЗЫКА» с использованием сервиса Яндекс. Разработчик для более легкого и понятного изучения иностранного языка в игровой форме.

Важной вехой развития компьютерных технологий стало создание интернета и, в частности, «Всемирной паутины». Важной частью всемирной паутины являются голосовые помощники. По статистике на начало 2023 года число используемых цифровых голосовых помощников составляет более 8 миллиардов [1].

Такое количество ресурсов предоставляет пользователям возможности, о которых 30 лет назад было невозможно представить. За эти годы было создано: большое количество чат-ботов, бесчисленное множество веб-инструментов. Использование искусственного интеллекта и других технологий помогает сократить до 30% затрат на поддержку клиентов в бизнесе, обеспечить нужный уровень персонализации, а также быстрее решать клиентские запросы.

Главной задачей голосового помощника является максимально быстрое выполнение запросов с последующей обработкой данных. Ассистент способен распознавать голос пользователя и выполнять заданные команды. При этом многие системы обладают широким набором полезных возможностей. К самым популярным можно отнести: выполнение звонков по заданным номерам, поиск различной информации, открывание/закрывание приложений на устройстве, включение музыки, фильмов и прочего контента, активация будильника, создание заметок, общение, отображение прогноза погоды и прочее.

Таким образом в обществе сформировался запрос на более продвинутые расширения пополнения знаний, изучения нового. В связи со всем описанным ранее, было принято решение разработать программное средство «Навык для голосового помощника Алиса. Изучение английского языка» в виде клиент-серверного расширения на языке JAVA с применением фреймворков Spring, Hibernate и внедрения сервиса Яндекс. Разработчик.

Разработка данного программного средства обеспечит пользователям возможность общения с голосовым помощником Алиса на английском языке, улучшать произношение, пополнять словарный запас новыми словами.

Было проанализировано множество аналогичных приложений и было определено, что большинство из них позволяют изучать только новые слова разных уровней сложности, а если и есть поддержка разговора в реальном времени, то обычно это определенные фразы, который пользователь со временем запоминает и изучение становится неинтересным.

Для повышения удобства использования веб-приложения пользователями, приложение выполнено с применением адаптивной верстки и полностью функционально на настольных компьютерах, планшетных компьютерах, смартфонах и колонках, с поддержкой голосового помощника Алиса.

Разрабатываемое программное средство может быть размещено в системе облачного хостинга с поддержкой системы Алисы, для безграничного пользования программным средством.

В данной работе были проанализированы аналоги, найдены сильные и слабые стороны данной категории приложений. В результате было принято решение о разработке программной системы, с устранением обнаруженных недочетов и установкой цели на улучшение многопользовательского опыта.

В ходе разработки были отработаны на практике навыки проектирования и разработки программных систем и, в частности, веб-приложений, использования современных шаблонов проектирования, применены на практике современные фреймворки разной сложности в освоении и масштабах. В перспективе возможно улучшение системы путем добавления новых модулей и *различных языков*, систем коммуникации между пользователями и улучшения пользовательского интерфейса для большего качества пользовательского опыта использования приложения.

Список использованных источников:

1. RG.digital.ru [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://rg.ru/2021/06/15/k-2023-godu-chislo-golosovyh-pomoshchnikov-v-mire-dostignet-vosem-mlrd.html>. Дата доступа: 15.02.2023.
2. Lavender A. ShoeBOX shuffle/ Lavender A. — том 3 — Proquest Social Sciences Journals, 1991. — 34

ВЕБ-ОРИЕНТИРОВАННОЕ ПРОГРАММНОЕ СРЕДСТВО «СИСТЕМА СТРАХОВАНИЯ ЖИЗНИ И ЗДОРОВЬЯ»

Рафеев П. Ю.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Болтак С. В. – ассистент

Данная работа посвящена разработке веб-ориентированного программного средства, объединяющего в себе функции автоматизации документооборота страховой компании, интернет-портала с информацией о страховании жизни и здоровья и системы подачи онлайн-заявок на страхование. Страхование жизни и здоровья выбрано как наиболее активно развивающийся сегмент страхового рынка Республики Беларусь и стран СНГ.

Страхование является формой управления рисками, разработанной для защиты от финансовых потерь, в которой одна сторона, называемая страховщиком, страховой компанией, страховым перевозчиком или андеррайтером, соглашается компенсировать другой стороне – страхователю или застрахованному лицу убытки, повреждения или травмы в обмен на оплату.

Перспективы цифровизации страхования и продажи страховых продуктов через интернет обсуждались еще в конце 1990-х гг., однако ее реализация в практической деятельности страховых компаний затянулась. Главными причинами были недостаточный уровень развития информационных технологий в целом и низкий уровень компьютеризации и интернетизации большинства населения постсоветских стран. Одной из причин задержки стала неготовность к изменениям как страхователей, которые не доверяли в этом страховым компаниям, так и самих страховщиков, опасавшихся развития страхового мошенничества.

В 2010-е годы процесс цифровизации во всех сферах жизнедеятельности значительно ускорился. О внедрении цифровых технологий в реальную страховую практику стран СНГ свидетельствует увеличение доли электронных полисов обязательного страхования автогражданской ответственности (ОСАГО) в общем количестве вновь заключаемых договоров. Если в январе 2016 г. она не превышала 1%, к началу 2018 г. этот показатель достиг 39%, а в 2019 г. составлял уже 56% [1].

На современном страховом рынке информационные технологии значительно повлияли на отрасль, став той движущей силой, которая позволяет страховым компаниям упрощать многие бизнес-процессы, успешно взаимодействовать со все более требовательными и нетерпеливыми клиентами. При этом ожидается, что в связи с развитием цифровизации на рынке труда в сфере страхования произойдут изменения, которые могут привести к высвобождению персонала или изменению структуры занятости работников страховых компаний. Использование данных, предоставляемых страхователями, связано с рисками раскрытия конфиденциальной информации и незаконного использования персональных данных [2].

Несмотря на обозначенные риски, цифровизация страхового рынка несет в себе положительные аспекты потенциальной экономии на комиссиях и снижения расходов, т.е. повышение эффективности и прибыльности страховых операций. Помимо этого к положительным последствиям цифровизации страхового рынка относятся:

- разработка индивидуальных предложений страховых услуг на основе анализа частной информации и ее сопоставления с большими массивами данных;
- конвергенция взаимного и фондового (коммерческого) страхования;
- социализация страховых отношений;
- появление новых страховых услуг и продуктов;
- реализация функции ведения и управления договорами страхования в «умных» устройствах (интернет вещей).

Благоприятными для цифрового рынка страхования являются растущее внимание населения к своему здоровью на фоне пандемии коронавируса, желание получить качественную помощь помимо обязательного медицинского страхования и всё большая востребованность услуг телемедицины. Это делает страхование жизни и здоровья наиболее интенсивно развивающимся сегментом рынка онлайн-страхования [3].

Целью данной работы стала разработка веб-ориентированного программного средства страхования жизни и здоровья, реализующего следующие функции:

- демонстрация пользователям политик, условий и иной информации по страхованию;
- создание пользователями заявок на прикрепление, открепление к страховым планам себя и своих родственников, а также на изменение личных данных и смену плана страхования;
- автоматизация документооборота, связанного с заключением страховых договоров.

Сочетание всех указанных функций в одном программном средстве избавляет страховую компанию от необходимости использовать отдельные средства для автоматизации документооборота, взаимодействия с заказчиками и предоставления информации о страховании, а также позволяет унифицировать и удешевить данные процессы.

Система предусматривает 2 роли пользователей: клиенты и администраторы. Диаграмма вариантов использования для них приведена на рисунке 1.



Рисунок 1 – Диаграмма вариантов использования системы клиентами и администраторами

Разработка веб-ориентированного программного средства произведена на языке C# с использованием системы управления базами данных «MS SQL». В качестве архитектуры использовано сочетание MVC подхода (для запросов, изменяющих данные и запросов, возвращающих страницы) и REST (преимущественно для запросов на выгрузку отчётов и иных документов в форматах Word и Excel). Пользовательский интерфейс системы представлен Razor-страницами и использует сценарии на языке JavaScript для выполнения действий, не требующих отправки запросов на сервер. Пример страницы программного средства представлен на рисунке 2.

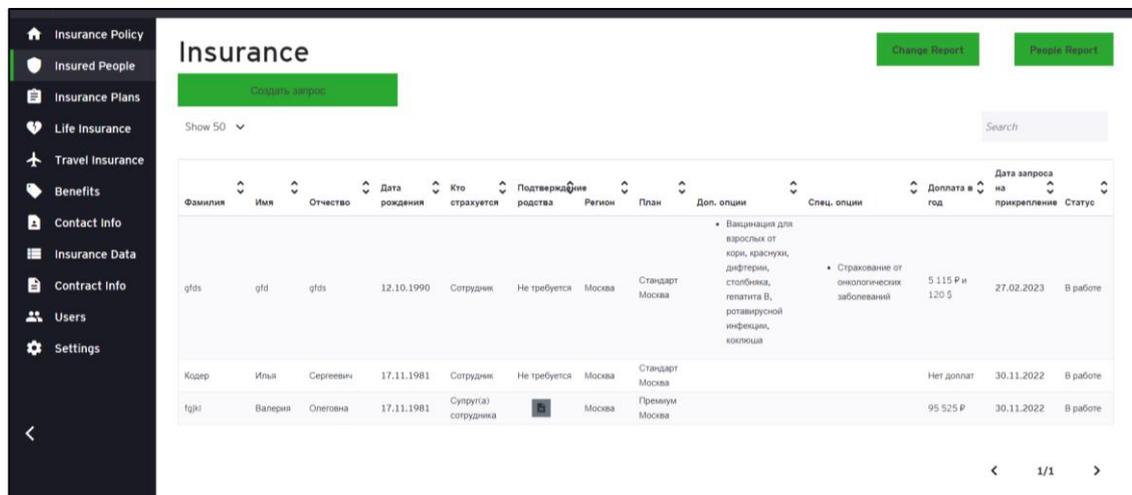


Рисунок 2 – Пример страницы программного средства

В заключение необходимо отметить, что разработанное веб-ориентированное программное средство страхования жизни и здоровья позволит страховщикам повысить скорость и качество обработки растущих потоков информации, подбора на основе анализа данных оптимальных предложений, следовательно, и качество обслуживания клиентов-страхователей, а клиентам, в свою очередь, даст возможность быстро и удалённо ознакомиться со всеми интересующими их аспектами страхования жизни и здоровья, выбрать план и оставить заявку на страхование.

Список использованных источников:

1. Будникова, И. К. Цифровая трансформация в сфере страхования / И. К. Будникова, А. М. Марданова // Национальная (с международным участием) научно-практическая конференция «Современные цифровые технологии: проблемы, решения, перспективы – 2022» сб. науч. тр. / отв. ред. Э.Ю. Абдуллазянов. – Казань: КГЭУ, 2022. – С. 143-144.

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. Бедердинова, А.И. Современное онлайн-страхование в России / А. И. Бедердинова, С.А. Калайда, И.А. Прилепкина // Экономика, предпринимательство и право. – 2021. – Т 11. – №12. – С. 2–3.

3. Волков А. Ю. Перспективы цифровизации в страховой сфере в России и зарубежных странах / А. Ю. Волков, А. А. Яркин // Сборник материалов межвузовской научно-практической конференции с международным участием «Юриспруденция 2.0: Новый взгляд на право». – Москва: РУДН, 2019. – С. 611.

ВЕБ-ПРИЛОЖЕНИЕ “ЦИФРОВОЙ УСИЛИТЕЛЬ С ЭФФЕКТАМИ” НА БАЗЕ ЯЗЫКА JAVASCRIPT

Ролевич С.И.

Белорусский государственный университет информатики и радиоэлектроники
Г. Минск, Республика Беларусь

Потапов В.Д. – канд. тех. наук, доцент

Аннотация. В работе рассматриваются способы обработки звуковых сигналов в реальном времени с помощью возможностей языка javascript.

Ключевые слова. Сигнал, звук, аудио контекст, наложение эффекта

У любого начинающего музыканта в определённый момент возникает необходимость в новой аппаратуре. В поиске нового звука можно перебрать множество инструментов, процессоров эффектов, устройств воспроизведения. Основная проблема – стоимость оборудования. С этой проблемой может помочь веб-приложение “Цифровой усилитель с эффектами”. Для удобного использования приложения хватит обычного микрофона, но для более комфортного использования необходима аудио карта с интерфейсом Jack 6.3.

Веб-приложение позволяет в режиме реального времени накладывать эффекты на звуковой сигнал пользователя, открывать аудиофайлы, осуществлять запись и экспортировать запись в виде файла. Работа веб-приложения организована на основе веб-технологии Web Audio API, позволяющей манипулировать звуковой составляющей на веб-странице.

Web Audio API позволяет взаимодействовать как с пользовательским аудио сигналом в режиме реального времени, так и с аудиофайлами. Базовые операции выполняются с помощью аудио узлов. Каждый из узлов можно отнести к трём типам: источники сигнала, фильтры сигнала, узлы вывода. Впоследствии узлы объединяются, формируя граф узлов. Схематическое представление графа узлов представлено на рисунке 1. [1]

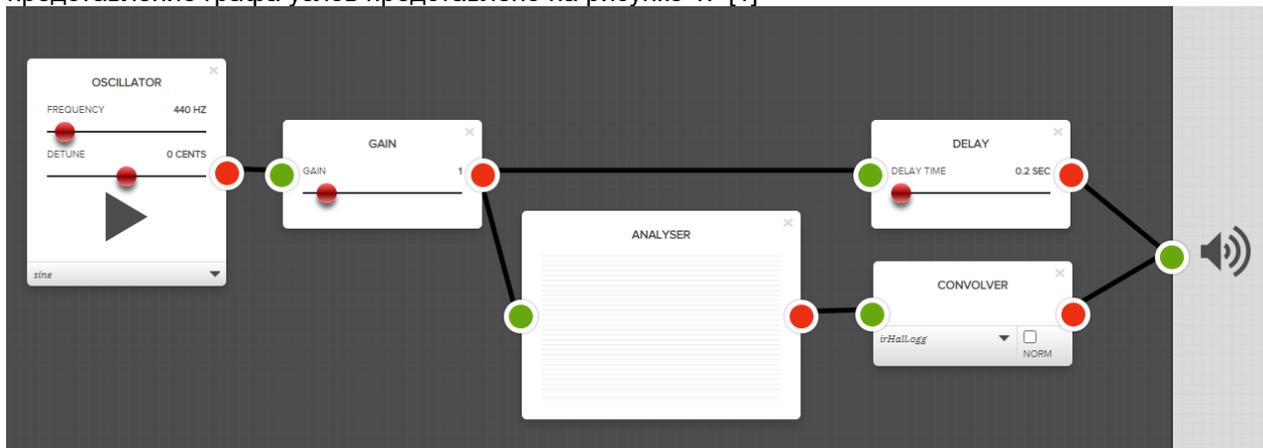


Рисунок 1 – Пример графа узлов Web Audio API

Приложение позволяет наложить следующие эффекты на звуковой сигнал: Компрессия, Delay, Distortion, Flanger, Quadrafuzz, Reverb, Tremolo. Каждый из эффектов представляет собой отдельный аудио узел, который на входе принимает дискретизированный сигнал, и возвращает преобразованный сигнал в следующий узел. Благодаря системе узлов возможна удобная для пользователя реализация изменения порядка наложения эффектов, их добавление и удаление.

Также веб-приложение позволяет сохранять свои цепочки графов для зарегистрированных пользователей. Клиентская сторона представляет собой многостраничное приложение, использующее фреймворк ReactJS. Для разработки серверной составляющей выбрана платформа NodeJS. Взаимодействие между клиентом и сервером происходит посредством PostgreSQL.

Интерфейс приложения представляет собой рабочую область со звуковыми дорожками, перемещение по которым реализовано с помощью временного маркера.

Список использованных источников:

1. Документация Web Audio API [электронный ресурс]. – Режим доступа: https://developer.mozilla.org/ru/docs/Web/API/Web_Audio_API. Дата доступа: 30.03.2023

ПРОГРАММНОЕ СРЕДСТВО АНАЛИЗА ТЕКСТОВ ПУБЛИКАЦИЙ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ FLUTTER И JAVA SPRING BOOT

Синицкая К.Д.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Сурков Д.А. – ст. преподаватель

В рамках данной работы рассматривается программное средство, позволяющее решать задачу получения новостных публикаций из различных источников, автоматической группировки и отображения их по темам.

На практике есть огромное множество задач, связанных с анализом текста на естественном языке. В рамках данной работы под анализом текста понимается процесс получения, изучения и обработки неструктурированных данных, представленных в виде текста.

Среди подзадач задачи анализа текста можно выделить распознавание текста, исправление ошибок, генерацию текста, выделение частей текста, устранение двусмысленности, перевод, анализ тематики, информационный поиск и многое другое. Выделяется среди этих подзадач проблема получения новостных публикаций из различных источников, автоматической группировки и отображения полученных новостных публикаций по темам. В рамках этой задачи также может производиться оценка эмоциональной окраски публикации, а, в связи с этим, достоверности и правдивости рассматриваемой новости. Возможность настройки анализа эмоциональной оценки может быть полезна для тех людей, которые изначально хотят знать, какие публикации по тональности будут совпадать с их мнением, а какие будут ему противоречить. Это позволяет иметь первоначальную оценку публикации, отталкиваясь от которой любой человек может решить, стоит ли ознакомиться с материалами статьи или он считает, что материалы публикации могут оказаться недостоверными, содержащими только субъективное мнение автора, не подкрепленное фактами.

Для решения проблемы группировки текстов публикаций по темам существует несколько основополагающих подходов, среди которых хорошо зарекомендовал себя подход, основанный на использовании морфологии, выделении частей текста, словосочетаний и морфем. На практике есть несколько вариантов реализации данного решения, среди которых основными являются использование нейронных сетей, создание специализированного ПО с жесткими настройками и комбинированный метод.

Подход с использованием нейронных сетей не оправдывает себя в ситуациях, когда может понадобиться быстрая подстройка ПО, т.к. изначально нейронные сети необходимо обучить на некоторой выборке, а уже затем такое приложение можно будет использовать. Также нейронные сети не дают пояснений, почему они приняли то или иное решение. В свете того, что темы новостных публикаций быстро появляются и меняются, становится очевидно, что использование нейронных сетей не сможет позволить приложению быстро подстроиться и отображать актуальную информацию: пока пройдет процесс обучения нейронной сети на выборке, то может потребоваться переобучение сети для выделения все новых и новых тем.

Необходимость постоянной надстройки зачастую требует создания нового специализированного ПО, которое готово к постоянно изменяющимся требованиям, настройкам анализа и классификации, поэтому было принято решение о создании такого приложения. Учитывая, что большую часть информации в наше время люди получают благодаря интернету, а для быстрого просмотра такого рода информации, к которой относятся и новости, используют телефон, стало очевидно, что лучшим вариантом станет создание приложения с клиент-серверной архитектурой, где в качестве клиента выступит мобильное приложение, а сервер будет построен с использованием быстрых и мощных средств, как архитектурный стиль REST.

В результате было создано такое программное средство, которое получает новостные публикации, автоматически группирует их по темам и отображает пользователю с оценкой тональности каждой конкретной публикации. Данное приложение позволяет отображать конечному пользователю новостные публикации, сгруппированные по темам, показывает оценку тональности каждой публикации, позволяет производить свои настройки классификации, формировать объекты мониторинга, сохранять избранные публикации и т.д.

Список использованных источников:

1. Мэннинг, К.Д. *Основы статистической обработки естественного языка / К.Д. Мэннинг, Х. Шютц. – Москва: Техносфера, 2013. – 1048 с.*

ПРОГРАММНОЕ СРЕДСТВО АВТОМАТИЗИРОВАННОГО СБОРА ИНФОРМАЦИИ HTML-САЙТА ПОСРЕДСТВОМ ТЕХНОЛОГИИ RUBY

Страчинский Н.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Сурков К.А. – ст. преп.

Выполнен анализ задач, решаемых автоматизированным сбором информации. Произведён анализ пригодности способов чтения данных для сбора информации HTML-сайта. Описаны преимущества расширения браузера для формирования наборов команд автоматизированного сбора информации HTML-сайта. Рассмотрена инфраструктура фоновых задач Sidekiq и синтаксического анализатора Nokogiri для выполнения поставленной задачи на языке программирования Ruby.

Процесс автоматизированного сбора информации, или же парсинг, заключается в получении и систематизировании данных из различных источников. С его помощью можно решать такие задачи, как сравнение цен и оценка конкуренции, обучение и тестирование моделей машинного обучения, наполнение каталогов информацией из открытых источников, извлечение данных о продукте или услугах третьих лиц для платформы электронной коммерции [1]. HTML-документ использует предопределённые теги и атрибуты, чтобы указать браузеру, как отображать контент.

Канонически для автоматизированного сбора информации приняты интерфейсы DOM (Document Object Model) и SAX (Simple API for XML). DOM предоставляет дерево объектов, которое содержит всю информацию о документе, он удобен для произвольного доступа к различным местам контекста данных, но при этом не экономит память, поскольку ему требуется читать весь документ целиком и сохранять дерево элементов в памяти. Интерфейс SAX же управляет событиями. Поэтому с его помощью проблематично получить произвольный доступ к различным местам контекста данных. SAX экономит память, поскольку приложение может хранить лишь ту часть документа, которая представляет интерес [2].

Среди требований к интерфейсам автоматизированного сбора информации можно выделить произвольный доступ к различным местам контекста данных для упрощения получения лишь требуемой информации. Также следует учитывать наличие в анализируемом HTML-документе как одинарных, так и парных тегов, присутствующих в контексте данных. Таким образом, можно прийти к выводу о пригодности DOM интерфейса для автоматизированного сбора информации HTML-сайта. Для произвольного доступа к объектам DOM дерева разумно использовать язык запросов XPath [3]. Для работы с рассмотренным языком запросов существует программная библиотека для анализа XML и HTML документов, работающая на языке программирования Ruby – Nokogiri. Данный синтаксический анализатор позволяет читать и изменять DOM дерево.

Набор команд для чтения информации с помощью XPath следует задать и хранить. Среди уверенных пользователей компьютера достаточно людей, навыков которых не хватает для написания кода, поэтому большую популярность набирают посode-подходы. Для разработки ИТ-продуктов с данным подходом требуется ПО, позволяющее визуальнo моделировать события.

Таким образом, программное средство автоматизированного сбора информации должно взаимодействовать с HTML-сайтом и при этом иметь пользовательский интерфейс, подходящий для визуального моделирования событий. Для вышеупомянутых целей в качестве ПО как можно лучше подходит расширение браузера. Данный тип программ не только является веб-приложением с пользовательским интерфейсом, но и способен встраивать код и выполнять его в открытом в браузере сайте. Следовательно, пользователь имеет возможность в режиме реального времени взаимодействовать с DOM-объектами для выбора требуемых для сбора HTML-элементов.

Составленный в пользовательском интерфейсе набор команд должен постоянно выполняться и собирать информацию для сохранения актуальности данных. Для данной цели предусмотрена инфраструктура Sidekiq, позволяющая управлять фоновыми задачами, которые можно реализовать на языке программирования Ruby.

Таким образом, можно сделать вывод о целесообразности создания программного средства автоматизированного сбора информации HTML-сайта, пользовательский интерфейс которого будет представлять из себя расширение для браузера, а внутренняя реализация – программный код, который в очереди считывает и запускает набор команд, предоставленный пользователем.

Список использованных источников:

- 1 *What is Web Scraping used for? [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.webharvy.com/articles/web-scraping-use-cases.html>.*
2. *XML Parsing, SAX/DOM [Электронный ресурс]. – Электронные данные. – Режим доступа: https://ranger.uta.edu/~cli/pubs/2009/XMLParsing_ChengkaiLi.pdf.*
3. *Парсим сайт при помощи XPath [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://siteclinic.ru/blog/seo-instrumenty/parsim-sajt-s-xp>*

ПРОГРАММНОЕ СРЕДСТВО МОДЕЛИРОВАНИЯ СВЯЗАННЫХ НЕИСПРАВНОСТЕЙ НА ОСНОВЕ НЕИСПРАВНОСТЕЙ ВЗАИМНОГО ВЛИЯНИЯ

Тюшев Т.А., студент гр.951003, Деменковец Д.В., аспирант

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Леванцевич В.А. – ст. преподаватель

Аннотация. В данной работе рассматривается архитектура симулятора неисправностей запоминающих устройств, ориентированного на неисправности взаимного влияния трёх и более ячеек памяти. Описывается модель и нотация связанной неисправности, состоящей из трех ячеек. Представлен алгоритм генерации неисправностей и внесения их в модель памяти. Описывается программное средство, моделирующее процесс тестирования модели памяти с помощью маршевых тестов.

Ключевые слова. ОЗУ, запоминающие устройства, тестирование ОЗУ, маршевые тесты, неисправности взаимного влияния, программное средство.

В настоящее время ОЗУ является неотъемлемой частью большинства современных вычислительных систем. Совершенствование технологий процесса изготовления элементов ОЗУ позволяют повышать степень интеграции, но также приводит к появлению большего количества дефектов. Это обусловлено увеличением плотности размещения элементов и уменьшением техпроцесса. Поэтому на сегодняшний день остается актуальной задача тестирования памяти [1].

Среди моделей неисправности запоминающих устройств связанные неисправности занимают одно из особых мест в силу сложности их обнаружения. При этом возможны различные типы и подтипы неисправностей, участвующих в конкретной связанной неисправности. В данных неисправностях часто участвуют более двух ячеек памяти. Их способность влиять на поведение друг друга приводит к маскированию неисправности и делает разработку алгоритма тестирования очень сложной задачей [2].

Для реализации симулятора необходимо создать модель взаимодействия неисправных ячеек, обеспечивающую взаимное влияние на несколько других ячеек. Например, при неисправности трёх ячеек может быть ситуация, когда каждая из них влияет на поведение остальных двух. Для визуализации неисправности используется граф, на котором изображены три ячейки памяти и стрелками показано влияние ячеек друг на друга. Над стрелками показан тип неисправности. Пример графа такой неисправности представлен на рисунке 1 (а).

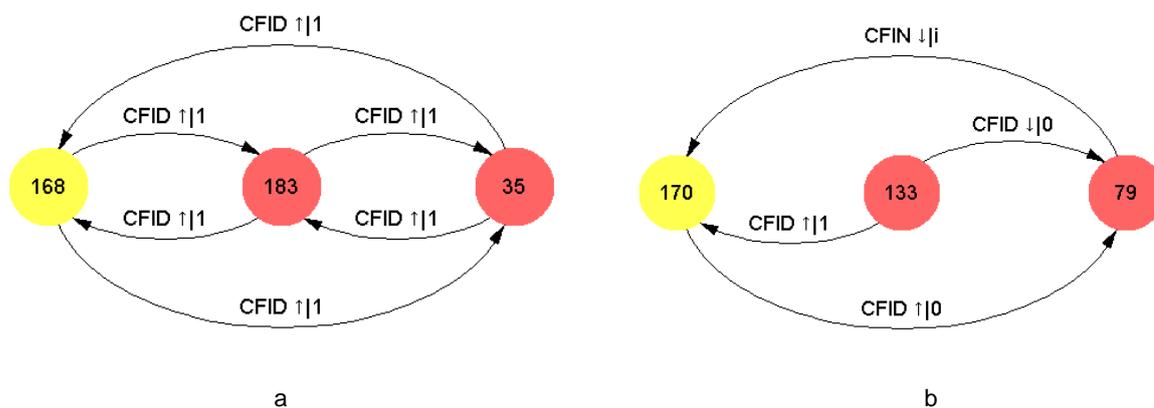


Рисунок 1 – Графы неисправностей взаимного влияния для трёх ячеек

Для удобного ввода данных об неисправностях используется нотация, описывающая неисправности взаимного влияния нескольких ячеек. Примером описания связанной неисправности для трёх ячеек может быть следующая строка: $\langle -, \uparrow, 1, i \rangle \langle \uparrow, \downarrow, -, - \rangle \langle \downarrow, -, 0, 0 \rangle$. Модель неисправности, соответствующая данной нотации представлена на рисунке 1 (b).

Для программной реализации поведения данных неисправностей необходимо, чтобы каждая ячейка поддерживала список ячеек, на которые она, как агрессор, влияет (массив жертв). Также необходимо хранить и тип ошибки каждой жертвы (тип влияния на ту или иную ячейку). Каждая ячейка поддерживает набор методов для тестирования её поведения, которые соответствуют стандартным операциям с памятью – r0 (чтение 0), r1 (чтение 1), w0 (запись 0), w1 (запись 1).

Ячейка жертва является отдельной сущностью инкапсулирует в себе ячейку памяти и тип неисправности. Диаграмма классов модели памяти представлена на рисунке 2 (а).

Основой модели всех неисправностей является интерфейс Fault. Для связанных неисправностей предусмотрен отдельный интерфейс LinkedFault, который реализует методы, оповещающие ячейку жертвы, что произошла определённая операция для ячейки агрессора. Диаграмма классов модели неисправностей представлена на рисунке 2 (b).

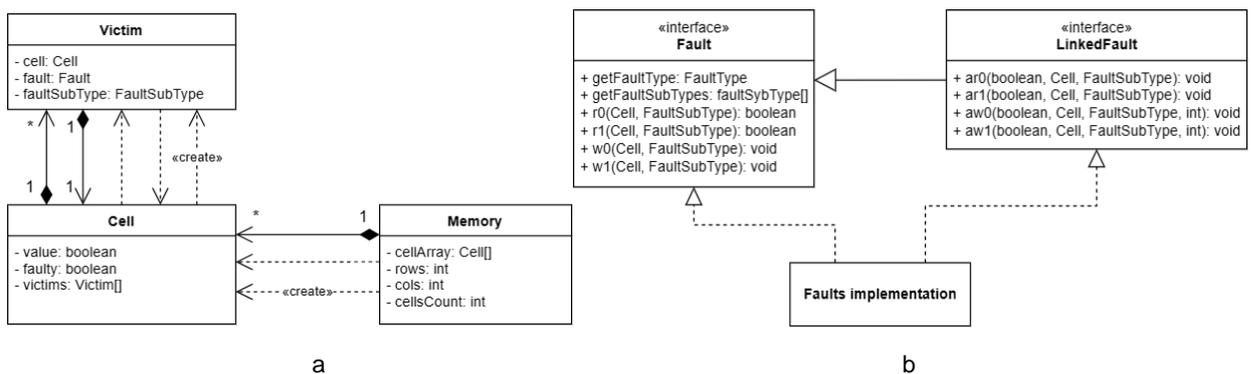


Рисунок 2 – Диаграмма классов программного средства

Таким образом, объект неисправности «следит» за поведением ячейки жертвы, тем самым является «наблюдателем». При этом происходит вызов определенных методов, когда выполняется какая-либо операция с ячейкой, у которой имеются ячейки жертвы. Реализация данных методов содержит логику неисправностей и изменяет состояние ячеек жертв в соответствии с типом неисправности и операцией над агрессором.

Каждый метод вызывающий изменение состояния ячейки жертвы содержит защиту от рекурсивного вызова для случая, когда две ячейки влияют друг на друга и могут бесконечно инвертировать значение друг друга. Данное решение позволяет смоделировать неисправности взаимного влияния любой сложности с любым количеством ячеек.

Для проверки алгоритмов тестирования в программном средстве предусмотрено два способа внесения неисправностей в модель памяти: с помощью текстового файла и случайной генерации.

Текстовый файл нотации содержит адреса ячеек, и тип неисправности в нотации, описанной выше. Пример записи в таком файле:

LR3<149,-,↓,-,i><113,-,↓,-,-><5,↓,-,0,1>

Пример генерации неисправности из файла данной нотации представлен на рисунке 3 (а).

Другой способ позволяет по заданному типу неисправности случайно сгенерировать нужное количество неисправностей. Для этого вводится тип неисправности в нотации, описанной выше и количество неисправностей. Для генерации адресов неисправных ячеек используется случайная последовательность неповторяющийся чисел.

Пример случайной генерации неисправностей:

- Размер памяти 16x16 ячеек
- Ввод типа неисправности: <-,↑,i,-><↓,↓,-,i><-,↑,i,0>;
- Ввод количества неисправностей: 2;
- Генерация последовательности адресов: 177, 35, 61, 93, 170, 236;
- Внесение неисправностей в модель памяти.

Результат генерации неисправностей случайным образом представлен на рисунке 3 (b) и (c).

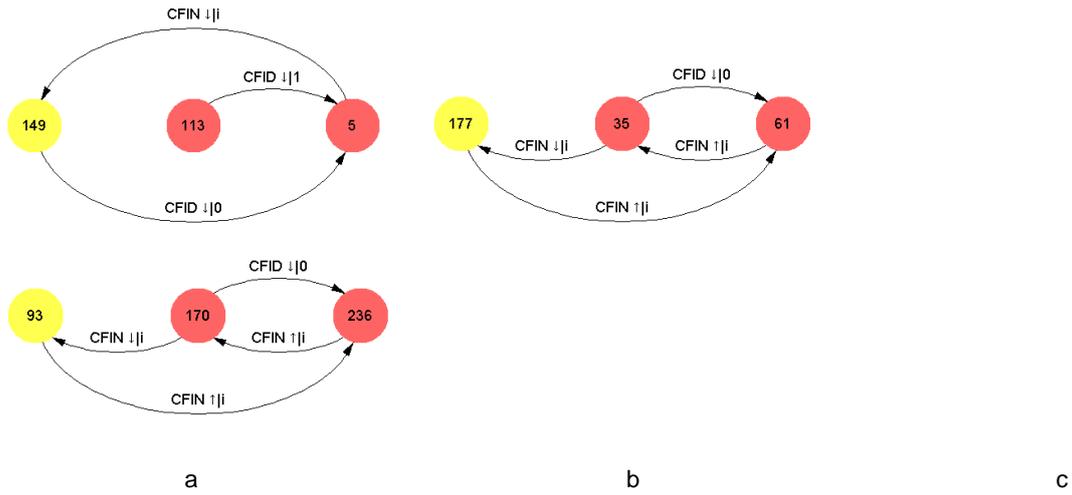


Рисунок 3 – Примеры сгенерированных неисправностей

Модель памяти после внесения случайно сгенерированных неисправностей представлена на рисунке 4 (а). Результат тестирования модели памяти с обнаруженными и не обнаруженными неисправностями на рисунке 4 (b).

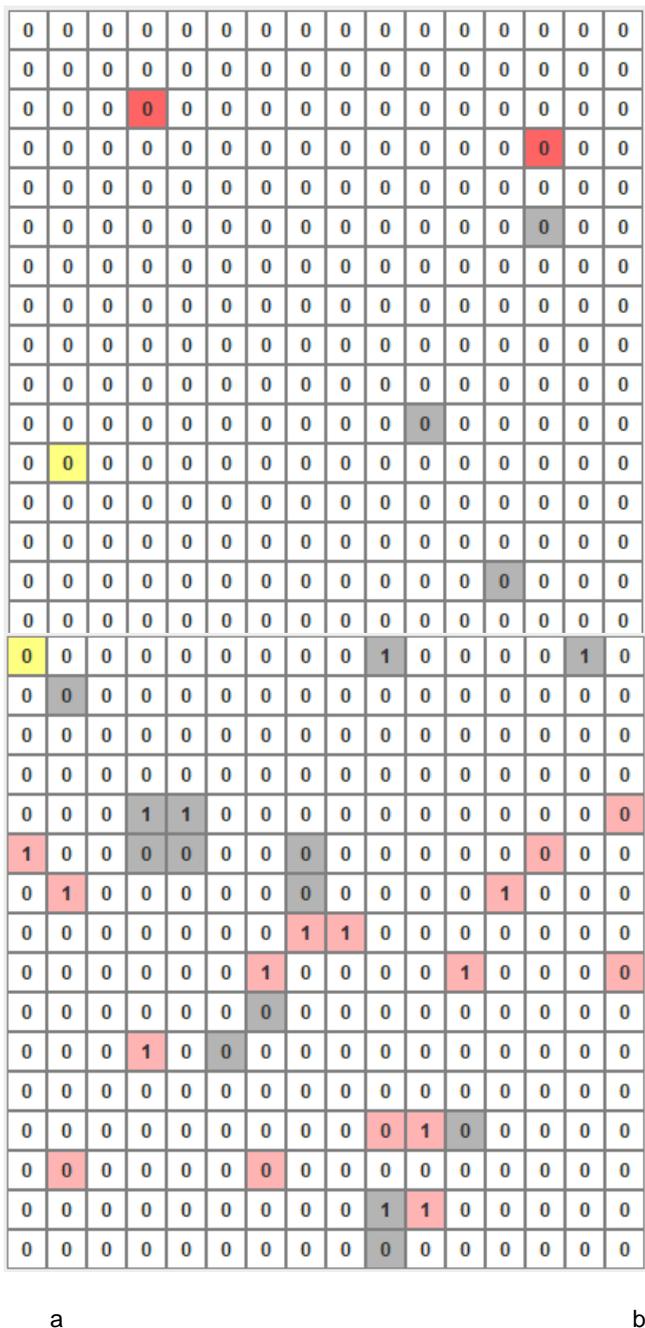


Рисунок 4 – Примеры моделей памяти

Стоит отметить, что данная модель отражает логические адреса ячеек запоминающего устройства, а не физические и не отражает расположение ячеек на кристалле микросхемы памяти. Поэтому ячейки, влияющие друг на друга могут быть сколь угодно удалены и иметь любые логические адреса [2].

Для тестирования модели памяти используются тесты линейной длины семейства «Марш» (March tests). В качестве последовательностей адресов ячеек памяти для маршевых тестов были реализованы различные псевдослучайные и квазислучайные последовательности (счетчиковая, M-последовательность, Холтона, Корпуга, Грея) [3-4].

Процесс тестирования отображается в реальном времени с помощью графического интерфейса. При этом разными цветами отображаются ячейки с обнаруженными и необнаруженными неисправностями.

В разработанном программном средстве реализована функция сохранения модели памяти в файл и загрузки из файла. При этом происходит сериализация и десериализация объекта памяти, с сохранением состояния ячеек.

В результате было разработано программное средство, позволяющее моделировать память запоминающего устройства. ПС позволяет генерировать различные виды связанных неисправностей различными способами. Также позволяет выполнять тестирование этой модели при помощи маршевых тестов с использованием различных адресных последовательностей. Позволяет визуально оценивать эффективность работы тестирования.

Список использованных источников:

1. Функциональное тестирование микросхем ОЗУ // Радиоавтоматика URL: https://radioautomatic.ru/news_and_articles/article/137/ (дата обращения: 20.03.2021).
2. Ярмолик, В. Н. Контроль и диагностика вычислительных систем: [монография] / В. Н. Ярмолик. – Минск: Бестпринт, 2019. – 387 с: ил. 75.
3. Петровская, В. В. Программное средство для анализа маршевых тестов / Петровская В. В., Деменковец Д. В. // Компьютерные системы и сети: сборник статей 58-й научной конференции аспирантов, магистрантов и студентов, Минск, 18–22 апреля 2022 г. / Белорусский государственный университет информатики и радиоэлектроники. – Минск, 2022. – С. 59–61.
4. Деменковец, Д. В. Программное средство моделирования и поиска неисправностей запоминающих устройств / Деменковец Д. В. // Компьютерные системы и сети: сборник тезисов докладов 56-й научной конференции аспирантов, магистрантов и студентов, Минск, апрель-май 2020 года / Белорусский государственный университет информатики и радиоэлектроники. - Минск: БГУИР, 2020. - С. 58-60

UDC 004.33.54

SOFTWARE TOOL FOR SIMULATION OF LINKED FAULTS BASED ON COUPLING FAULTS

Tyushev T.A., Demenkovets D.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Levantsevich V.A. – senior lecturer

Annotation. This paper discusses the architecture of the memory faults simulator focused on coupling faults of three or more memory cells. Memory faults model and notation for their description are described. The algorithm for generating faults and inserting them to the memory model is presented. A software that simulates the process of testing the memory model using march tests is described.

Keywords. RAM, storage devices, RAM testing, march testing, coupling fault, software.

ПРИМЕНЕНИЕ ТЕСТА КАСИСКИ ПРИ ПОТОКОВОМ ШИФРОВАНИИ

Шлык П. А., Болтак С. В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Болтак С. В. – ассистент

В работе описываются условия применимости теста Касиски для криптоанализа шифротекстов, полученных как результат потокового шифрования. Потоковое шифрование реализуется с использованием регистра сдвига с линейной обратной связью.

Функцией потокового шифрования определяется такой вид преобразования, результат которого зависит не только от значения используемого ключа шифрования, но и положения единицы шифрования в исходном тексте [1]. Используемой функцией потокового шифрования является функция, основанная на работе регистра сдвига с линейной обратной связью. Регистр сдвига с линейной обратной связью является конечным автоматом с конкретным количеством возможных состояний. Уникальное количество состояний регистра определяет размерность ключа. Элементом шифрования является бит информации. Предполагается, что исходным текстом для потокового шифрования является осмысленная последовательность символов естественного языка. Рассматриваемый язык задаётся соответствующим алфавитом. Так как максимальная размерность алфавита существующих языков не превышает количества возможных значений одного байта [2], то каждый символ исходного текста представлен в однобайтовой кодировке (например, КОИ-8).

Процесс расшифровки можно разделить на два этапа:

1. определение размерности ключа;
2. определение значения самого ключа.

Для определения размерности ключа предполагается использование частной реализации теста Касиски. Тест основан на следующих допущениях:

1. Ключ для шифрования имеет ограниченную размерность. Следовательно, использование ключа для заданного исходного текста носит периодический характер.

2. Периодическое применение ключа на различные фрагменты текста может привести к тому, что конкретные последовательности символов в исходном тексте будут отображены на соответствующие символы в шифротексте.

3. Размерность ключа достаточно мала по сравнению с размерами шифротекста, что позволяет установить периодический, неслучайный характер появления одинаковых последовательностей.

Значение самого ключа определяется с помощью перебора всех значений ключа заданного размера и поиска такого значения, при котором текст можно считать расшифрованным.

Вероятность появления случайной последовательности байтов, не соответствующей ни одной последовательности символов исходного текста, есть величина меньшая, чем при анализе соответствующих текстов, зашифрованных простым полиалфавитным шифром (например, шифром Вижинера). Это объясняется тем, что функция потокового шифрования переводит значение из пространства значений исходного текста, задающегося частичным, ограниченным диапазоном байта, в пространство значений шифротекста, задающегося всеми возможными значениями байта.

Неслучайный характер отображения последовательностей исходного текста на соответствующие последовательности шифротекста определяется следующим:

1. Если размерность ключа кратна 8, то результат функции потокового шифрования аналогичен результату применения полиалфавитного шифра. Но в данном случае тест Касиски даёт ответ, который необходимо перевести в битовую форму.

2. Если размерность ключа не кратна 8, то полученные значения расстояний между L-граммами будут находиться в линейной зависимости от размерности ключа (то есть, возможны значения, которые не будут определены как L-граммы вообще). По эмпирическим наблюдениям, результат теста Касиски будет кратным или равным истинному значению длины ключа шифрования.

Таким образом, реализация теста Касиски исходит из того, что:

1. количество L-грамм определённого размера мало;
2. количество случайных L-грамм мало.

Предлагается разделить значения расстояний между L-граммами одного размера на сильный пул и слабый пул значений. В сильном пуле поддерживается следующий инвариант: наибольший общий делитель значений отличен от 1. Новое значение добавляется в пул, если изменение общего делителя пула не превысит установленной величины. Иначе значение добавляется в слабый пул. В слабом пуле ведётся создание группы чисел, конкурирующей со значениями сильного пула. Если размер конкурирующей группы превысит размер сильного пула, то сильный пул будет заменён

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

конкурирующей группой. Ответом частной реализации теста Касиски является наибольший общий делитель среди всех общих делителей сильных пулов L-грамм конкретного размера.

Список использованных источников:

1. Ярмолик, В. Н. Элементы теории информации: Практикум / В. Н. Ярмолик, А. П. Занкович, С. С. Портянко. — Минск: БГУИР, 2007. — 39 с.
2. Guinness World Records — [Электронный ресурс]. — Режим доступа: <https://www.guinnessworldrecords.com/world-records/longest-alphabet> — Дата доступа: 31.03.2023.

ПРОГРАММНОЕ СРЕДСТВО УПРАВЛЕНИЯ ВЗАИМОДЕЙСТВИЕМ С КЛИЕНТАМИ МЕДИЦИНСКОГО УЧРЕЖДЕНИЯ

Щемелев М.Ю.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Куликов С.С. – к.т.н., доцент

Следить за своим здоровьем – обязанность каждого человека. Однако выполнению этой обязанности часто препятствуют многие факторы: трудности записи на приём к некоторым специалистам, множество бумажной волокиты, плохо организованные отдельные рабочие процессы в медицинских учреждениях, элементарная человеческая забывчивость [1].

Основной задачей работы является создание приложения, которое бы упростило жизнь как пациенту учреждению здравоохранения, так и медицинскому персоналу.

Для наиболее эффективного и комфортного ведения медицинской ухода за своим здоровьем, пациенту необходимо как минимум:

- 1) Иметь возможность быстро и просто записаться на приём к нужному врачу в удобное для себя время;
- 2) Иметь возможность просматривать результаты медицинских приёмов и обследований даже через долгое время;
- 3) Видеть все предстоящие посещения докторов, приёмы лекарств, терапии доступными в одном месте и получать напоминания накануне.

В свою очередь любому доктору нужно:

- 1) Видеть информацию о пациенте, находящемся на приёме, а именно историю его болезни, актуальные параметры физического состояния и так далее;
- 2) Возможность ставить диагнозы, выписывать лекарства;
- 3) Фиксировать процесс медицинского приёма;
- 4) Возможность выдавать направления на приём к другим врачам и на обследования.

Каждому пользователю, будь то пациент, врач, или администратор, будет предоставляться доступ к веб-приложению и аккаунту текущей медицинской организации. Помимо стандартного функционала средства ведения учёта медицинских приёмов, пользователям, в зависимости от настроек организации, могут быть доступны такие функции, как самостоятельная отправка заявки на приём лично пациентом (без привлечения администратора организации), просмотр возможных диагнозов и проблем со здоровьем, сгенерированных исходя из общего физического состояния, врачом учреждения, увеличения максимального числа пациентов и докторов администратором и другие.

Помимо очевидного удобства использования, приложение так же будет помогать пациентам и врачам не держать в голове все предстоящие и прошедшие события. После медицинского приёма все прописанные лекарства будут автоматически добавляться в календарь событий пациента и напоминать о себе накануне необходимости приёма. Такая система поможет человеку свести к минимуму ситуации нарушения критически важной части любого лечения – периодичности приёма прописанных лекарств. Все предстоящие и прошедшие посещения врачей будут отображаться в календаре и медицинской истории соответственно, благодаря чему человек сможет не забыть о предстоящей записи к врачу и вспомнить прописанное лечение даже спустя годы. Календарь врача должен содержать как информацию о ближайших пациентах, так и разные рабочие встречи, консилиумы, собрания и так далее.

Также нельзя оставлять без внимания такой фактор, как качество работы врача и его отношение к пациенту – для отслеживания динамики роста или падения этих показателей должна быть введена система оценок и отзывов. Публичность таких данных поможет пациентам принять решение о профессионализме и человеческих качествах доктора, а администрации медицинского учреждения – о необходимости поощрить или наказать сотрудника.

Такая же система оценок и отзывов должна быть введена и для пациентов – при систематических опозданиях на приём, несоблюдении назначенного лечения доктор в праве оставить отрицательный отзыв о пациенте, который так же будет виден другим врачам. И наоборот, при хороших оценках и положительных отзывах администрация учреждения может принять решения об участии пациента в дисконтной программе.

Таким образом, программное средство сможет облегчить безмерно тяжёлый процесс здравоохранения как для пациентов, так и для врачей учреждений.

Список использованных источников:

1. S. S. Cutler et al., "Overcoming barriers to effective blood pressure control in patients with diabetes: a systematic review of intervention trials," *American Journal of Medicine*, vol. 123, no. 6, pp. 541-554, Jun. 2010. DOI: 10.1016/j.amjmed.2009.11.019

**СЕКЦИЯ 4
«Информатика»**

УДК

НЕЙРОСЕТЕВОЙ ПОДХОД К ПРОГНОЗИРОВАНИЮ СПОРТИВНЫХ ТЕННИСНЫХ ДАННЫХ

Харкевич А.П., студент гр.953504

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Рыкова О. В. – канд. физ.-мат. наук

Аннотация. В данной работе рассмотрены основные этапы получения готовой нейросети для прогнозирования спортивных теннисных данных, анализа качества обучения и последующей оптимизации модели.

Ключевые слова. Python, pandas, xgboost, нейросеть, прогнозирование результатов, теннис.

Спортивное прогнозирование – процесс предсказания, необходимый для планирования в процессе спортивной подготовки и работы в сфере спорта.

Планирование в спорте на основе прогнозов используется различными специалистами в сфере физической культуры и спорта, тренерскими штабами, селекционерами, «скаутами» команд, аналитическими и букмекерскими агентствами.

Прогнозирование в спорте различается по срокам и может быть краткосрочным, среднесрочным, долгосрочным и сверхдолгосрочным. Вид и результат прогноза в спорте может быть различным и в зависимости, от цели, таким результатом может быть как определенный численный результат, процент вероятности исхода того или иного состязания, так и указание наступления конкретного события во временных рамках состязания. Однако, вне зависимости от вида прогноза и ожидаемого результата, спортивное прогнозирование требует анализа набора факторов, на него влияющих. Компьютерные технологии и современные методы искусственного интеллекта позволяют производить такой анализ факторов и осуществлять прогнозирование, получая при этом результат.

Получение готовой нейросети для прогнозирования спортивных данных будет состоять из нескольких этапов:

- 1) Нахождение данных с матчами и спортсменами
- 2) Очистка данных
- 3) Построение модели нейронной сети
- 4) Их обучение на части данных
- 5) Валидация моделей на другой части данных и на реальных данных
- 6) Оптимизация модели

Рассмотрим подробнее каждый из этих этапов.

На первом этапе необходимо произвести поиск нужно нам датасета.

После проведенного поиска было обнаружено, что для текущей задачи хорошо подойдет github репозиторий с названием «tennis_atp» за авторством пользователя JeffSackmann.

Данный репозиторий содержит огромное количество данных о теннисных матчах более чем за 50 лет. Все данные в нём хранятся в формате csv файлов.

Однако с точки зрения нашей задачи в данном репозитории есть существенный недостаток – данные в нём разбиты по годам и по турнирам. С точки зрения нашей задачи хотелось бы иметь один файл, в котором все данные по годам и турнирам будут объединены. Очевидное решение данной проблемы – загрузить данные вручную или средствами библиотеки какого-либо из языков программирования.

Первый вариант был отвергнут в связи со своей неудобностью и длительностью ручных операций. Однако стоит отметить, что для более узконаправленной задачи он может являться более гибким.

При рассмотрении второго варианта был найден github репозиторий с названием «deuce» за авторством пользователя skoval.

Данный репозиторий содержит пакет, написанный на языке программирования R. Данный пакет парсит данные из репозитория Джеффа Сакмана преобразуя их в единый объект.

Для скачивания пакета была использована библиотека devtools, затем данные были загружены в код и преобразованы в csv файл.

На этом первый этап получения готовой нейросети для прогнозирования спортивных данных был завершен.

На втором этапе был написан скрипт на питоне с использованием библиотеки pandas.

Pandas — программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Pandas предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. Название библиотеки происходит от эконометрического термина «панельные данные», используемого для описания многомерных структурированных наборов информации.

Основная область применения — обеспечение работы в рамках среды Python не только для сбора и очистки данных, но и для задач анализа и моделирования данных, без переключения на более специфичные для статобработки языки (такие, как R и Octave).

Также активно ведётся работа по реализации в pandas «родных» категориальных типов данных.

Pandas прежде всего предназначен для очистки и первичной оценки данных по общим показателям, например среднему значению, квантилям и так далее; статистическим пакетом он в полном смысле не является, однако наборы данных типов DataFrame и Series применяются в качестве входных в большинстве модулей анализа данных и машинного обучения (SciPy, Scikit-Learn и других).

Основные возможности библиотеки pandas:

- 1) Объект DataFrame для манипулирования индексированными массивами двумерных данных
- 2) Инструменты для обмена данными между структурами в памяти и файлами различных форматов
- 3) Встроенные средства совмещения данных и способы обработки отсутствующей информации
- 4) Переформатирование наборов данных, в том числе создание сводных таблиц
- 5) Срез данных по значениям индекса, расширенные возможности индексирования, выборка из больших наборов данных
- 6) Вставка и удаление столбцов данных
- 7) Возможности группировки позволяют выполнять трёхэтапные операции типа «разделение, изменение, объединение» (англ. *split-apply-combine*).
- 8) Слияние и объединение наборов данных
- 9) Иерархическое индексирование позволяет работать с данными высокой размерности в структурах меньшей размерности
- 10) Работа с временными рядами: формирование временных периодов и изменение интервалов и так далее

Библиотека оптимизирована для высокой производительности, наиболее важные части кода написаны на Cython и Си.

Т. к. количество турниров достаточно большое и многие турниры уже не актуальны, т.е. на игроков, участвующих в них, данные прогнозироваться уже не будут, первым шагом будет подвыборка нужных нам турниров. Стоит задать временной диапазон порядка 20 лет и рассматривать турниры только в этом временном диапазоне. Иначе мы будем работать с устаревшими данными, не интересными нам для прогнозирования будущих матчей, однако стоит отметить, что для другой задачи эти данные могли бы сыграть определенную полезную роль.

Также, чтобы сделать прогнозы модели более точными, мы ограничим множество матчей матчами, которые происходили на кортах с жестким покрытием. Так как от типа покрытия сильно разнится стиль игры, то на разных кортах разные игроки будут показывать разные результаты. Мы же возьмём самый распространенный тип, чтобы не вносить лишние помехи.

Следующим шагом мы переименуем названия колонок на более понятные, чтобы с ним было удобнее работать и их названия были более осмыслены.

Далее для удобства мы преобразуем год и дату старта турнира к формату даты в питоне.

Затем с помощью регулярных выражений мы запоминаем сколько выигранных игр было у выигравшего и проигравшего игрока во всех сетах, а также общее количество игр во всех играх.

Затем запоминаем число игр, где подавал только один человек (победитель или побежденный). Запоминаем также число раз, когда в игре создавался брейк-поинт. Также запоминаем число раз, когда брейк-поинт был выигран одним из игроков. Мы считаем также число удачных вторых подач игрока. Считаем мы также и такие метрики, как сколько подач было выиграно игроком и сколько очков он выиграл, когда отбивался. Считаем также общее число выигранных очков для каждого игрока и просто общее число очков в матче.

Убираем лишние колонки данных, которые не будут нам полезны. Колонки, где данные были потеряны и не заполнены, мы заполняем дефолтными или средними значениями.

Следующим шагом будет преобразование данных к нужному формату, чтобы для конкретного матча у нас была статистика игроков как для победителя, так и для проигравшего, в отдельных строках.

Затем нам нужно будет преобразовать необработанную статистику матчей игроков из абсолютных значений в относительные соотношения. Это важно, так как абсолютное значение статистики игрока зависит от длительности матча.

Например, предположим, что Федерер выиграет Кирьоса со счётом 6-4, 7-5. Всего Федерер выиграет 13 партий. Если Джокович выиграет партию у Нишикори со счётом 7-6, 3-6, 5-7, 6-2, 7-6, то он выиграет в общей сложности 28 партий, более чем в два раза больше, чем Федерер. Это несправедливое сравнение. Больше смысла имеет сравнивать их коэффициенты выигрыша в игре. Для Федерера коэффициент выигрыша в игре $(6+7)/(6+4+7+5) = 0.59$, для Джоковича коэффициент выигрыша в игре 0.51. Сравнение этих двух коэффициентов более разумно, чем использование итоговых значений.

Далее мы создадим новые метрики, по которым удобно будет обучать нашу модель.

Первой такой метрикой станет процент выигранных подач. Она рассчитывается следующим образом: число выигранных первых подач плюс число выигранных вторых подач, деленное на общее число первых подач игрока плюс общее число вторых подач игрока плюс общее число двойных ошибок игрока.

Вторая метрика — это процент выигранных очков, когда игрок отбивался: число выигранных очков, когда игрок отбивался, деленная на общее число очков, когда игрок отбивался.

Третьей и четвертой метриками являются среднее число брэйкпоинтов за игру и процент реализации брэйкпоинта.

Далее пятой и шестой метриками будут процент выигранных игр и процент выигранных очков.

Затем мы подсчитаем логарифм ранга игрока и с помощью экспоненциальной функции от ранга его противника считаем вес его победы.

Затем мы подсчитаем взвешенный процент выигранных игр и взвешенный процент выигранных очков в матче.

Ещё одной важной метрикой является так называемый «клатч-фактор».

Клатч – напряженный момент в самом конце игры, от которого зависит исход матча. Умение игрока выложиться на полную является важной метрикой.

Клатч-фактор мы будем рассчитывать, как разницу между процентов выигранных игр и процентом выигранных очков.

Следующим важным шагом будет суммирование статистики по данному игроку за предыдущие 10 матчей.

На рисунке 1 мы можем увидеть, как будет выглядеть часть результатов для Роджера Федерера:

tournament_date_index	player_name	player_serve_win_ratio	player_return_win_ratio	player_bp_per_game
2005-01-17	Roger Federer	0.700709	0.408396	0.732828
2006-01-16	Roger Federer	0.704199	0.433687	0.734755
2007-01-15	Roger Federer	0.727015	0.428965	0.721652
2008-01-14	Roger Federer	0.738650	0.407721	0.744655

Рисунок 1 – Часть результатов для Роджера Федерера

Эти показатели должны быть объединены с данными матча. Ключами, которые будут однозначно идентифицировать нашу группу данных, будут дата турнира и имя игрока как для player_1, так и для player_2. Пример данных после объединения с данными матча мы можем увидеть на рисунке 2.

player_1	player_2	tourney_start_date	player_serve_win_ratio_p1	player_serve_win_ratio_p2
Roger Federer	Marcos Baghdatis	2005-01-17	0.700709	0.634170
Roger Federer	Andre Agassi	2005-01-17	0.700709	0.692173
Roger Federer	Denis Istomin	2006-01-16	0.704199	0.660193
Roger Federer	Florian Mayer	2006-01-16	0.704199	0.614321
Roger Federer	Max Mirnyi	2006-01-16	0.704199	0.682761

Рисунок 2 – Пример данных после объединения с данными матча

Мы также возьмем различия между агрегатами `player_1` и `player_2`, чтобы уменьшить количество функций и, следовательно, наше время вычисления. Интуитивно это работает, потому что, выиграет ли Федерер матч, зависит от того, насколько хорош его соперник по сравнению с ним. Пример данных после взятия разницы между агрегатами можно увидеть на рисунке 3.

<code>player_1</code>	<code>player_2</code>	<code>tourney_start_date</code>	<code>player_serve_win_ratio_diff</code>
Roger Federer	Marcos Baghdatis	2005-01-17	0.066539
Roger Federer	Andre Agassi	2005-01-17	0.008536
Roger Federer	Denis Istomin	2006-01-16	0.044006
Roger Federer	Florian Mayer	2006-01-16	0.089877
Roger Federer	Max Mirnyi	2006-01-16	0.021438

Рисунок 3 – Пример данных после взятия разницы между агрегатами

В разрезе последних 10 матчей для ранга игрока и логарифмического ранга игрока мы возьмем его наиболее повторяющееся значение по матчам. Для оставшихся метрик мы считаем среднее значение скользящим окном по матчам и берем наиболее часто повторяющееся значение среднего.

Далее мы добавляем полученные новые метрики к исходным данным игроков.

И затем считаем разницу в метриках двух соперников в матче.

Таким образом функции, которые мы будем использовать для наших прогнозов, будут представлять собой разницу в средней статистике каждого игрока за предыдущие 10 матчей. Например, если мы рассматриваем матч Александра Зверева как `player_1` и Стефаноса Циципаса как `player_2`, то для Александра Зверева мы хотим усреднить его статистику (например, процент выигранных партий) по последним 10 матчам, допустим, это число 0.63. То же самое мы сделаем и для Стефаноса Циципаса, допустим, его средний коэффициент выигрыша в матчах - 0.68. Отметим, что это усредненное значение по матчам, в которых каждый игрок участвовал индивидуально, а не по их общим предыдущим поединкам.

Возьмем разницу между двумя характеристиками игроков, $0.63 - 0.68 = -0.05$ и используем ее как характеристику для прогнозирования того, выиграет ли `player_1` (Александр Зверев). Мы можем сделать это для множества других статистических данных игроков, таких как ранг игрока, процент выигранных первой и второй подачи или процент выигранных ответных очков.

Для генерации предсказаний была использована модель для классификации `XGBClassifier` на основе `XGBOOST`.

`XGBoost` — алгоритм машинного обучения, основанный на дереве поиска решений и использующий фреймворк градиентного бустинга. В задачах предсказания, которые используют неструктурированные данные (например, изображения или текст), искусственная нейронная сеть превосходит все остальные алгоритмы или фреймворки. Но когда дело доходит до структурированных или табличных данных небольших размеров, в первую очередь оказываются алгоритмы, основанные на дереве поиска решений.

`XGBoost` разрабатывался как исследовательский проект Вашингтонского Университета. Tianqi Chen и Carlos Guestrin представили их работу на конференции SIGKDD в 2016 году и произвели фурор в мире машинного обучения. С момента его введения этот алгоритм не только лидировал в соревнованиях Kaggle, но и был основой нескольких отраслевых передовых приложений. В результате образовалось общество специалистов по анализу данных, вносящих вклад в проекты `XGBoost` с открытым исходным кодом с ~350 участниками и ~3,600 коммитами на GitHub.

Особенности фреймворка:

- 1) Широкая область применения: может быть использован для решения задач регрессии, классификации, упорядочения и пользовательских задач на предсказание.
- 2) Совместимость: Windows, Linux и OS X.
- 3) Языки: поддерживает большинство ведущих языков программирования, например, C++, Python, R, Java, Scala и Julia.
- 4) Облачная интеграция: поддерживает кластеры AWS, Azure и Yarn, хорошо работает с Flink, Spark

Дерево принятия решений — простой в визуализации и достаточно понятный алгоритм. Однако не так уж просто понять следующее поколение алгоритмов, основывающихся на деревьях. Поэтому для понимания обратимся к несложной аналогии.

Представьте, что вы специалист по подбору персонала и собеседуете нескольких отличных кандидатов. Каждый шаг эволюции алгоритмов, основанных на деревьях, может быть представлен как версия хода собеседования.

- 1) Дерево принятия решений: Каждый специалист по подбору персонала при собеседовании кандидата ориентируется по своему списку критериев: образование, опыт работы, успешность прохождения собеседования.
- 2) Бэггинг: Представьте, что вместо одного специалиста по подбору персонала теперь за каждым кандидатом наблюдают несколько, и каждый имеет возможность проголосовать. Этот алгоритм при принятии окончательного решения учитывает все высказанные мнения.
- 3) Случайный лес: Этот алгоритм основан на бэггинге. Отличается он тем, что выбирает случайные признаки. То есть, каждый специалист по подбору персонала может проверить знания кандидата лишь в какой-то одной случайно выбранной области.
- 4) Бустинг: Это альтернативный подход, в котором каждый специалист по подбору персонала основывается на оценке кандидата предыдущим специалистом. Это ускоряет процесс собеседования, так как не подходящие кандидаты сразу же отсеиваются.
- 5) Градиентный бустинг: Частный случай бустинга, в котором ошибка минимизируется алгоритмом градиентного спуска. То есть, наименее квалифицированные кандидаты отсеиваются как можно раньше.
- 6) XGBoost: Экстремальный градиентный бустинг. Это идеальная комбинация оптимизации ПО и железа для получения точных результатов за короткое время с минимальным использованием вычислительных ресурсов.

XGBoost и Gradient Boosting Machines (GBM) — ансамбли методов деревьев, которые используют принцип бустинга (чаще всего, алгоритм построения бинарного дерева решений) при помощи архитектуры градиентного спуска. В свою очередь, XGBoost — улучшение фреймворка GBM через системную оптимизацию и усовершенствование алгоритма.

Системная оптимизация:

- 1) Параллелизация: В XGBoost построение деревьев основано на параллелизации. Это возможно благодаря взаимозаменяемой природе циклов, используемых для построения базы для обучения: внешний цикл перечисляет листья деревьев, внутренний цикл вычисляет признаки. Нахождение цикла внутри другого мешает параллелизовать алгоритм, так как внешний цикл не может начать своё выполнение, если внутренний ещё не закончил свою работу. Поэтому для улучшения времени работы порядок циклов меняется: инициализация проходит при считывании данных, затем выполняется сортировка, использующая параллельные потоки. Эта замена улучшает производительность алгоритма, распределяя вычисления по потокам.
- 2) Отсечение ветвей дерева: В фреймворке GBM критерий остановки для разбиения дерева зависит от критерия отрицательной потери в точке разбиения. XGBoost использует параметр максимальной глубины `max_depth` вместо этого критерия и начинает обратное отсечение. Этот «глубинный» подход значительно улучшает вычислительную производительность.
- 3) Аппаратная оптимизация: Алгоритм был разработан таким образом, чтобы он оптимально использовал аппаратные ресурсы. Это достигается путём создания внутренних буферов в каждом потоке для хранения статистики градиента. Дальнейшие улучшения, как, например, вычисления вне ядра, позволяют работать с большими наборами данных, которые не помещаются в памяти компьютера.

Улучшения алгоритма:

- 1) Регуляризация: Он штрафует сложные модели, используя как регуляризацию LASSO (L1), так и Ridge-регуляризацию (L2) для того, чтобы избежать переобучения.
- 2) Работа с разреженными данными: Алгоритм упрощает работу с разреженными данными, в процессе обучения заполняя пропущенные значения в зависимости от значения потерь. К тому же он позволяет работать с различными уровнями разреженности.
- 3) Метод взвешенных квантилей: XGBoost использует его для того, чтобы наиболее эффективно находить оптимальные точки разделения в случае работы со взвешенным датасетом.
- 4) Кросс-валидация: Алгоритм использует свой собственный метод кросс-валидации на каждой итерации. То есть, нам не нужно отдельно программировать этот поиск и определять количество итераций бустинга для каждого запуска.

XGBoost может похвастаться лучшей комбинацией «производительность-время обучения» среди других алгоритмов.

Остается лишь грамотно настроить алгоритм путём подбора гиперпараметров.

В нашем случае мы используем логистическую регрессию для бинарной классификации и на выходе получаем вероятность победы одного из игроков.

Для обучения используем 300 деревьев. Для управления взвешиванием новых деревьев, добавленных в модель, используется параметр `learning_rate`, мы установили его равным 0.02. Параметр максимальной глубины дерева мы установили равным 6.

В качестве метрики оценки для проверки данных была выбрана AUC, что отлично подходит для бинарной классификации.

Чтобы лучше понять, что такое AUC метрика разберемся сначала, что из себя представляет ROC-кривая.

Кривая ROC представляет собой график, показывающий эффективность модели классификации при всех пороговых значениях классификации. Эта кривая отображает два параметра:

- 1) True Positive Rate
- 2) False Positive Rate

True Positive Rate (TPR) определяется следующим образом:

$$TPR = TP / (TP + FN) \quad (1)$$

False Positive Rate (FPR) определяется следующим образом:

$$FPR = FP / (FP + TN) \quad (2)$$

Кривая ROC отображает соотношение TPR и FPR при различных порогах классификации. Снижение порога классификации позволяет классифицировать больше элементов как положительные, тем самым увеличивая количество ложных срабатываний и истинных срабатываний. На рисунке 4 показан пример кривой ROC.

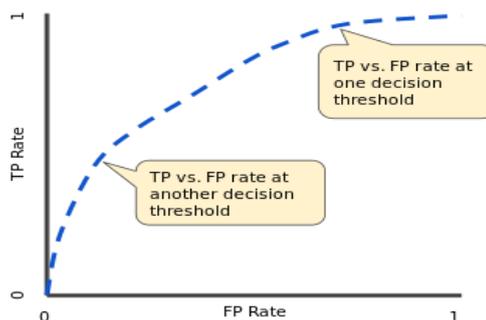


Рисунок 4 – Коэффициент TP и FP при различных порогах классификации

Чтобы вычислить точки на ROC-кривой, мы могли бы много раз оценивать модель логистической регрессии с разными порогами классификации, но это было бы неэффективно. К счастью, существует эффективный алгоритм, основанный на сортировке, который может предоставить нам эту информацию, называемый AUC.

AUC означает «Площадь под кривой ROC». То есть AUC измеряет всю двумерную область под всей кривой ROC (например, интегральное исчисление) от (0,0) до (1,1). Пример можно увидеть на рисунке 5.

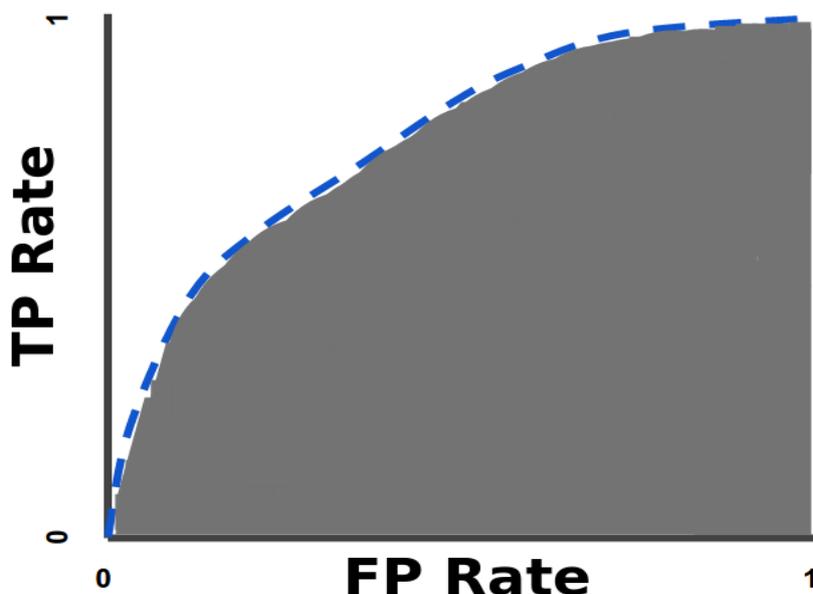


Рисунок 5 – AUC (площадь под кривой ROC)

AUC обеспечивает совокупный показатель производительности по всем возможным пороговым значениям классификации. Один из способов интерпретации AUC — это вероятность того, что модель ранжирует положительный прогноз выше, чем отрицательный прогноз.

Значение AUC варьируется от 0 до 1. Модель, чьи прогнозы на 100 % неверны, имеет значение AUC, равное 0,0; тот, чьи предсказания верны на 100%, имеет AUC 1,0.

AUC желательно использовать по следующим двум причинам:

- 1) AUC не зависит от масштаба. Он измеряет, насколько хорошо ранжируются прогнозы, а не их абсолютные значения.
- 2) AUC не зависит от порога классификации. Он измеряет качество прогнозов модели независимо от выбранного порога классификации.

Однако обе эти причины сопровождаются оговорками, которые могут ограничивать полезность AUC в определенных случаях использования:

- 1) Масштабная инвариантность не всегда желательна. Например, иногда нам действительно нужны хорошо откалиброванные по вероятности выходные данные, и AUC не скажет нам об этом.
- 2) Инвариантность порога классификации не всегда желательна. В тех случаях, когда существуют большие различия в стоимости ложноотрицательных и ложноположительных результатов, может быть важно минимизировать один тип ошибки классификации. Например, при обнаружении спама в электронной почте вы, вероятно, захотите свести к минимуму ложноположительные срабатывания (даже если это приведет к значительному увеличению ложноотрицательных результатов). AUC не является полезным показателем для этого типа оптимизации.

Обучение нашей модели досрочно заканчивается, если в течение 20 раундов обучения не было обнаружено улучшения метрики.

После выбора модели и значений гиперпараметров нужно будет разделить данные на тренировочные и данные для тестирования (валидации) модели, чтобы предотвратить перегрузку xgboost.

Разбивать данные мы будем следующим образом: первые взятые 20 лет будут использоваться для тренировки, 20 + 1 год будет использоваться для валидации и 20 + 2 год будет использоваться для тестирования.

Подключив наши тренировочные и валидационные сетки данных в модель xgboost, мы получаем окончательную валидацию AUC со значением в 0,78, которое можно увидеть на рисунке 6.

```
[164] validation_0-auc:0.78196
```

Рисунок 6 – Значения метрики AUC

Для анализа вклада функций в предсказательную мощьность мы можем использовать встроенный метод `feature_importances_`. Он, по сути, вычисляет долю времени, в течение которого функция появляется в дереве решений. Чем больше она появляется, тем больше вероятность того, что она будет сильным фактором точности прогнозирования. Значение анализа приведены на рисунке 7.

```
pd.Series(model.feature_importances_, index=X_train.columns).sort_values(ascending=False)

player_log_rank_diff          0.619010
player_game_win_ratio_diff    0.108943
player_point_win_ratio_weighted_diff  0.080545
player_serve_win_ratio_diff   0.075001
player_rank_diff              0.060340
player_return_win_ratio_diff   0.056161
dtype: float32
```

Рисунок 7 – Значения метрик модели при анализе с помощью `feature_importances`

Неудивительно, что самой существенной характеристикой, определяемой `xgboost`, является разница логарифмов рангов игроков.

Это также подтверждается ещё одним фактором, а именно важностью перестановки.

Рассмотрим детальнее вопрос о том, что представляет собой важность перестановки.

По сравнению с большинством других подходов важность перестановки:

- 1) можно быстро рассчитать,
- 2) она широко используется
- 3) соответствует свойствам, которые мы хотели бы иметь для меры важности признаков.

Важность перестановки использует модель не самым обычным способом, многие сначала находят это запутанным. Для понимания мы начнем с примера. Рассмотрим данные в следующем формате на рисунке 8:

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24

Рисунок 8 – Данные о человеке, доступные в возрасте, когда ему было 10 лет.

Мы хотим предсказать рост человека, когда ему исполнится 20 лет, используя данные, доступные в возрасте, когда человеку было 10 лет.

Наши данные включают полезные функции (рост в возрасте 10 лет), функции с небольшой прогностической силой (владение количеством носков), а также некоторые другие функции, на которых мы не будем акцентировать внимание в этом примере.

Важность перестановки рассчитывается после обучения модели. Таким образом, мы не будем менять модель или прогнозы, которые мы получим для заданного значения роста, количества носков и т. д.

Вместо этого мы зададим следующий вопрос: если я случайным образом перетасую один столбец проверочных данных, оставив целевой столбец и все остальные столбцы на месте, как это повлияет на точность прогнозов в этих уже перемешанных данных?

Случайное изменение порядка одного столбца должно привести к менее точным прогнозам, поскольку полученные данные больше не соответствуют ничему, наблюдаемому в реальном мире. Точность модели особенно страдает, если мы перемешиваем столбец, на который модель сильно опиралась при прогнозировании. В этом случае перетасовка роста в 10 лет вызовет ужасные последствия. Если бы вместо этого мы перетасовали столбец владения носками, полученные в результате прогнозы не пострадали бы так сильно.

Таким образом процесс выглядит следующим образом:

- 1) Получаем обученную модель.
- 2) Перемешиваем значения в одном столбце, делаем прогнозы, используя полученный набор данных. Используем эти прогнозы и истинные целевые значения, чтобы вычислить, насколько функция потерь пострадала от перетасовки. Это ухудшение производительности измеряет важность переменной, которую мы только что перетасовали.
- 3) Возвращаем данные в исходный порядок (отменив перетасовку с шага 2). Теперь повторяем шаг 2 со следующим столбцом в наборе данных, пока не рассчитаем важность каждого столбца.

Рассмотрим важность перестановки для метрик нашей модели на рисунке 9:

```
perm = PermutationImportance(model).fit(X_val, y_val)
eli5.show_weights(perm, feature_names = X_val.columns.tolist())
```

Weight	Feature
0.1689 ± 0.0195	player_log_rank_diff
0.0132 ± 0.0132	player_rank_diff
0.0098 ± 0.0062	player_game_win_ratio_diff
0.0039 ± 0.0100	player_point_win_ratio_weighted_diff
0.0039 ± 0.0050	player_return_win_ratio_diff
-0.0065 ± 0.0051	player_serve_win_ratio_diff

Рисунок 9 – Значения важности перестановки для метрик модели

Значения сверху являются наиболее важными характеристиками, а значения внизу имеют наименьшее значение.

Первое число в каждой строке показывает, насколько снизилась производительность модели при случайном перемешивании (в данном случае с использованием «точности» в качестве показателя производительности).

Иногда можно увидеть отрицательные значения важности перестановок. В этих случаях предсказания перетасованных (или зашумленных) данных оказывались более точными, чем реальные данные. Это происходит, когда случайность делает прогнозы на перетасованных данных более точными. Это чаще встречается с небольшими наборами данных, такими как в этом примере, потому что здесь больше места для удачи/случайности.

Важность перестановки, по сути, заключается в том, что она включает в себя перестановку в случайном порядке очередности измерений, а также в том, как она влияет на точность прогнозирования. Если точность резко снижается, то это хороший индикатор того, что данная функция была действительно важна, если она не сильно меняется, то эта функция, вероятно, не важна для вашей модели.

Таким образом проанализировав с помощью различных способов вклад метрик в обучение приходим к выводу, что наиболее влиятельной метрикой оказалась разница логарифмов рангов игроков.

Учитывая, что большая часть инфраструктуры уже заложена, делать прогнозы теперь относительно просто.

К примеру, для практического использования мы можем использовать список игроков, с помощью `itertools` создать множество всех перестановок между парами игроков и для каждого из них вычислить результат их матча. Таким образом каждый раз не нужно будет вычислять один и тот же результат.

При прогнозах особое внимание стоит уделять игрокам, которые не имели до этого большого количества серьезных матчей, по которым есть данные. Такие игроки вносят непредсказуемость, т. к. не всегда понятно, чего от них можно ожидать. В дальнейшем в качестве развития работы можно разработать использование одного из алгоритмов «холодного старта» для таких игроков.

Также можно углубить анализ и посчитать, к примеру, среднее значение вероятностей победы игрока в матчах со всеми его потенциальными соперниками.

Дальнейшим улучшением модели будет служить внедрение `elo` (Эло).

Для начало рассмотрим, что такое Эло.

Эло — это более совершенная рейтинговая система по сравнению с формулами рейтинга, используемыми АТР и WТА.

Принцип любой системы Эло заключается в том, что рейтинг каждого игрока является оценкой его силы, и каждый матч (или турнир) позволяет нам обновлять эту оценку. Если игрок выигрывает, его рейтинг повышается; если она проигрывает, он падает.

Отличие Эло заключается в определении величины, на которую должен увеличиваться или уменьшаться рейтинг. Учитываются две основные переменные: сколько матчей уже есть у игрока в системе (то есть насколько мы уверены в предматчевом рейтинге), и рейтинг соперника.

Если вы задумаетесь об этом на мгновение, то увидите, что эти две переменные являются хорошим приближением того, как мы обычно думаем о силе игрока. Чем больше мы уже знаем об игроке, тем меньше мы изменим свое мнение на основании одного матча. Проигрыш Новака Джоковича по круговой системе Доминику Тиму в Лондоне стал неожиданностью, но только самые скептически настроенные фанаты Джоковича увидели в этом результате катастрофу, которая должна существенно изменить нашу оценку его игровых способностей. Точно так же мы корректируем наше мнение в зависимости от рейтинга оппонента. Поражение от Тима разочаровывает, но поражение, скажем, от Марко Чеккинато вызывает большее беспокойство. Система Эло включает в себя эту естественную интуицию.

Традиционно игроку при входе в систему присваивается рейтинг Эло 1200 — до того, как появятся какие-либо результаты. Это число само по себе совершенно произвольно. Все, что имеет значение, — это разница между рейтингами игроков, поэтому, если мы начнем считать рейтинг каждого участника с 0, 100 или 888, конечный результат этих различий останется прежним.

На данный момент лучшими игроками ATP и WTA являются Рафаэль Надаль и Эшли Барти с 2203 и 2123 очками соответственно. Лучшие игроки часто находятся в этом диапазоне, а лучшие из лучших часто приближаются к 2500. Согласно последней версии алгоритма, пик Джоковича был 2470, а лучший результат Серены Уильямс — 2473.

Отметка в 2000 баллов — это хорошее практическое правило, позволяющее отделить элиту от остальных. На данный момент такие высокие рейтинги имеют шесть мужчин и семь женщин. 16 мужчин и 18 женщин имеют рейтинг Эло не ниже 1900, а рейтинг 1800 примерно эквивалентен месту в топ-50.

Как только мы присвоим каждому игроку единый пиковый рейтинг, вполне естественно начать сравнивать его по эпохам.

Можно сравнить по эпохам, как каждый игрок справлялся с его конкурентами. В 1990 году Хелена Сукова достигла рейтинга 2123 — точно такого же, как сегодня у Барти. Это не значит, что Сукова тогда была так же хороша, как сейчас Барти. Но это означает, что их показатели по сравнению с их сверстниками были одинаковыми. Второй эшелон игроков был значительно слабее тридцать лет назад, поэтому добиться такого рейтинга в каком-то смысле было проще. В то время рейтинг Суковой был хорош только для 11-го места, намного отставая от 2600 Штеффи Граф.

Таким образом, Ело не позволяет вам ранжировать игроков по эпохам, если вы не уверены, что уровень конкуренции был одинаковым, или если у вас нет другого способа решить эту проблему.

Связанный с этим вопрос - инфляция Эло, которая также может усложнить сравнения между эпохами. Каждый раз, когда играется матч, победитель и проигравший эффективно «торгуют» некоторыми из своих очков, поэтому общее количество рейтинговых очков Эло в системе не меняется. Однако каждый раз, когда в систему заходит новый игрок, общее количество очков увеличивается. И всякий раз, когда игрок уходит из игры, общее количество очков уменьшается.

Было бы неплохо, если бы сложения и вычитания компенсировали друг друга, но для многих соревнований, в которых используется Эло, этого не происходит. Добавления, как правило, перевешивают вычитания, поэтому рейтинг Эло со временем увеличивается. Следовательно количество очков в системе меняется со временем, по причинам, не связанным с силой топ-игроков.

Эло дает нам рейтинг для каждого игрока, и мы получаем представление о том, что мы можем и не можем делать с ними.

Одной из основных целей любой рейтинговой системы является прогнозирование исхода матчей, что Эло делает лучше, чем большинство других, включая рейтинги ATP и WTA. Единственным входом, необходимым для составления прогноза, является разница между рейтингами двух игроков, которую затем можно подставить в следующую формулу:

$$1 - (1 / (1 + (10^{((разница)/400)}))) \quad (3)$$

Если бы мы хотели спрогнозировать матч-реванш последнего матча финала Кубка Дэвиса, мы бы взяли рейтинги Эло Надаля и Дениса Шаповалова (2203 и 1947), нашли разницу (256) и подставили ее в формулу, получив результат 81,4% для Надаля на победу. Если бы мы использовали отрицательную разницу (-256), мы бы получили в результате 18,6%, для Шаповалова на победу.

В теннисе рейтинги и прогнозы должны сильно различаться в зависимости от покрытия.

Для каждого игрока нужно вести четыре отдельных рейтинга Эло: общий, только корт с твердым покрытием, только корт с грунтовым покрытием и только корт с травяным покрытием. Например, рейтинг Тима составляет 2066 в целом, 1942 на харде, 2031 на грунте и 1602 на траве.

(Поверхностные рейтинги, как правило, ниже: грунтовый рейтинг Тима занимает третье место, намного опережая всех, кроме Надаля и Джоковича.)

Рейтинги на одной поверхности говорят нам, как бы мы ранжировали игроков, если бы просто отбрасывали результаты на всех остальных поверхностях. Однако это неверно. Односторонние рейтинги не очень хороши для прогнозирования результатов матчей. Лучшим решением будет взять смесь 50/50 одноповерхностных и общих рейтингов. Если бы мы хотели предсказать шансы Тима в матче на грунтовом корте, мы бы использовали половинную смесь его общего рейтинга 2066 и его рейтинга на грунтовом корте 2031.

Однако стоит отметить, что не существует естественного закона, который диктует смесь рейтингов именно 50/50.

Спросите у фанатов тенниса, какие матчи турниров имеют большее значение для рейтингов — и вы получите длинный подробный список факторов, определяющих это значение. Например, финалы или олимпийские игры являются более важными, чем обычные личные встречи. Для таких матчей нужно вносить определенные корректировки.

Эло предусматривает такие корректировки. Коэффициент, обычно называемый «коэффициентом k », позволяет придавать больший вес определенным матчам. Это распространено в рейтингах Эло и для других видов спорта, например, при использовании более высокого коэффициента k для постсезонных игр, чем для игр регулярного сезона.

Обычно этот коэффициент полагают равным 10 для сильнейших игроков (рейтинг 2400 и выше), 20 (было 15) — для игроков с рейтингом меньше, чем 2400 и 40 (было 30) — для новых игроков (первые 30 партий с момента получения рейтинга ФИДЕ), а также для игроков до 18 лет, рейтинг которых ниже 2300.

В нашем случае этот коэффициент будет зависеть от числа выигранных и проигранных матчей игрока.

Когда игроки пропускают значительное количество времени, нужно снижать их рейтинг, а затем повышать коэффициент k на несколько матчей после их возвращения.

Эти шаги являются логическим продолжением системы Эло. Если игрок травмирован на несколько месяцев, мы никогда не знаем, чего ожидать, когда он вернется. Может быть, он так же силен, как и прежде; может быть, он уже на шаг медленнее. Возможно, что он быстро вернется в норму, но также он может и никогда полностью не вернуться в форму. Длительное отсутствие вызывает много вопросов. Игрок с травмой редко возвращается в лучшей форме, чем до ухода, в то время как многие игроки возвращаются хуже, что дает нам средний уровень производительности после травмы, который хуже, чем до её отсутствия.

Следовательно, когда игрок впервые возвращается, наша оценка должна состояться таким образом будто бы он играет немного хуже. Однако некоторым сильным ранним результатам следует придать больший вес — отсюда и более высокий коэффициент k . Коэффициент k отражает тот факт, что сразу после отсутствия игрока мы не так уверены в своей оценке, как обычно.

Алгоритм усложняется, но логика проста. По сути, это просто попытка выработать строгую версию утверждений вроде: «Я не знаю, насколько хорошо он будет играть, когда вернется, но я буду внимательно следить».

Одним из дополнительных преимуществ штрафа за отсутствие является то, что он противодействует естественной склонности Эло к завышению рейтингов. В то время как больше игроков входит в систему, чем выходит из нее, штраф удаляет некоторые очки, не перераспределяя их другим игрокам.

Самый простой способ сравнить рейтинговые системы — это показатель под названием «точность», который подсчитывает правильные прогнозы. В финале Кубка Дэвиса было 50 одиночных матчей, и Эло правильно выбрал победителя в 36 из них с рейтингом точности 72%. Рейтинг АТР правильно выбрал победителя (в том смысле, что игрок с более высоким рейтингом выиграл матч) в 30 из них с рейтингом точности 60%. В этом маленьком эксперименте Эло превзошел официальные рейтинги. Эло был также значительно лучше в течение всего сезона.

Лучшей метрикой для сравнения рейтинговых систем является оценка Брайера, которая учитывает достоверность каждого прогноза. Ранее мы видели, что Эло дает Надалю шанс победить Шаповалова в 81,4%. Если Надаль в конечном итоге выигрывает, 81,4% — это более хороший прогноз, чем, скажем, 65%, но это хуже прогноза 90%. Оценка Брайера представляет собой квадрат расстояния между прогнозом (81,4%) и результатом (0% или 100%, в зависимости от победителя) и усредняет эти числа для всех прогнозируемых матчей. Он вознаграждает близкие к 100 прогнозы, которые оказываются верными, но поскольку он использует квадрат расстояния, он сурово наказывает близкие к 100, но неверные прогнозы.

Более интуитивный способ понять, к чему ведет счет Брайера, — представить, что Надаль и Шаповалов играют 100 матчей подряд. (Или, более точно, но менее интуитивно, представьте, что 100 одинаковых Надалей играют одновременные матчи против 100 одинаковых Шаповаловых.) Прогноз 81,4% означает, что мы ожидаем, что Надаль выигрывает в 81 матче. Если Надаль в итоге

выиграет 90, прогноз не был достаточно благоприятным для него. Мы никогда не получим 100 таких одновременных матчей, но у нас есть тысячи отдельных матчей, многие из которых имеют одинаковые прогнозы, например, 60-процентный шанс на победу фаворита. Оценка Брайера объединяет все эти пары «прогноз-результат» и выдает число, сообщающее нам, какой у нас результат.

Трудно прогнозировать результат отдельных теннисных матчей. Любая система, какой бы сложной она ни была, в большинстве случаев будет ошибаться. Во многих случаях «правильный» прогноз едва ли лучше, чем отсутствие прогноза вообще, если данные свидетельствуют о том, что конкуренты равны. Таким образом, «точность» имеет ограниченное применение — более важно иметь правильную степень уверенности в победе, чем просто выбирать победителей.

Рейтинги Эло имеют гораздо более низкие (лучшие) баллы Брайера, чем прогнозы, полученные на основе рейтингов ATP и WTA.

Оценка Брайера также является мерой, которая говорит нам о том, является ли определенная корректировка, такая как смешивание поверхностей, отсутствие травм или тип турнира, улучшением системы. Штраф за травму снижает оценку Брайера в общем наборе прогнозов Эло, поэтому мы сохраняем эту корректировку. А уменьшение k-фактора для матчей первого раунда не имеет никакого эффекта, поэтому мы не используем эту корректировку.

Теперь зная информацию об оценке Эло мы сможем внедрить её в качестве метрики в нашу модель. Проведем предрасчёт рейтинга Эло для каждого из спортсменов. Затем проведем очистку данных, такую же как проводили для прошлой модели, добавив рейтинг Эло в качестве дополнительного параметра для обучения.

Обучим нашу модель аналогично тому, как делали это до этого и посмотрим на результаты. Проанализировав вклад функций в обучение, получим следующий результат на рисунке 10.

```
pd.Series(model.feature_importances_, index=X_cols_subset_1).sort_values(ascending=False)
```

player_old_elo_diff	0.440281
player_win_weight_diff	0.120508
player_log_rank_diff	0.079821
player_rank_diff	0.079811
player_game_win_ratio_diff	0.078194
player_point_win_ratio_weighted_diff	0.068934
player_return_win_ratio_diff	0.066744
player_serve_win_ratio_diff	0.065708

Рисунок 10 – Значения метрик модели с Эло при анализе с помощью feature_importances

Как мы видим, Эло сыграло очень важную роль в обучении модели в несколько раз обходя остальные метрики.

Рассмотрим теперь важность перестановок.

Weight	Feature
0.1173 ± 0.0320	player_old_elo_diff
0.0528 ± 0.0242	player_win_weight_diff
0.0291 ± 0.0162	player_log_rank_diff
0.0213 ± 0.0118	player_game_win_ratio_diff
0.0157 ± 0.0100	player_rank_diff
-0.0016 ± 0.0094	player_point_win_ratio_weighted_diff
-0.0024 ± 0.0107	player_return_win_ratio_diff
-0.0039 ± 0.0141	player_serve_win_ratio_diff

Рисунок 11 – Значения важности перестановки для метрик модели с Эло

И здесь Эло показывает прекрасный результат, обходя остальные метрики.

Таким образом исходя из анализа метрик можно утверждать, что логарифм ранга игрока оказывает большее влияние на обучение, чем просто ранг игрока. Но Эло в свою очередь обходит как ранг игрока, так и логарифм ранга игрока по важности влияния на обучение.

Возьмём определенный список игроков, между которыми были сыграны матчи, и они известны. Пример части игроков приведен на рисунке 12.

```
'Rafael Nadal',
'Hugo Dellien',
'Federico Delbonis',
'Joao Sousa',
'Christopher Eubanks',
'Peter Gojowczyk',
'Jozef Kovalik',
'Pablo Carreno Busta',
'Nick Kyrgios',
'Lorenzo Sonego',
```

Рисунок 12 – Пример части игроков взятых для валидации

Введем вручную результаты этих матчей. Пример части результатов матчей приведен на рисунке 13.

```
['Rafael Nadal', 'Hugo Dellien', 1],
['Hugo Dellien', 'Rafael Nadal', 0],
['Federico Delbonis', 'Joao Sousa', 1],
['Joao Sousa', 'Federico Delbonis', 0],
['Christopher Eubanks', 'Peter Gojowczyk', 0],
['Peter Gojowczyk', 'Christopher Eubanks', 1],
['Jozef Kovalik', 'Pablo Carreno Busta', 0],
['Pablo Carreno Busta', 'Jozef Kovalik', 1],
```

Рисунок 13 – Пример части результатов матчей взятых для валидации

Таким образом мы имеем список игроков и известные результаты матчей, на которых мы сможем провалидировать качество наших обученных моделей.

Для валидации предсказаний моделей будем использовать следующие метрики accuracy, AUC, logloss.

Метрика AUC была описана выше.

Рассмотрим две другие метрики.

Логарифмическая потеря – метрика оценки эффективности Модели Бинарной классификации.

Для того, чтобы разобраться в том, что это, обратимся к концепции бинарной классификации. Такой алгоритм сначала предсказывает вероятность того, что Наблюдение будет отнесено к классу 1, а затем причисляет его к одному из двух классов (1 или 0) на основе того, пересекла ли вероятность пороговое значение, которое устанавливается по умолчанию равным 0,5. На рисунке 14 приведены результаты классификации email (1 – "спам").

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	ПРЕДСКАЗАННЫЙ КЛАСС
64247	Попробуйте бесплатную защиту от телефонного спама и выиграйте один	1	0,95	1
28591	Привет! Прости пожалуйста что пропал – очень много было работы. В целом, ничего не меняет	0	0,2	0
70282	Здравствуйте, Елена Александровна! Запрошенная вами справка "О наличии счетов"	0	0,65	1

Рисунок 14 – Результаты классификации email

Итак, прежде чем предсказывать класс записи, модель должна спрогнозировать вероятность того, что запись будет отнесена к классу 1. Помните, что именно от этой вероятности предсказания записи данных зависит значение логарифмической потери.

Логарифмическая потеря указывает, насколько близка вероятность предсказания к соответствующему истинному значению (0 или 1 в случае бинарной классификации). Чем больше прогнозируемая вероятность отклоняется от фактического значения, тем выше значение логарифма потерь. Формула расчета Log-Loss будет приведена чуть позже.

Для примера рассмотрим задачу классификации электронных писем. Давайте представим спам как класс 1, а класс "нормальных" писем как 0. Давайте изучим настоящее спам-письмо (фактическое значение равно 1) и статистическую модель, которая классифицирует это письмо как спам с вероятностью 1. Поскольку вероятность предсказания равна почти 1, то и разность между предсказанной вероятностью и фактическим классом равна почти 0. Нулю равен, следовательно, и логарифм этой разности. Пример такого предсказания приведен на рисунке 15:

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	РАЗНОСТЬ	ЛОГАРИФИЧЕСКАЯ ПОТЕРЯ
1318	Попробуйте бесплатную защиту от телефонного спама и выиграйте один из пяти iPhone 12 на 256 ГБ...	1	1,00	0,00	0

Рисунок 15 – Результаты предсказания с вероятностью 1

Рассмотрим на рисунке 16 еще одно спам-письмо, классифицированное как спам с вероятностью 0,9. Вероятность прогноза модели на 0,1 отличается от фактического значения 1, и, следовательно, значение логарифмической потери больше нуля (равно 0,105).

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	РАЗНОСТЬ	ЛОГАРИФИЧЕСКАЯ ПОТЕРЯ
64247	Привет. Надеюсь, у тебя было время посмотреть, какие услуги по строительству предлагает моя компания...	1	0,90	0,10	0,105

Рисунок 16 – Результаты предсказания с вероятностью 0.9

А теперь давайте посмотрим на рисунке 17 на обычное электронное письмо. Модель классифицирует его как спам с вероятностью 0,2, то есть считает нормальным письмом (при условии, что порог по умолчанию равен 0,5). Абсолютная разница между вероятностью предсказания и фактическим значением, равным 0 (так как это нормально), составляет 0,2, что больше, чем то, что мы наблюдали в предыдущих двух наблюдениях. Значение логарифма потерь, связанное с прогнозом, составляет 0,223.

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	РАЗНОСТЬ	ЛОГАРИФИЧЕСКАЯ ПОТЕРЯ
2101	Скажи, когда тебе необходимо настроить рабочий ПК. У меня есть...	0	0,20	-0,20	0,223

Рисунок 17 – Результаты предсказания с вероятностью 0.2

Обратите внимание, как теперь значение Log-Loss худшего прогноза (удаленного от фактического значения) выше, чем у лучшего прогноза (ближе к фактическому значению).

Теперь предположим, что существует набор из 5 различных спам-писем, прогнозируемых с широким диапазоном вероятностей 1.0, 0.7, 0.3, 0.009 и 0.0001. Обученная статистическая модель неидеальна и, следовательно, выполняет (действительно) плохую работу по последним трем наблюдениям (классифицирует их как нормальные, поскольку значения вероятности ближе к 0, чем к 1). На рисунке 18 видно, что значение логарифмических потерь экспоненциально возрастает по мере того, как растет разность между реальным классом и предсказанной вероятностью:

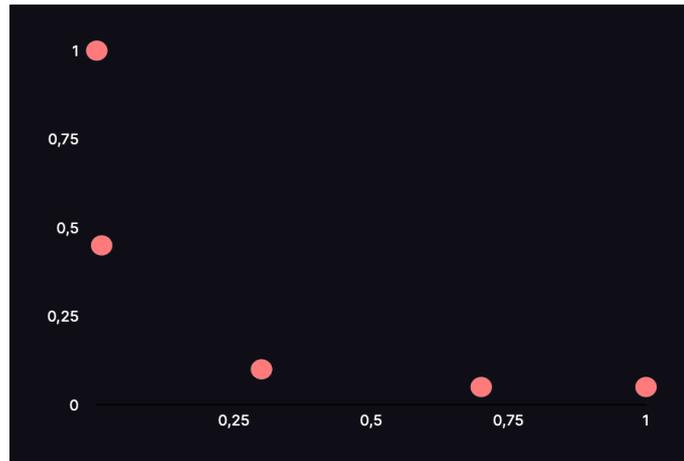


Рисунок 18 – Значения логарифмических потерь при прогнозах с разной вероятностью

Если мы построим график логарифмических потерь для перечня спам-писем со всеми возможными видами вероятностей, график будет выглядеть как на рисунке 19.

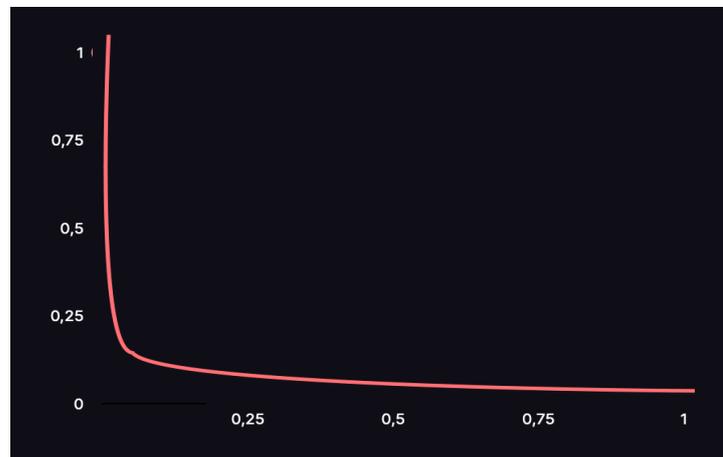


Рисунок 19 – Значения логарифмических потерь при всех возможных вероятностях прогнозов

В случае с нормальными письмами график будет зеркальным отображением приведенного выше и будет изображаться как на рисунке 20.

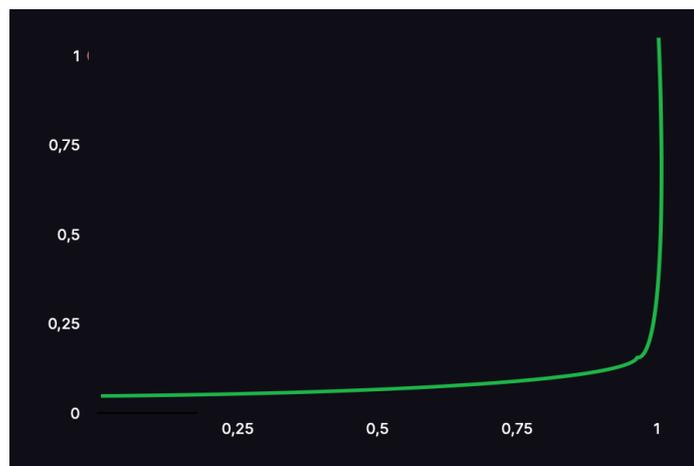


Рисунок 20 – Значения логарифмических потерь при всех возможных вероятностях прогнозов для нормальных писем

Подводя итог, можно сказать, что чем дальше вероятность предсказания от фактического значения, тем выше значение логарифмических потерь. При обучении модели классификации мы хотели бы, чтобы наблюдение предсказывалось с вероятностью, максимально приближенной к фактическому значению (0 или 1). Следовательно, Log-Loss – хороший выбор в качестве функции

потери для обучения и оптимизации. Чем дальше вероятность предсказания от ее истинного значения, тем выше штраф.

Теперь, когда мы понимаем логику, лежащую в основе метрики, мы можем посмотреть на конечную формулу на рисунке 21.

$$\text{Log} - \text{Loss} = -[y_i \times \ln(p_i) + (1 - y_i) \times \ln(1 - p_i)]$$

y_i – истинный класс наблюдения,

p_i – предсказанная вероятность

Рисунок 21 – Формула логарифмических потерь

Чтобы оценить модель в целом, вычисляется среднее арифметическое логарифмических потерь всех наблюдений. Модель с совершенными предсказаниями имеет логарифм потерь, равным нулю. Другими словами, идеальная модель предсказывает вероятность каждого наблюдения как фактическое значение.

Логарифмическая потеря для бинарной классификации – тоже, что и среднеквадратическая ошибка для регрессии. Обе метрики показывают, насколько хороши или плохи результаты прогнозов, указывая на дистанцию между прогнозом и фактическим значением.

Логарифмические потери можно рассчитать с помощью SkLearn. Для начала импортируем функцию:

```
import matplotlib.pyplot as plt
```

Применим функцию "на бегу", передав аргументы-списки:

```
log_loss(["спам", "нормальное письмо", "нормальное письмо", "спам"],
[[.1, .9], [.9, .1], [.8, .2], [.35, .65]])
```

Система здесь уже вынесла свой вердикт, и во втором списке находятся, как можно догадаться, пары значений, описывающих вероятности "спам" и "нормального письма". Функция сама определяет, что первый элемент внутреннего списка – число 0,1, описывает вероятность письма быть нормальным письмом, а второй – соответственно, спамом, и применив такой паттерн ко всем остальным парам значений, вычисляет разность между реальным классом и предсказанной вероятностью. Следуя формуле, описанной выше, она находит значение Log-Loss для каждого наблюдения и усредняет полученный результат получая в итоге 0.21616.

Ассигасу — это показатель, который описывает общую точность предсказания модели по всем классам. Это особенно полезно, когда каждый класс одинаково важен. Он рассчитывается как отношение количества правильных прогнозов к их общему количеству.

Метрика асс содержит результат деления суммы True Positive и True Negative прогнозов на количество всех прогнозов. Таким образом, ассигасу, равная 0.5714, означает, что модель с точностью 57,14% делает верный прогноз.

В модуле sklearn.metrics есть функция precision_score(), которая также может вычислять ассигасу. Она принимает в качестве аргументов достоверные и предсказанные метки.

Стоит учесть, что метрика ассигасу может быть обманчивой. Один из таких случаев — это несбалансированные данные. Предположим, у нас есть всего 600 единиц данных, из которых 550 относятся к классу Positive и только 50 — к Negative. Поскольку большинство семплов принадлежит к одному классу, ассигасу для этого класса будет выше, чем для другого.

Если модель сделала 530 правильных прогнозов из 550 для класса Positive, по сравнению с 5 из 50 для Negative, то общая ассигасу равна $(530 + 5) / 600 = 0.8917$. Это означает, что точность модели составляет 89.17%. Полагаясь на это значение, вы можете подумать, что для любой выборки (независимо от ее класса) модель сделает правильный прогноз в 89.17% случаев. Это неверно, так как для класса Negative модель работает очень плохо.

Для вычисления значения метрик возьмём следующие модели: нашу первую модель, модель в которой мы ввели новую метрику эло для анализа и "наивную" модель, где вероятность победы каждого из игроков равна 0.5.

Получаем следующие результаты, которые можем увидеть на рисунке 22

	logloss	AUC	accuracy
model	0.529107	0.800347	0.712598
elo_model	0.515281	0.825118	0.763780
true_naive	9.654653	0.720486	0.720472

Рисунок 22– Результаты валидации моделей

На основе результатов можно сказать следующее: первичная модель и модель на основе Эло показали сравнимые хорошие показатели. Однако Эло модель по всем показателям обошла первичную.

В то же время модель, всегда отдающая результат 0.5 сильно уступила лишь в logloss метрике. Это подчеркивает важность выбора метрики для валидации модели обучения. Некоторые метрики являются неинформативными, в другое время как остальные помогают качественно провалидировать прогнозы полученной модели.

Дальнейшими перспективами развития данной работы можно назвать вычисление оптимального размера скользящего окна по матчам для агрегирования метрик, оптимальной длительности обучения модели и обучения модели для различных типов кортов.

В ходе данной работы было получено понимание того, что выбор правильных метрик для обучения зачастую важнее подбора гиперпараметров для обучения.

А также выявлена необходимость подбора правильной метрики для валидации, чтобы она была информативной.

Список использованных источников:

1. Forecasting the results of the world winner universiade [Electronic Resource] /ScienceReview. – Mode of access: <https://science-engineering.ru/ru/article/view?id=1238>. Date of access: 07.04.2023.
2. ATP Tennis Rankings, Results, and Stats [Electronic Resource] / JeffSackmann. – Mode of access: https://github.com/JeffSackmann/tennis_atp. Date of access: 07.04.2023.
3. Deuce [Electronic Resource] / Skoval. – Mode of access: <https://github.com/skoval/deuce>. Date of access: 07.04.2023.
4. Introduction to R [Electronic Resource] / Ahmedushka7. – Mode of access: https://ahmedushka7.github.io/R/scripts/hse_data_analysis/sem_1/introduction_to_R.html. Date of access: 07.04.2023.
5. Parsing data from sites [Electronic Resource] / Ringostat. – Mode of access: <https://blog.ringostat.com/ru/parsing-dannyh-s-saytov-chto-eto-i-zachem-on-nuzhen/>. Date of access: 07.04.2023.
6. Nuances of Programming [Electronic Resource] / Medium. – Mode of access: <https://medium.com/nuances-of-programming>. Date of access: 07.04.2023.
7. XGBoost Model [Electronic Resource] / Techcave. – Mode of access: <https://techcave.ru/posts/81-sozdaem-pervuyu-xgboost-model-na-python-s-ispolzovaniem-scikit-learn.html>. Date of access: 07.04.2023.
8. Permutation importance [Electronic Resource] / Kaggle. – Mode of access: <https://www.kaggle.com/code/dansbecker/permutation-importance/tutorial>. Date of access: 07.04.2023.
9. ROC and AUC metrics [Electronic Resource] / Google. – Mode of access: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>. Date of access: 07.04.2023.
10. Elo rating ranges [Electronic Resource] / TennisAbstract. – Mode of access: <http://www.tennisabstract.com/blog/2019/12/03/an-introduction-to-tennis-elo/>. Date of access: 07.04.2023.
11. Numpy [Electronic Resource] / PythonWorld. – Mode of access: <https://pythonworld.ru/numpy/1.html>. Date of access: 07.04.2023.
12. Log-loss Metric [Electronic Resource] / Helenkapatsa. – Mode of access: <https://www.helenkapatsa.ru/logharifmichieskaia-potieria>. Date of access: 07.04.2023.
13. Accuracy Metric [Electronic Resource] / PythonRu. – Mode of access: <https://pythonru.com/baza-znaniij/metriki-accuracy-precision-i-recall>. Date of access: 07.04.2023.

NEURAL NETWORK APPROACH TO PREDICTION OF SPORTS TENNIS DATA

Kharkevich A.P.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Rykova O.V. – PhD in Physics and Mathematics

Annotation. In this paper, we consider the main stages of obtaining a ready-made neural network for predicting sports tennis data, analyzing the quality of training, and then optimizing the model.

Keywords. Python, pandas, xgboost, neural network, predicting results, tennis.

УДК

ОЦЕНКА СТОЙКОСТИ ГЕНЕРАТОРОВ СЛУЧАЙНЫХ ЧИСЛОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

Касьян В.А., студент гр. 253501, Новицкий З.Я., студент гр. 253501

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Стройникова Е.Д. – старший преподаватель каф. информатики

Аннотация. Генераторы псевдослучайных чисел (PRNG) представляют собой алгоритмы, которые генерируют числа, кажущиеся случайными, но при этом являются определенными. Они являются неотъемлемой частью многих приложений, от криптографии до статистических имитаций. Различные операционные системы и языки программирования имеют свои собственные реализации PRNG. Данное исследование направлено на анализ и сравнение алгоритмов, преимуществ и производительности PRNG в обеих операционных системах, а также предлагает тесты для оценки их эффективности.

Ключевые слова. Linux, Windows, PRNG, HRNG, CSPRNG, LCG, SHA-1, CryptoAPI, НИСТ, NIST

Определения: случайная числовая последовательность (СЧП) — числовая последовательность, полученная каким-либо способом, в которой невозможно предсказать следующее число ввиду его неопределенности до генерации. Псевдо-СЧП (ПСЧП) — СЧП, которую можно предсказать. Генератор случайных чисел (RNG) — реальное или виртуальное устройство, генерирующее случайные числовые последовательности. Аппаратный генератор случайных чисел (HRNG) — RNG, использующий физические эффекты, которые современная физика не может предсказать, например квантовые явления, или например физически неклонированные функции [1]. Генератор псевдослучайных чисел (PRNG) — виртуальный RNG, являющийся комбинацией математических функций для генерации (ПСЧП).

Рассмотрим различные PRNG и HRNG. Все программные генераторы являются PRNG, кроме случая использования HRNG. Приведем пример. В операционной системе (OS) GNU/Linux основным генератором псевдослучайных чисел (PRNG) является файл устройства `/dev/urandom`, который реализует криптографически стойкий генератор псевдослучайных чисел (CSPRNG). CSPRNG использует энтропию, собранную из различных источников, таких как ввод с клавиатуры и мыши, активность диска и аппаратные события. Собранная энтропия используется вместе с криптографической хэш-функцией (SHA-1) и линейным конгруэнтным генератором (LCG) для создания псевдослучайных чисел. Он разработан с расчетом прежде всего на быстроедействие. Для большей надежности используется файл `/dev/random`, который всегда набирает гораздо больший пул энтропии. Также оба файла используют доступные HRNG, если таковые есть в системе.

В OS Windows функция `CryptGenRandom`, входящая в состав Microsoft Cryptographic Application Programming Interface (CryptoAPI), отвечает за генерацию псевдослучайных чисел. Она полагается на поточный шифр RC4 в качестве алгоритма генерации псевдослучайных чисел. Аналогично GNU/Linux, Windows собирает энтропию из различных аппаратных источников и объединяет ее с заданным пользователем значением семени. Значение семени затем используется алгоритмом RC4 для генерации псевдослучайных чисел.

Во многих языках программирования (ЯП) используются случайные числа, представляемые OS, однако в некоторых применяется собственная генерация. Так, в языке C используется достаточно старый алгоритм LCG без добавления каких-либо усовершенствований вроде хэш-функций или собирания энтропии. В ЯП C++, Python, Java, Ruby для генерации ПСЧП реализован алгоритм Мерсенна – Твистера. Это генератор линейной обратной связи сдвигового регистра (LFSR), что означает, что он использует регистр сдвига для хранения последовательности битов и функцию обратной связи для генерации новых битов на основе содержимого регистра. Алгоритм использует большой вектор состояния (обычно 624 или 19937 бит) для генерации последовательности 32-битных целых чисел.

Теперь рассмотрим устройство HRNG. Наиболее часто используемая технология в HRNG — физически неклонированные функции (Physical Unclonable Functions, PUF) — это тип криптографических функций, которые используются для генерации уникальных ключей на основе

физических 1 характеристик электронных компонентов. PUF — это аппаратные системы, которые генерируют уникальный ключ на основе физических отличительных особенностей микрочипа, таких как индивидуальные различия в параметрах транзисторов, шумах в кристаллической решетке и т.д.

PUF функции генерируют уникальный ключ, который невозможно воспроизвести в другом микрочипе, что делает PUF ключи полезными для криптографических приложений, таких как аутентификация и шифрование.

Существует несколько типов PUF, таких как термальные PUF, оптические PUF, акустические PUF и т.д., каждый из которых использует различные методы для генерации уникальных ключей. PUF функции являются привлекательной альтернативой для традиционных методов генерации ключей, т.к. они генерируют ключи на основе физических характеристик, которые трудно воспроизвести, и не требуют хранения ключей в памяти устройства, что делает их более устойчивыми к атакам по сравнению с традиционными методами. Таким образом, СЧП, полученные с помощью PUF или квантовых эффектов невозможно воспроизвести еще раз, что делает их идеальным решением для создания HRNG. Однако основная проблема и отслеживания квантовых эффектов, и PUF — дороговизна оборудования и относительно низкая скорость работы, поэтому в современном мире преимущественно используются PRNG. Однако стоит оценить различные PRNG, ведь если они ненадежные, то их нельзя уверенно использовать для генерации ПСЧП. В данной работе изучим различные PRNG и дадим им оценку стойкости.

Прежде всего, как любые HRNG, PRNG должны максимально соответствовать следующим критериям: быть криптографически стойкими, иметь источники энтропии, быть производительными.

Оценка алгоритмов генерации псевдослучайных чисел выполняется с использованием следующих основных критериев:

- **Равномерность:** распределение созданных чисел должно быть равномерным по всем возможным значениям.
- **Независимость:** созданные числа не должны иметь никакого заметного шаблона или корреляции.

Равномерность относится к распределению сгенерированных чисел по всем возможным значениям. Хороший PRNG должен производить числа, которые следуют равномерному распределению, обеспечивая равную вероятность генерации каждого значения. Это свойство является важным для поддержания случайности и непредсказуемости сгенерированных чисел. В PRNG GNU/Linux и Windows используются хеш-функции для обеспечения генерации равномерно распределенных ПСЧП (ПСЧП). ЯП, реализующие алгоритм Мерсенна – Твистера также генерируют РСЧП. Алгоритмы, работающие только на основе LCG, не достигают равномерного распределения ПСЧП.

Независимость относится к отсутствию видимых закономерностей или корреляций между последовательными числами в сгенерированной последовательности. Качественный генератор псевдослучайных чисел должен производить числа, которые независимы друг от друга, обеспечивая то, что информация о предыдущих числах не дает никакой информации о будущих числах в последовательности. Это свойство особенно важно для криптографических приложений, где предсказуемость сгенерированных чисел может подорвать безопасность системы. В PRNG GNU/Linux и Windows используются пул энтропии для обеспечения независимости сгенерированных случайных чисел. Алгоритм Мерсенна – Твистера является предсказуемым, однако для этого необходимо получить сид генерации, что является проблематичной задачей, учитывая диапазон генерации алгоритма: $Z^p - 1$, p – простое число, обычно 19937.

Для проверки производительности и качества PRNG-алгоритмов воспользуемся несколькими хорошо известными тестовыми наборами. Эти тесты оценивают различные свойства генерируемых псевдослучайных чисел, такие как равномерность, независимость и воспроизводимость. Следующие тестовые наборы широко признаны своей строгостью и эффективностью при оценке PRNG:

Оперативное тестирование Diehard. Оперативное тестирование Diehard является еще одним широко используемым набором тестов для оценки PRNG. Он состоит из 18 тестов:

Тест на расстояние между днями рождения (Birthday Spacings Test), тест на перекрывающиеся перестановки из 5 элементов (Overlapping 5-Permutations Test), двоичный тест на ранг матриц размера 31x31 (Binary Rank Test for 31x31 Matrices), двоичный тест на ранг матриц

размера 32x32 (Binary Rank 2 Test for 32x32 Matrices), двоичный тест на ранг матриц размера 6x8 (Binary Rank Test for 6x8 Matrices), тест на битовую последовательность (Bitstream Test), тест на попарное пересечение разреженности (OPSO — Overlapping-Pairs-Sparse-Occupancy Test), тест на пересечение четверок разреженности (OQSO — Overlapping-Quadruples-Sparse-Occupancy Test), тест ДНК (DNA Test), тест на подсчет единиц в определенных байтах (Count the 1s Test for specific bytes), тест на подсчет единиц во всех байтах (Count the 1s Test for all bytes), тест парковки (Parking Lot Test), тест минимального расстояния (Minimum Distance Test), тест на сферу в 3D (3D Sphere Test), тест сжатия (Squeeze Test), тест на перекрывающиеся суммы (Overlapping Sums Test), тест на серии (Runs Test), тест на кости (Craps Test) [2].

TestU01 — это более новый и продвинутый набор тестов для тестирования ГПСЧ, предлагающий строгую оценку качества их выходных данных. Разработанный Пьером Лекье и Ричардом Симардом, TestU01 включает несколько заранее определенных наборов тестов, таких как:

1. SmallCrush: базовый набор тестов для предварительной оценки качества PRNG.
2. Crush: более расширенный набор тестов, обеспечивающий полную оценку качества PRNG.

3. BigCrush: самый строгий набор тестов, созданный для тщательной оценки качества PRNG и предназначенный для выявления даже незначительных недостатков в сгенерированных числах.

Набор тестов Национального института стандартов и технологий США (NIST) является всесторонним набором статистических тестов, разработанных для оценки случайности и качества PRNG. NIST предлагает несколько тестов для проверки свойств конечной последовательности, и для этого вычисляется статистика, которая может быть как одним значением, так и множеством значений. Затем эта статистика сравнивается с эталонной статистикой, которая представляет собой математический вывод идеально случайной последовательности. Для вывода эталонной статистики используется множество теорем и научных исследований.

Набор включает 15 тестов, каждый из которых нацелен на различные свойства и характеристики генераторов случайных чисел. Тесты, включенные в набор NIST:

Частотный (побитовый) тест, частотный блочный тест, тест на одинаковые идущие подряд биты, тест на максимальную длину серии, тест на ранг, дискретное преобразование Фурье (спектральный тест), тест совпадения шаблона без перекрытия, тест совпадения шаблона с перекрытием, универсальный статистический тест, тест на приближенную энтропию, тест на случайные блуждания, вариант теста на случайные блуждания, серийный тест, тест на линейную сложность, тест на кумулятивную сумму [3], [4].

Также было принято решение реализовать и исследовать несколько тестов для проверки последовательности чисел на случайность.

В статистических исследованиях используется понятие нулевой гипотезы, которая предполагает отсутствие взаимосвязи между рассматриваемыми факторами. Например, в случае исследования связи курения и заболевания раком легких нулевая гипотеза предполагает, что курение не вызывает рак легких. Также есть альтернативная гипотеза, которая опровергает нулевую гипотезу и предполагает наличие взаимосвязи между факторами.

Для проверки гипотез используется собранная статистика, которая сравнивается с эталонной статистикой, полученной по математическим методам. При этом вводится погрешность, например в 5%. Если отклонение собранной статистики от эталонной превышает заданный уровень погрешности, делается вывод о том, что нулевая гипотеза не верна. Существуют 4 возможных варианта вывода: правильный вывод о случайности исследуемой последовательности, ошибки первого рода (когда последовательность признается не случайной, хотя она является случайной), ошибки второго рода (когда последовательность признается случайной, хотя она не является таковой) и отбраковка.

В статистике вероятность ошибки первого рода обозначается как α . Это вероятность отбросить “хорошую” случайную последовательность. Значение α определяется областью применения, например, в криптографии α обычно берут от 0.001 до 0.01.

При проведении тестов на случайность генерируется т.н. P-значение, которое означает вероятность того, что подопытный генератор произведет последовательность не хуже, чем

гипотетический “истинный”. Если Р-значение равно 1, это означает, что наша последовательность идеально случайна, а если оно равно 0, то последовательность полностью предсказуема.

Для проведения тестов в данном случае берется $\alpha = 0.01$. Если Р-значение больше или равно 0.01, то последовательность признается случайной с уровнем доверия 99%. Если Р-значение меньше 0.01, то последовательность отбраковывается с уровнем доверия 99%. Было сгенерировано 4 последовательности:

Алгоритмом Мерсенна – Твистера

```
00110100110101111010011000010010110111111000111101000011011001001101100100011011111
1110010011001001000001101100010111000111100110001000010000100011001111100110101
010010100010001100011001101000000110101101101111011011001010011001101001111010010001
01111000110010001000001011100011010010110101001101110110010100010100110100101011101
11001001001011010010101111100 01111000001011001110001011010001111
```

PRNG OS GNU/Linux/dev/random

```
10100110001101110110100011000001100111110000111111000101011110110001100001110010110
110011110101100111000010011101110000101001100111000010000100001100001001100000001111
011001011100110110100010101110111010001010011110010001010110001111011000011000110110
00001111001001000011100101100100101101111011100110101000100010111101010000010000110
10100111100010001111101011011 00000000000011110000110000101011110
```

PRNG OS GNU/Linux/dev/urandom

```
0010111010001111000101111100110001001111111011010011101011101011010110101110000101
101000111100110000001110001110001011011011001001110100101100110100000000110001100101
010110011110100101110010010010010001011001010100111111001101100010100101000110000
1101110101000111000100110110101111101111001111000001000001110001111110110011011101
00101111011101011010010100110 1011001010000010001100110101101010
```

Устаревший метод в ЯП С

```
010100110001011001100110011010011000011011100100010010011110111011010100010000100101
100110101111010110110100111101000111100011100011001011100000100111100111001000010111
000000000110000011010100001011011010010001010111011011111010111110111100001010000101
010101111000010100110010110100000010101000101100111100110100001000011101110100111001
00001011001001011010111110000 00100101011010100010101010100000011
```

Частотный (побитовый) тест.

$$S_n = \sum_{i=1}^n X_i, \text{ где } X_i = 2x_i - 1, \text{ где } x_i - i\text{-ый бит.}$$

Статистика:

$$S_{obs} = \frac{|S_n|}{\sqrt{n}}, \text{ где } n - \text{ количество битов.}$$

Р-значение:

$$P_{value} = \text{erfc}\left(\frac{S_{obs}}{\sqrt{2}}\right), \text{ где } \text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt - \text{дополнительная функция ошибок.}$$

Рекомендуется тестировать последовательности длиной не менее 100 бит.

Таблица 1. Последовательность 400 битов частотным (побитовым) методом (Генерация: алгоритмом Мерсенна – Твистера, PRNG OS GNU/Linux /dev/random, PRNG OS GNU/Linux /dev/urandom, устаревшим методом в ЯП С)

Генерация	Sn	Sobs	Р-значение
Алгоритм Мерсенна-Твистера	-8	0.4	0.69
PRNG OS GNU/Linux /dev/random	-16	0.8	0.42
PRNG OS GNU/Linux /dev/urandom	14	0.7	0.48
Устаревший метод в ЯП С	-22	1.1	0.27

Этот тест делается на основе предыдущего, только теперь значения пропорции «1»/«0» для каждого блока анализируются методом Хи-квадрат. Ясно, что это соотношение должно быть приблизительно равным 1. Для проведения теста требуется разбить последовательность битов на N блоков по M битов, неполный блок отбрасывается.

$X_{obs}^2 = 4 \cdot M \cdot \sum_{i=1}^N \left(\pi_i - \frac{1}{2} \right)^2$, где $\pi_i = \frac{K}{M}$, где K – количество бит, равных единице в данном блоке битов.

P-значение: $P_{value} = Q\left(\frac{N}{2}, \frac{x_{obs}^2}{2}\right)$, где $Q(a, x) = \frac{1}{\Gamma(a)} \int_x^\infty e^{-t} t^{a-1} dt$, – неполная верхняя гамма-функция, $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ – стандартная гамма-функция.

Для достаточно высокой точности примем $n = 400, M = 40 \Rightarrow N = \frac{n}{M} = 10$.

Таблица 2. Последовательность 400 битов частотным блочным методом (Генерация: алгоритмом Мерсенна – Твистера, PRNG OS GNU/Linux /dev/random, PRNG OS GNU/Linux /dev/urandom, устаревшим методом в ЯП С)

Генерация	X_{obs}^2	P-значение
Алгоритм Мерсенна-Твистера	9.2	0.51
PRNG OS GNU/Linux /dev/random	7.2	0.71
PRNG OS GNU/Linux /dev/urandom	7.3	0.70
Устаревший метод в ЯП С	14.1	0.17

Как видно из таблицы 2, во всех 10-ти тестах P-значение было больше 0.01, значит, все последовательности битов прошли тест.

Тест на одинаковые идущие подряд биты.

Этот тест проверяет последовательности одинаковых битов в заданной последовательности и определяет, соответствуют ли количество и размеры этих последовательностей ожидаемым для случайной последовательности. Если смена 0 на 1 или наоборот происходит слишком редко, то такая последовательность не является случайной.

$\pi = \frac{\sum_{i=1}^n X_i}{n}$ – доля единиц в последовательности битов.

Далее проверяется условие:

$$\left| \pi - \frac{1}{2} \right| < \frac{2}{\sqrt{n}}$$

Если оно не удовлетворяется, то весь тест считается неуспешным и на этом он заканчивается.

В противном случае вычисляем суммарное число знакоперемен V_n :

$V_n = \sum_{k=1}^{n-1} r(k) + 1$, где $r(k) = 0$, если $X_i = X_{i+1}$, или $r(k) = 1$ в противном случае.

Для вычисляем P-значение:

$$P_{value} = \operatorname{erfc} \left(\frac{|V_n - 2n\pi(1-\pi)|}{2\sqrt{2n\pi(1-\pi)}} \right).$$

Таблица 3. Последовательность 400 битов на одинаковые идущие подряд биты (Генерация: алгоритмом Мерсенна – Твистера, PRNG OS GNU/Linux /dev/random, PRNG OS GNU/Linux /dev/urandom, устаревшим методом в ЯП С)

Генерация	π	$\left \pi - \frac{1}{2} \right < \frac{2}{\sqrt{n}}$	V	P-значение
Алгоритм Мерсенна-Твистера	0.54	+	212	0.19
PRNG OS GNU/Linux /dev/random	0.48	+	188	0.24
PRNG OS GNU/Linux /dev/urandom	0.52	+	213	0.18
Устаревший метод в ЯП C	0.47	+	214	0.14

Эти наборы тестов могут применяться для проверки любых PRNG с целью оценки их производительности и сравнения качества их выходных данных. Анализируя результаты этих тестов, пользователи могут получить представление о сильных и слабых сторонах каждого PRNG и принимать обоснованные решения относительно их пригодности для конкретных приложений. Кроме того, регулярное тестирование PRNG с использованием этих наборов тестов помогает гарантировать, что любые обновления или модификации базовых алгоритмов не нарушают качество сгенерированных псевдослучайных чисел.

В целом, оценка стойкости генераторов случайных чисел является важной задачей, которая требует специальных знаний и опыта. Можно сделать вывод, что современные генераторы случайных числовых последовательностей проходят актуальные тесты на случайность, что означает пригодность их использования для генерации случайных чисел и числовых последовательностей для различных целей, например для криптографии. Лишь устаревшие PRNG проходят такие тесты, но и то не всегда.

Список использованных источников:

1. Хабр [Электронный ресурс]. — Режим доступа: URL: <https://habr.com/ru/articles/343386/>.
2. Duke. Department of physics [Электронный ресурс]. — Режим доступа: URL: <https://webhome.phy.duke.edu/~rgb/General/dieharder.php>.
3. Хабр [Электронный ресурс]. — Режим доступа: URL: <https://habr.com/ru/companies/securitycode/articles/237695/>.
4. NIST [Электронный ресурс]. — Режим доступа: URL: <https://csrc.nist.gov/publications/detail/sp/800-22/rev-1a/final>.

UDC

RANDOM NUMBER SEQUENCE GENERATORS SECURITY ESTIMATION

Kasyan V.A., Novickiy Z.Y.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Stroinikova E.D. – senior lecturer, Department of Informatics

Annotation. Pseudo-random number generators (PRNGs) are algorithms that generate numbers that appear to be random but are certain. They are an integral part of many applications, from cryptography to statistical simulations. Various operating systems and programming languages have their own PRNG implementations. This study aims to analyze and compare the algorithms, benefits, and performance of PRNGs on both operating systems, and proposes tests to evaluate their effectiveness. sequences

Keywords. Linux, Windows, PRNG, HRNG, CSPRNG, LCG, SHA-1, CryptoAPI, NIST

УДК 004.971

ПРИМЕНЕНИЕ ТЕХНОЛОГИЙ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ В ХИРУРГИЧЕСКОМ ОБРАЗОВАНИИ

Канавальчик А.Д., студентка группы 253506, Довнар А.Н., студент группы 253503,

Владымцев В.Д. – ассистент кафедры информатики

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Владымцев В.Д. – ассистент кафедры информатики

Аннотация. В научной статье исследовано, как технологии виртуальной реальности могут использоваться в хирургическом образовании, и какие преимущества они могут предоставить студентам и молодым хирургам. Также в работе приведены конкретные примеры технологий VR-обучения. Рассматривается роль преподавателя в процессе обучения при помощи VR-технологий. Проведен опрос среди студентов на тему применения VR технологий в медицинской хирургии в условиях, связанных с вынужденной мобильностью.

Ключевые слова. Виртуальная реальность, хирургия, образование, симулятор.

Введение.

В 21 веке технологии все больше и больше включаются в образовательные программы. Это помогает улучшить процесс обучения и сделать его более эффективным и продолжительным по сравнению со стандартными методами обучения с помощью книг. Появление симуляторов виртуальной реальности позволило осуществлять практическое обучение в различных областях. Одной из областей, которая больше всего выиграет от разработки обучающих программ виртуальной реальности, является медицина. В свою очередь хирургия является одной из самых важных областей медицины. Специальная виртуальная среда для студентов-хирургов предназначена для практики хирургических навыков с использованием разнообразных сценариев реальных операций, таких как удаление аппендикса, операция на легких, операция на сердце и т.д.

Основная часть.

Хирургия - это одна из наиболее сложных и ответственных областей медицины, где на высоком уровне требуются как теоретические, так и практические знания. Хирургическое обучение должно быть надежным и эффективным, чтобы студенты могли получить необходимые навыки и опыт без риска для жизни пациентов. В последние годы виртуальная реальность (VR) стала все более популярной в медицинском обучении, в том числе и в хирургическом образовании. VR может предоставлять студентам и молодым хирургам возможности для обучения без риска для жизни пациентов, максимально реалистично имитируя всевозможные реальные сценарии. Это особенно важно для сложных и редких операций, которые могут быть опасными для пациентов, если они выполняются неопытными врачами [1].

В нынешнем виде, хотя технология виртуальной реальности способна имитировать визуальные эффекты и изображения, чтобы они казались реальными, и проецировать движения человеческого тела в виртуальный 3D-мир, она неспособна имитировать другие факторы, которые играют важную роль в практическом медицинском труде. Например, ощущение давления и прикосновения. Для того, чтобы студенты чувствовали, что они оперируют на живых пациентах, а не размахивают инструментами в воздухе, необходимо создать среду, которая компенсировала бы недостатки существующей системы виртуальной реальности. Для этого комната для занятий должна быть оборудована так, чтобы она была похожа на настоящую операционную. К примеру, на рисунке 1 мы можем видеть, как хирург "оттачивает" свое мастерство, используя при этом симулятор FundamentalVR, однако только таких симуляторов недостаточно для полноценного обучения. В ней должны быть все инструменты и предметы, обычно используемые в хирургической практике, а также аппарат для искусственного дыхания и другое оборудование. В дополнение к практике хирургии, студенты смогут изучать и практиковать роли вспомогательного персонала, такие как медсестры-хирурги.



Рисунок 1 – Пример оттачивания медицинских навыков посредством VR технологий

Для более продвинутых студентов, VR может использоваться для практики хирургических процедур. Симуляторы могут создавать реалистичные 3D-модели органов, на которых студенты могут учиться и практиковаться в различных процедурах. Симуляторы могут также включать различные сценарии и условия, такие как кровотечение, что позволит студентам получить опыт работы с различными хирургическими ситуациями.

VR может использоваться для обучения профессиональным навыкам, таким как управление инструментами и точность движений. Студенты могут практиковать свои навыки в виртуальной среде, что даст им более полное представление о том, как использовать инструменты в реальных хирургических ситуациях.

Также VR может быть использована для обучения командной работе в операционной. Студенты могут практиковать коммуникацию, координацию и распределение ролей в хирургической команде, используя виртуальную среду. Это может помочь студентам стать более уверенными и опытными участниками хирургических команд в реальных ситуациях.

Программа обучения будет основана на симуляторе виртуальной реальности. Важнейшим компонентом всей системы будет интерфейс виртуальной реальности, который будет проецировать виртуальную среду на существующую область обучения. Другая важная технологическая реализация будет включать разработку программного обеспечения, которое сможет взаимодействовать с симулятором виртуальной реальности и проецировать действия студента в виртуальную реальность, предлагая соответствующую визуальную и тактильную обратную связь [3]. Другими словами, каждое действие, которое студент совершает во время обучения, должно казаться реальным. Неправильный разрез разорвет артерию, в то время как правильный разрез сделает процедуру намного чище.

Другим потенциальным технологическим инструментом, помогающим повысить реалистичность моделирования и улучшить процесс обучения, было бы предоставление модели тела, которая взаимодействовала бы с инструментами и оборудованием, используемыми студентами. Эта модель должна быть подключена к программному обеспечению и инструменту виртуальной реальности для получения обратной связи. Другими технологическими инструментами могут быть различные медицинские приборы, используемые в хирургии [3]. Студентов научат оперировать их в условиях, максимально приближенных к симуляции. Симуляция должна быть способна поддерживать множество пользователей одновременно. Таким образом, она сможет повторить усилия всей команды медсестер. Программное обеспечение должно предоставлять множество потенциальных сценариев. Некоторые из них будут включать экстренные операции, операции по удалению скоб и другие.

Во время виртуальной симуляции преподаватель возьмет на себя роль эксперта. Для того, чтобы контролировать действия студентов, преподаватель может либо лично принять участие в симуляции, либо наблюдать за действиями всех студентов с помощью экрана, который показывает действия каждого отдельного участника с разных ракурсов камеры. Первый метод хорош для

инициации и практических инструкций, поскольку преподаватель сможет представить студентам правильный способ выполнения операций. Последняя отлично подходит для анализа индивидуальной работы и предоставления отзывов об их усилиях. Методы обучения, связанные с моделированием виртуальной реальности, следующие:

1. Демонстрационный метод – преподаватель демонстрирует правильные способы выполнения различных хирургических техник и объясняет их обоснование.
2. Метод ролевой игры – студенты, участвующие в виртуальной симуляции, берут на себя роль хирургов и помощников хирурга, чтобы выполнить операцию в виртуальной среде.

Одним из примеров VR-обучения для будущих хирургов является программное обеспечение Surgical Theater, которое используется в Нью-Йоркском университете. Эта технология предоставляет студентам возможность практиковать различные операции в виртуальной среде, которая имитирует настоящие операционные залы и оборудование. Это позволяет студентам получать практический опыт без необходимости рисковать здоровьем пациентов [2].

Однако, стоит отметить, что не всегда использование VR-технологий на практике возможно. Например, в экстренных условиях в поле боя врачи вынуждены спасать жизни людей собственными руками и с минимальным набором инструментов. В таком случае использование VR-технологий на практике элементарно невозможно.

Анализируя результаты опроса, проведенного нами среди студентов ВМУ, на тему применения VR технологий в медицинской хирургии в условиях, связанных с вынужденной мобильностью, мы сделали следующий вывод:

Использование VR технологий в процессе обучения медицинских работников способно значительно улучшить их профессиональные теоретические навыки в области хирургии. Однако, полное замещение ручной работы VR технологиями не рекомендуется.

Это объясняется тем, что врачам все еще нужно иметь навыки работы с реальными инструментами и уметь справляться с непредвиденными ситуациями в хирургическом процессе. Кроме того, VR технологии могут не учитывать факторы, такие как давление, вибрации и другие физические параметры, которые могут возникнуть в реальных условиях.

В связи с этим, мы рекомендуем комбинировать обучение с использованием VR технологий и реальной практики, чтобы медицинские работники могли получить максимальный эффект в обучении и научиться работать в реальных условиях. Также важно отметить, что использование VR технологий позволяет сократить риски для пациентов и медицинских работников, что является важным фактором в медицинской практике.

The role of the simulator



Рисунок 2 – Роль симуляторов в обучении

График, изображенный на рисунке 2, подтверждает нашу теорию о том, что для достижения лучшего результата следует использовать комбинированное обучение [4].

Несмотря на все выше сказанное, потенциал для обучения с помощью специального модуля виртуальной реальности для хирургов огромен. Студенты смогут практиковать двигательные навыки во время выполнения практической операции, улучшить идентификацию различных признаков и симптомов, улучшить навыки принятия решений, научиться работать в составе

хирургической бригады и ознакомиться со всеми необходимыми практиками и процедурами в обстановке, очень близкой к реальности. Хотя виртуальная реальность с ее нынешними технологическими ограничениями не может полностью воссоздать опыт реальной операции, которая включает в себя не только точность изображений и звуков, но также запахов и тактильных проявлений, она предоставит студентам необходимый практический опыт обучения, которого многим студентам-хирургам часто не хватает по окончании учебы в университете. Это позволит намного быстрее подготовить квалифицированные кадры хирургов, что должно повысить уровень знаний и общее качество медицинской помощи. Однако, использование VR технологий больше нацелены на повышение теоретических навыков, нежели практических, так как далеко не все процессы, происходящие во время реальной операции, способны воплотить VR технологии. Именно поэтому для повышения качества практических навыков эффективнее комбинировать реальную практику и обучение с использованием VR технологий.

Заключение.

Технологии VR могут быть полезными для обучения студентов, так как они могут обеспечить доступ к определенным видам оборудования, которые могут быть дорогими или редкими в использовании. Это может помочь студентам развить практические навыки, связанные с обслуживанием и использованием сложного оборудования. Кроме того, использование VR может помочь улучшить безопасность пациентов, так как студенты могут учиться в безопасной и контролируемой среде без риска для жизни и здоровья пациентов. Однако для достижения максимальной эффективности, мы рекомендуем комбинировать обучение с использованием VR технологий и реальной практики.

Список использованных источников:

1. *Virtual Reality in Medicine* [Электронный ресурс] / – Режим доступа: <https://www.frontiersin.org/journals/virtual-reality/sections/virtual-reality-in-medicine>. – Дата доступа: 10.04.2023
2. *Virtual Reality in Medicine* [Электронный ресурс] / – Режим доступа: <https://www.frontiersin.org/journals/virtual-reality/sections/virtual-reality-in-medicine>. – Дата доступа: 10.04.2023
3. *Virtual Reality in Clinical Practice and Research* [Электронный ресурс] / – Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8968556>. – Дата доступа: 10.04.2023
4. *Better Patient Care and Safety* [Электронный ресурс] – Режим доступа: <https://surgicalscience.com/simulators/why-simulation>. – Дата доступа: 10.04.2023

UDC 004.971

THE USE OF VIRTUAL REALITY IN SURGICAL EDUCATION

Kanavalchik A. D., Dovnar A. N., Vladymtsev V.D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Annotation. The scientific article examines how virtual reality technologies can be used in surgical education, and what advantages they can provide to students and young surgeons. The paper also provides specific examples of VR-learning technologies. The role of the teacher in the learning process with the help of VR technologies is considered. A survey was conducted among students on the use of VR technologies in medical surgery in conditions associated with forced mobility.

Keywords. Virtual reality, surgery, education, simulator.

УДК 004.021

ИМИТАЦИЯ МОДЕЛИ ЗАРОЖДЕНИЯ ЖИЗНИ

Протьюко М.А., студент гр.050502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Борисенко О.Ф. – канд. физ.-мат. наук

Аннотация. Данная статья содержит описание «жизни» с точки зрения биоинформатики, а также возможный способ симуляции модели развития систем, способных этому определению соответствовать.

Ключевые слова. Континуальные модели, клеточный автомат, COMETS

Введение

Среди огромного разнообразия гипотез о возникновении жизни, особо выделяется та, что утверждает, будто бы жизнь развилась постепенно, все больше и больше усложняя цепочки элементов, которые позже образовали РНК и белки, а еще позже, первую функциональную клетку (или же клеточную структуру) [1]. Доказательством теорий возникновения жизни занимается синтетическая биология, основная цель которой - в лабораторных условиях воспроизвести биологическое поведение неестественных биологических систем. Аналогично синтетической биологии, было создано множество программных симуляций, позволяющих подробно описать законы жизни (COMETS, VacArena, Matnet, 3DdFBA и др.) [2], настолько подробно, что возможно имитировать поведение различных микроорганизмов, процесс мутации и эволюции, а также многие химические реакции, описывающие метаболизм.

Свойство, которое позволяет живому организму изменять свое поведение и структуру называется адаптацией, а в долгосрочном плане – эволюцией. Если разложить понятие «жизнь» на элементарные составляющие, сможем ли мы создать некое подобие эволюционного процесса, только в своей «неестественной системе»?

Цель данной статьи – определить такие элементы и свойства живого, которые возможно реализовать вычислительными средствами, подобно [2].

Природа жизни

Определим желаемый результат – выход данной симуляции, или же, «жизнь».

Согласно [3]:

«Биологический организм – это естественно возникающее, самовоспроизводящееся устройство, которое действует за счет управляемого превращения вещества, энергии и информации.» - далее, в данной работе будет использоваться именно это определение.

Живая клетка обладает следующими характеристиками:

- Метаболизм
- Энергия
- Клеточный рост, клеточное деление, синтез ДНК
- Транскрипция
- Синтез белков
- Пассивный транспорт
- Клеточный транспорт и транспортные механизмы
- Защита клеток, клеточная смерть и старение

Согласно [3], «можно предположить, что минимальный организм должен обладать способностью к воспроизведению, но не обязательно должен конкурировать в способности роста и размножения с другими организмами. Можно предположить, что минимальный организм должен усваивать питательную среду, обеспечивая организму биосинтез, а также обеспечивать ответ на стресс, в том числе и восстановление поврежденной ДНК.»

Т.е., необходимо задать такой набор правил, чтобы возможно было с их помощью описать метаболизм, а также набор катализаторов данного процесса (или же, аналогию с ДНК, что позволяет хранить информацию в системе).

Типы моделей

Рассмотрим общие математические модели, описывающие жизнь на примере биопленки (biofilm dynamics [4], биопленка – сообщество микроорганизмов, адаптирующееся к внешним условиям, причем микроорганизмы по отдельности могут это свойство не иметь).

Существуют следующие модели имитации сообщества микроорганизмов:

- Континуальные модели (continuum models)
- Модели IbM (individual based - индивидуальные)
- Модели CA (клеточный автомат)
- Гибридные

Как показали источники [3-4] использование вышеописанных моделей обладает следующими свойствами:

Континуальная модель позволяет имитировать физические законы, но не организм, способный адаптироваться (рассматривают микроорганизм как непрерывный материал и граничные условия, по которым на основе формул вязкости и диффузии, уравнения массопереноса и т.п. рассчитывается конечное состояние, представляемое набором численных параметров).

Индивидуальный подход (IbM) крайне тяжело оптимизировать и посчитать. По сути, IbM представляют собой список из одного и того же уравнения с разными параметрами, такими как скорость роста (условия роста) и распространение биомассы. Для каждого конкретного вещества для описания его реакций (а следовательно, и жизненного цикла) необходима своя формула, и свой расчет. Таких сфер (микроорганизм в данной модели – твердая сфера) в одной симуляции минимум 10^6 на единицу поверхности (по аналогии с квадратным сантиметром). Причем данная модель не предусматривает эволюцию, а больше подобна на игру «Жизнь» Конвея, позволяя из заданных начальных значений составить топологию.

Клеточный автомат (по определению из [3]) подобен объединению идеи генетических алгоритмов, описывающих имитацию конкуренции (Роевой интеллект, симуляция «хищник – жертва»). Основная проблема данного подхода – необходимо выбрать следствия, относительно которых будет строиться модель. Следствия – это набор простых правил (питание, гибель, размножение), взятые на основе биологических принципов. CA способны моделировать многие простые взаимодействия, но для более комплексных (формирование биопленки, образование матрицы EPS) они, как показала практика, не пригодны.

Из вышеописанного сделаем вывод, что для имитации процесса эволюции жизни необходимы гибридные модели.

Далее, опишем свойства гибридной модели, разделив процесс на несколько уровней, подобно уровням организации живого.

«Физика жизни»

Начнем с выбора такой системы простейших правил, которая позволяет рассчитать (1-4). В данном случае, определим, что будет в данной системе истинной, а что будет ложью (система аксиом [5])

Попробуем описать правила этого простейшего клеточного автомата, подобно правилам «Жизни» Конвея.

Элемент (на данном уровне – клетка, точка на пространстве, слот в «хромосоме» генетического алгоритма и т.д.) обладает двумя параметрами: скоростью передвижения (или же энергией) и положением в пространстве (описываемое неким радиусом действия элемента). Свойство энергии – стремление к энтропии, или же, равномерному ее распределению по всей симуляции. Т.е., элемент с большим Q стремится к элементу с малым Q, образуя тем самым связь. Данную связь можно разорвать другим элементом.

Т.е., поведение всей симуляции на данном уровне определяется двумя константами и двумя переменными: энергия связи, энергия разрыва – константы; энергия элемента, радиус действия элемента в некой системе координат - переменные. В системе задается время t , общее для всех уровней.

«Химия жизни»

Большая часть уравнений, описанных в данном разделе, рассчитывается с помощью матрицы стехиометрических коэффициентов. Разумеется, данная матрица должна присутствовать для каждого метаболита.

В качестве метаболита в имитационной модели может использоваться любой набор элементов «физического» уровня, стехиометрические коэффициенты которого будут посчитанные на основе формул, позволяющих определить характеристики живой клетки [3] (к примеру, матрица стехиометрических коэффициентов, описывающая элемент, подобный клеточной стенке бактерий и т.д.).

Уравнения, позволяющие описать усвоение питательной среды и биосинтез:

$$\frac{\partial B^\alpha}{\partial t} = \vartheta^\alpha B^\alpha \quad (1)$$

Где B^α – биомасса вида α , ϑ^α – скорость роста, посчитанная через FBA (Flux Balance Analysis), α – количество элементов некоего вида.

$$\frac{\partial Q^i}{\partial t} = \sum_{\alpha} \vartheta_i^\alpha B^\alpha \quad (2)$$

Где Q^i – объем внешнего метаболита (abundance of external metabolite), ϑ_i^α – скорость обмена метаболитом i в виде α .

Уравнения, позволяющие определить самовоспроизведение [6], это возможные зависимости между внешним и внутренним метаболитом: линейная функция концентрации, монотонная (Михаэль-Ментен) функция, или псевдо-монотонная функция.

Рассмотрим монотонную модель, описывающую отношения между ростом и концентрацией питательного субстрата (в случае данной модели – взаимоотношение между связанными элементами (уже на «химическом» уровне) и свободными).

$$\mu = \mu_{max} \left[\frac{S_i}{S_i + K_i} \right] \quad (3)$$

Где μ – скорость роста элементов α , S_i – концентрация метаболита i . μ_{max} – максимально допустимая скорость роста для данного α (выбираемая константа), K_i – насыщение i (в данной системе насыщением можно назвать доступную энергию метаболита. Расчет данной энергии Q (из уровня «физиологии» жизни производится расчет K_i по формуле, аналогичной формуле расчета сатурации кислорода. Данный расчет также зависит от выбора констант (уже для «химического» уровня) – для разделения метаболитов на активные, катализаторы и т.д.)

На основе модели Верхулста, или же метода идентификации логической динамики и жизненного цикла метаболита:

$$X(t) = \frac{X_i X_0 e^{\mu t}}{X_i - X_0 + X_0 e^{\mu t}} \quad (4)$$

Где X – концентрация выбранного метаболита i , X_i – его максимальная концентрация, X_0 – начальная концентрация метаболита i , t – время, μ – скорость роста элемента из (3).

Формулы (3-4) были выбраны поскольку для них существует статистика из исследований роста и эволюции микроорганизмов ([6] и [4])

Формулы (1-2) были выбраны, поскольку они хорошо показывают себя в симуляции COMETS.

Матрицу стехиометрических коэффициентов можно рассчитать на основе FBA (Flux Balance Analysis), которые позволяют, на основе некоторых ограничений предсказывать функцию или фенотип организма на основе его метаболизма [2]. Расчет данной функции также требует задания специфических формул и констант.

Информация

Согласно [7], в качестве параметра передачи информации выступает соответствие между одной группой элементов другой. Т.е., в данной модели должен присутствовать код, где изначальным алфавитом будет выступать набор неких «органических» метаболитов, а кодовым алфавитом – набор метаболитов-катализаторов. Данный процесс позволяет переходить из одной группы элементов к другой. Причем, данный код обладает избыточностью.

Избыточность кода можно получить путем сохранения соотношения между емкостью кода и количеством символов источника, подобно соотношению ДНК – аминокислоты.

К примеру, для длины кода 2 с основанием 4 получаем емкость 16, причем не кодирующих элементов может быть от 1 до 6. Оставшиеся последовательности кодируют 5 изначальных элементов.

Причем не столь важно, чтобы каждый метаболит из множества «органических» имел свою кодовую последовательность. Достаточно описания минимальных метаболитов, способных посредством дальнейших взаимодействий друг с другом создать необходимый набор (данный набор описывается FBA на «химическом» уровне).

Заключение

Данная модель требует большого количества параметров, требующих выбора (задание начальных параметров). Выбор параметров возможно оптимизировать. Поскольку имеется огромное количество данных, описывающих взаимодействия (1-4), можно использовать аналогию с

помощью нейронной сети, которая, обучаясь на выборке из работ, подобных [4] будет способна распознавать потенциально выигрышнее последовательности, задавая нужные входные параметры «физического» уровня и константы «химического». Т.е., цель данной нейронной сети - найти такой набор констант, чтобы принципы и формулы, описанные на каждом уровне, были верными.

Полученные решения теоретически должны соответствовать определению жизни.

Но стоит учитывать, что формулы (1-4) – лишь немногие из возможных разновидностей. Т.е., решения может и не быть.

Основные отличия данной модели от генетических алгоритмов и прочих симуляций органических процессов:

Элементы конечны – их количество остается неизменным с начала симуляции (т.е., энергия не возникает из ниоткуда и не исчезает в никуда).

Законы более высокого уровня организации выводимы из законов низкого уровня (т.е., данная модель – не набор следствий, как в классическом клеточном автомате биопленок, а набор причин)

Список использованных источников:

1. Беннер С.А., Ким Х.Дж., Ян З.: Подготовка сцены: история, химия и геобиология, стоящие за РНК. Колд Спринг Харб Перспектива Биол. 2012, 4: a003541-10.1101
2. Ilija Dukovski, Djordje Bajić, Jeremy M Chacón u др., *Computation Of Microbial Ecosystems in Time and Space (COMETS): An open source collaborative platform for modeling ecosystems metabolism* – [эл.источник] – URL - <https://arxiv.org/ftp/arxiv/papers/2009/2009.01734.pdf>
3. Леск А. Введение в биоинформатику / под ред. Миронова А.А, Шведаса В.К. // Москва, БИНОМ.Лаборатория знаний 2009 – 326 с.
4. David Rodriguez Espeso. *MODELING AND SIMULATION OF BACTERIAL BIOFILMS / Thesis Doctoral, DEPARTAMENTO DE CIENCIA E INGENIERÍA DE MATERIALES E INGENIERÍA QUÍMICA, Leganés, Junio de 2013 – 245.*
5. Протько М.А. Формализация и исследование замкнутых систем. 59 конференция студентов, магистрантов и аспирантов БГУИР, 2023, 3 с.
6. Siti suhana Jamaian *The Extended Monod Model for Microalgae Growth and Nutrient Uptake in Different Wastewaters* // *Universiti Tun Hussein Onn Malaysia International Journal of Engineering & Technology– 2018. 5 p.*
7. Протько М.А. Алгоритм кодирования процесса трансляции белков в клетке. Международный научно-технический семинар «Технологии передачи и обработки информации». БГУИР. 2023,6 с.

UDC 004.021

DESCRIPTION OF THE MODEL OF THE ORIGIN OF LIFE

Protsko M.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Borisenko O.F. – PhD in Physics and Mathematics

Annotation. This article contains a description of "life" from the point of view of bioinformatics, as well as a possible way to simulate a model of the development of systems capable of meeting this definition.

Keywords. Continuum models, cellular automata, COMETS

УДК

МЕДИЦИНСКОЕ ПРИЛОЖЕНИЕ ДЛЯ КЛАССИФИКАЦИИ ПЯТЕН НА КОЖЕ С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

Ермолович Д.С., студент гр.053504, Владимирцев В.Д. – ассистент кафедры информатики

*Белорусский государственный университет информатики и радиоэлектроники¹
г. Минск, Республика Беларусь*

Владимирцев В.Д. – ассистент кафедры информатики

Аннотация. С помощью машинного обучения было разработано медицинское приложение, которое позволяет людям молниеносно определять тип и поражение родинки, также определять степень и расположение ожога. Также для более точной классификации и сегментации был разработан алгоритм, который проверяет входные данные на аномалии.

Ключевые слова. Машинное обучение, математика, компьютерное зрение, классификация, детекция, сегментация, аномалии, анализ данных.

Введение:

С каждым годом число людей, страдающих раком кожи непрерывно увеличивается. Больше всего от этого страдают жители теплых стран, в которых круглый год солнце.

Также люди из этого региона подвергаются постоянным ожогам будь то от солнца или от других факторов.

Чтобы минимизировать вероятность получения рака кожи нужно оценивать свои родинки и смотреть за их изменениями, также при получении ожога необходимо его быстро детектировать и классифицировать, чтобы незамедлительно принять меры. Но к врачу обращаться постоянно дорого и иногда невозможно. Поэтому мною было принято решение, разработать приложение, которое помогает людям определять тип их родинки и выделить поражение родинки, классифицировать и детектировать ожоги.

Проблемы:

1 Проблема в распознавании родинок в том, что классов различных родинок много, а сами родинки друг на друга очень похожи. Также одна из основных проблем, это то, что данных мало, и классы не совсем сбалансированы. Поэтому нужно выяснить существует ли архитектура нейросети, которая хорошо решает поставленную задачу.

2 Так же нужно выяснить, существует ли алгоритм, который умеет определять аномалии, то есть если на вход подается не изображение родинки, а изображение какого-нибудь другого объект или плохое изображение родинки, то следует предупредить человека, о неправильных входных данных. В качестве основных нейросетей будут рассмотрены ae и vaе. Нужно понять какая из двух архитектур лучше решает поставленную задачу.

3 Проблема в сегментации родинок в том, что родинки на изображении маленькие и контуры по размеру и по структуре сильно не отличаются. Поэтому нужно выяснить существует ли такая архитектура нейросети, которая могла бы решить эту задачу с высоким уровнем точности.

4 Проблема классификации и детекции ожогов это то, что данных очень мало, из-за этого мы получаем плохие классификационные и детекционные результаты. Например, просто красные объекты на изображениях могут быть детектированы как ожоги.

Гипотезы:

1 Классификация

Проверить данные сверточные архитектуры и их вариации и определить, какая лучше всего подходит классификации родинок:

- 1.1 VGG-16 с заморозкой промежуточных слоев;
- 1.2 VGG-16 без заморозки промежуточных слоев;
- 1.3 WIDE_RESNET50_2 с коротким fc;
- 1.4 WIDE_RESNET50_2 с длинным fc;
- 1.5 EfficientNet_V2_L без заморозки промежуточных слоев;
- 1.6 EfficientNet_V2_L с заморозкой промежуточных слоев;
- 1.7 MOBILENET_V3_SMALL;
- 1.8 Дистилляция;

2 Определение аномалий

Посмотреть какая сеть справляется лучше с кодированием изображений родинок:

- 2.1 Variational autoencoder;
- 2.2 Autoencoder;

3 Сегментация

Определить, какая архитектура сверточной нейросети лучше всего подходит для сегментации родинок:

- 3.1 UNET(encoder resnet18);
- 3.2 UNET(encoder resnet101);
- 3.3 LinkNet;
- 3.4 PSPNet;
- 3.5 PAN;
- 3.6 Дистилляция;

4 Детекция и классификация

Определить, какая архитектура YOLO лучше всего подходит для детекции и классификации ожогов:

- 1 YOLOv5s(640x640);
- 2 YOLOv5x(256x256);
- 3 YOLOv5x(640x640);
- 4 YOLOv5m(640x640);

1 ОПРЕДЕЛЕНИЕ АНОМАЛИЙ

1.1 Идея алгоритма

Любое изображение представляет собой функцию вида $f(x, y, z)$. Любое изображение можно представить в виду вектора в латентном пространстве, где схожие объекты располагаются рядом друг с другом. Идея в том, чтобы найти такое сжатое представление изображения, чтобы оно содержало сжатую информацию о изображении.

Для этого я буду использовать autoencoder и variational autoencoder.

После обучения модели, декодер будет пытаться из латентного пространства восстановить изображения тех типов, на которых он обучался. Даже если шум подать декодеру, он будет пытаться его приблизить теми данными на которых он обучался.

На рисунке 1 представлена упрощенная архитектура автоэнкодера.

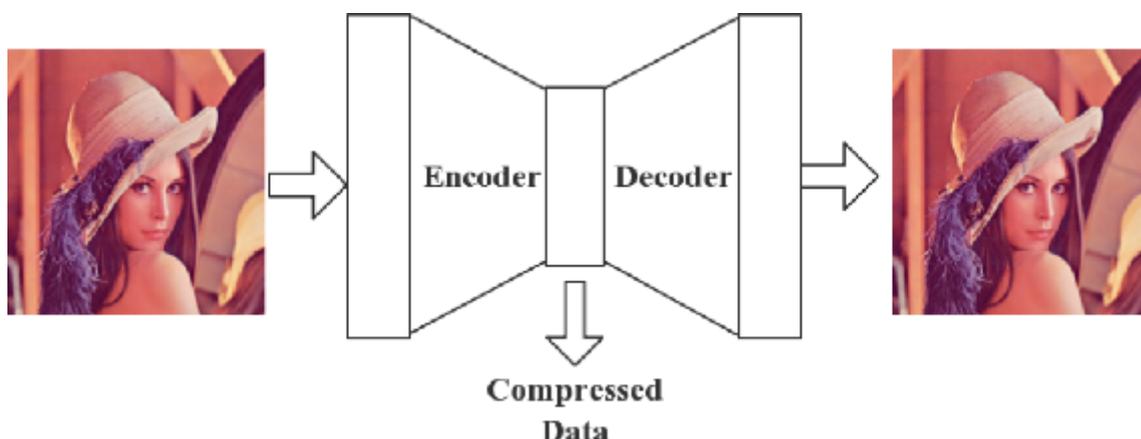


Рисунок 1 – Упрощенная архитектура автоэнкодера

Рассмотрим мои данные. Они представляют родинки. Следовательно, если autoencoder обучить на родинках, то он будет хорошо кодировать и декодировать только родинки. Если на вход подать изображение, не принадлежащее классу родинки, то decoder из латентного пространства будет пытаться восстановить именно родинку, а не то изображение, которое мы подали и, следовательно, разница между входным изображением и выходным будет очень большая, если подать изображение из класса, на котором мы обучали модель, то разность между входным и выходным изображением будет низкая. Если разница очень большая, то это считается аномалией.

Рассмотрим два способа сжатия данных:

- 1 Autoencoder
 - 2 Variational Autoencoder
- ### 1.2 Autoencoder

На рисунке 2 представлена архитектура автоэнкодера.

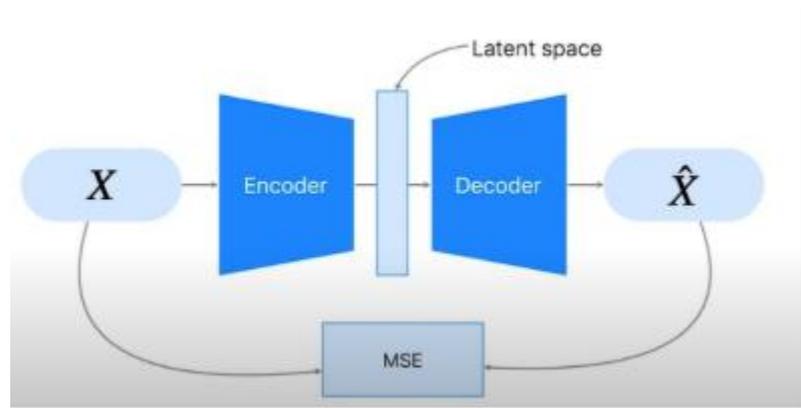


Рисунок 2 – Архитектура автоэнкодера

На вход подается изображение, оно сворачивается в латентное пространство после декодируется.

Рассмотрим случайную величину Z , которое представляет собой сжатое представление изображения.

В качестве функции стоимости мы будем использовать MSE
Функция стоимости MSE определяется по формуле

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (1)$$

где y_i – входное изображение, \hat{y}_i – выходное изображение.

Определив такую функцию стоимости, мы хотим просто расположить схожие изображения рядом друг с другом, поэтому случайная величина Z имеет $\mu \neq 0$ и $\sigma^2 \neq 1$.

На рисунке 3 приведено распределение данных в латентном пространстве при кодировании autoencoder.

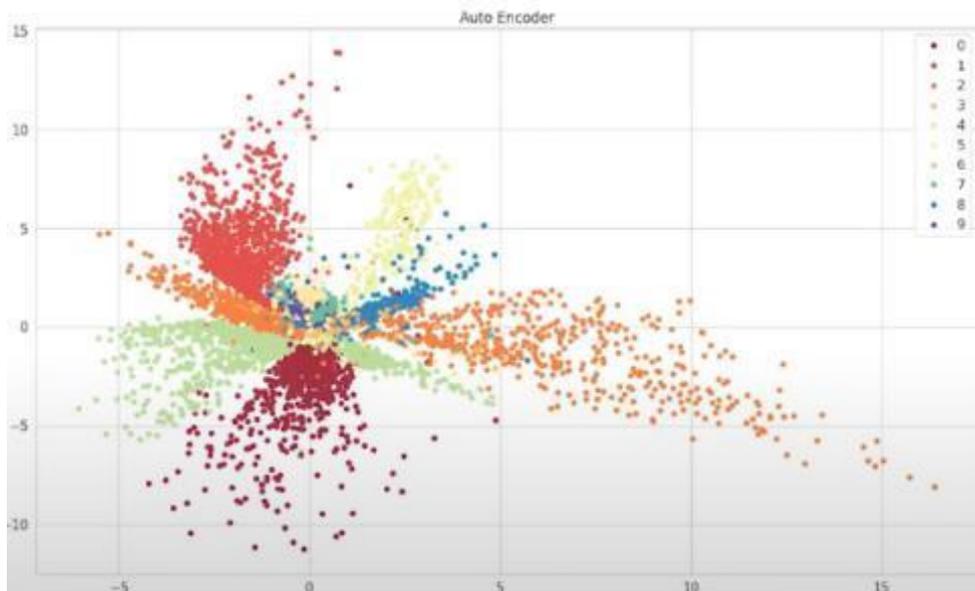


Рисунок 3 – Распределение данным при кодировании autoencoder

В качестве encoder и decoder я взял resnet18.

ResNet-18 — это сверточная нейронная сеть, состоящая из 18 слоев, обученная на более чем миллионе изображений из базы данных ImageNet. Предварительно обученная сеть может классифицировать изображения по 1000 категориям объектов, таким как клавиатура, мышь, карандаш и многие животные. В результате сеть изучила богатые представления функций для широкого спектра изображений. Сеть имеет входной размер изображения 224 на 224 [1].

На рисунке 4 приведена архитектура resnet18.

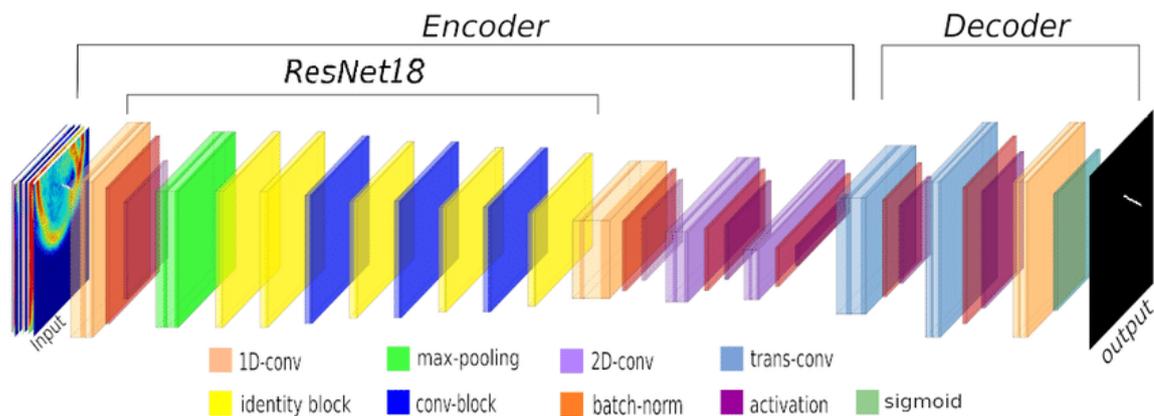


Рисунок 4 – Архитектура resnet18

Задав архитектуру и функцию стоимости можно начать обучение модели. На рисунке 5 приведена ошибка на val по эпохам.

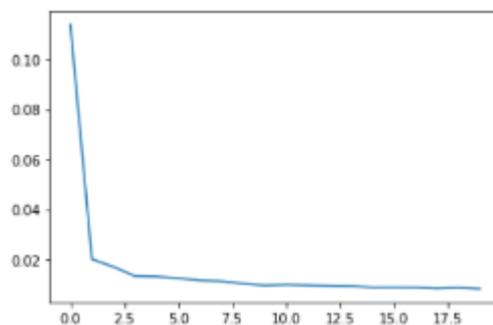


Рисунок 5 – Ошибка на val

Как видно loss на val падает.

Теперь определим порог, который будет определять, является изображение аномалией или нет. Прогоним через сеть все данные и найдем самую большую разницу между входным и выходным изображением. То, что выше порога считается аномалией.

На рисунке 6 представлена гистограмма, представляющая собой все разности.

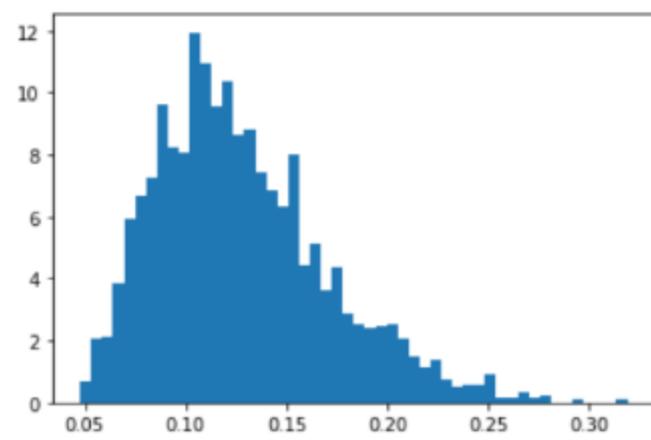


Рисунок 6 – Гистограмма разностей

Порог равен 0.31.

На рисунке 7 приведены результаты работы программы. Сверху картинка написано предсказание, а снизу правильный ответ.

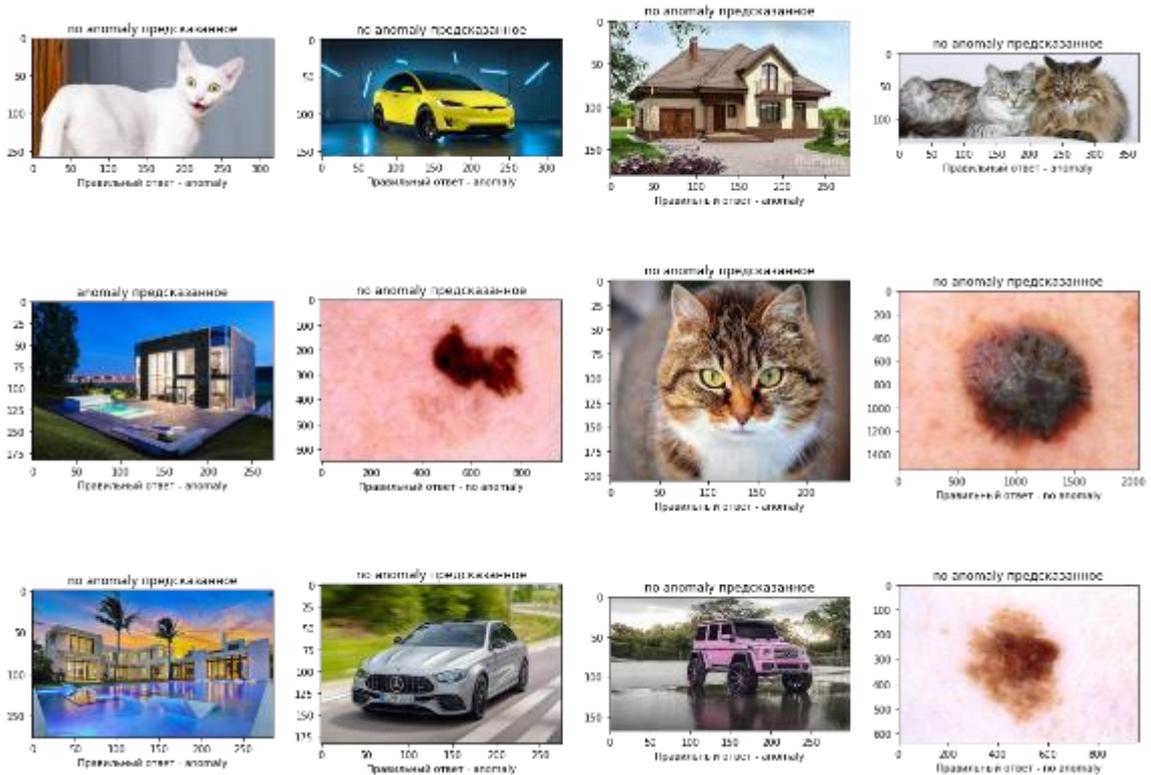


Рисунок 7 – Результаты autoencoder

На рисунке 8 представлены разности между входными и выходными изображениями и метка, является изображение аномалией или нет. Первые три картинки не аномалии, остальные аномалии.

```

'ISIC_0000142.jpg': ['no anomaly', 0.1862474],
'ISIC_0000165.jpg': ['no anomaly', 0.19463664],
'ISIC_0000175.jpg': ['no anomaly', 0.07419035],
'download (10).jpg': ['no anomaly', 0.2004559],
'download (2).jpg': ['anomaly', 0.29417548],
'download (3).jpg': ['no anomaly', 0.20179622],
'download (4).jpg': ['no anomaly', 0.23983575],
'download (5).jpg': ['no anomaly', 0.23260792],
'download (6).jpg': ['no anomaly', 0.21175432],
'download (7).jpg': ['no anomaly', 0.24151534],
'download (8).jpg': ['no anomaly', 0.2134906],
'download (9).jpg': ['no anomaly', 0.19779365]]
    
```

Рисунок 8 – Разницы и метки

Как видно несмотря на то, что autoencoder показал хорошие результаты на val. Аномалии он определяет плохо. Можно заметить, что тае между родинками и аномалиями не сильно отличается. Из 9 аномалий, он правильно определил только одну. Точность меньше 10%

Результаты такие плохие потому что случайная величина в скрытом пространстве не распределена по нормальному закону распределения.

1.3 Variational Autoencoder

В идеале мы хотим, чтобы наше латентное пространство объединяло семантически схожие точки данных рядом друг с другом и располагало семантически несхожие точки далеко друг от друга.

Основная проблема автоэнкодеров в том, что выученные латентные пространства не обязательно обладают ни одним из этих свойств. Поэтому на помощь приходит Variational Autoencoder

На рисунке 9 приведена архитектура variational autoencoder.

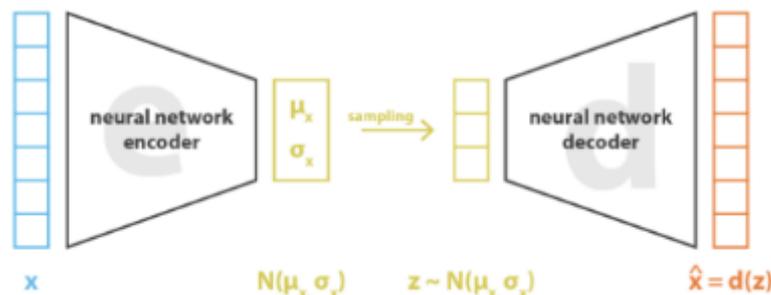


Рисунок 9 – Архитектура variational autoencoder

Здесь идея в том, что encoder генерирует параметры нормального распределения, а уже с помощью этих параметров генерируется сам вектор в латентном пространстве. Тем самым наше изображение в латентном пространстве, которое представляет собой случайную величину Z , будет иметь $\mu = 0$ и $\sigma^2 = 1$.

Для этой нейросети будем использовать такую функцию стоимости:

$$loss = BCEloss(y_i, \hat{y}_i) + KL(p||q), \quad (2)$$

где y_i – входное изображение, \hat{y}_i – выходное изображение, $p = N(\mu, \sigma)$ гауссово распределение с мат ожиданием μ и среднеквадратическим отклонением σ , $q = N(0,1)$ гауссово распределение с $\mu = 0$ и $\sigma = 1$

Формула BCEloss:

$$BCEloss = -\frac{1}{N} \sum_i^N y_i * \log \log (p(\hat{y}_i)) + (1 - y_i) * \log (1 - p(\hat{y}_i)), \quad (3)$$

BCEloss показывает на сколько мы хорошо кодируем изображения.

KL показывает насколько наше выходное распределение отличается от нормального

Формула KL:

$$KL(p||q) = -\sum_{x \in X} p(x) * \log_e \left(\frac{q(x)}{p(x)} \right), \quad (4)$$

Мы хотим оценить насколько наше выходное распределение отличается от нормального.

Пусть p – Гауссово распределение. Формула Гауссова распределение:

$$p = \frac{1}{\sqrt{2 * \pi * \sigma_1^2}} e^{-\frac{(x - \mu_1)^2}{2 \sigma_1^2}}, \quad (5)$$

Пусть q – Гауссово распределение. Формула Гауссова распределение:

$$q = \frac{1}{\sqrt{2 * \pi * \sigma_2^2}} e^{-\frac{(x - \mu_2)^2}{2 \sigma_2^2}}, \quad (6)$$

Формула KL дивергенции между двумя распределениями в общем виде:

$$\begin{aligned}
 KL(p||q) &= -\sum_{x \in X} p(x) * \log_e \left(\frac{q(x)}{p(x)} \right) = -\sum_{x \in X} p(x) * \log_e \left(\left(\frac{\sigma_1^2}{\sigma_2^2} \right)^{\frac{1}{2}} * e^{-\frac{(x-\mu_2)^2 + (x-\mu_1)^2}{2\sigma_2^2 + 2\sigma_1^2}} \right) \\
 &= -\left(\sum_{x \in X} \frac{1}{2} p(x) * \log_e \left(\left(\frac{\sigma_1^2}{\sigma_2^2} \right) \right) + \sum_{x \in X} p(x) * \log_e \left(e^{-\frac{(x-\mu_2)^2 + (x-\mu_1)^2}{2\sigma_2^2 + 2\sigma_1^2}} \right) \right) \\
 &= -\left(\frac{1}{2} * \log_e \left(\left(\frac{\sigma_1^2}{\sigma_2^2} \right) \right) + \sum_{x \in X} p(x) * \left(\frac{(x-\mu_1)^2}{2\sigma_1^2} - \frac{(x-\mu_2)^2}{2\sigma_2^2} \right) \right) \\
 &= -\left(\frac{1}{2} * \log_e \left(\left(\frac{\sigma_1^2}{\sigma_2^2} \right) \right) + \sum_{x \in X} p(x) * \left(\frac{(x-\mu_1)^2}{2\sigma_1^2} \right) - \sum_{x \in X} p(x) * \left(\frac{(x-\mu_2)^2}{2\sigma_2^2} \right) \right) \\
 &= -\left(\frac{1}{2} * \log_e \left(\frac{\sigma_1^2}{\sigma_2^2} \right) + \frac{1}{2} - \frac{1}{2\sigma_2^2} \sum_{x \in X} p(x) * (x-\mu_2)^2 \right) \\
 &= -\left(\frac{1}{2} * \log_e \left(\frac{\sigma_1^2}{\sigma_2^2} \right) + \frac{1}{2} - \frac{1}{2\sigma_2^2} \sum_{x \in X} p(x) * (x-\mu_1 + \mu_1 - \mu_2)^2 \right) \\
 &= -\left(\frac{1}{2} * \log_e \left(\frac{\sigma_1^2}{\sigma_2^2} \right) + \frac{1}{2} \right. \\
 &\quad \left. - \frac{1}{2\sigma_2^2} \left(\sum_{x \in X} p(x) * (x-\mu_1)^2 + (\mu_1 - \mu_2)^2 \sum_{x \in X} p(x) + 2 * (\mu_1 - \mu_2) \sum_{x \in X} p(x) * (x-\mu_1) \right) \right) \\
 &= -\left(\frac{1}{2} * \log_e \left(\frac{\sigma_1^2}{\sigma_2^2} \right) + \frac{1}{2} - \frac{1}{2\sigma_2^2} (\sigma_1^2 + (\mu_1 - \mu_2)^2) \right) \\
 &= -\frac{1}{2} \left(\log_e \left(\frac{\sigma_1^2}{\sigma_2^2} \right) + 1 - \frac{1}{\sigma_2^2} (\sigma_1^2 + (\mu_1 - \mu_2)^2) \right)
 \end{aligned}$$

Конечная формула, которая показывает расстояние между двумя Гауссовыми распределениями:

$$KL(p||q) = -\frac{1}{2} \left(\log_e \left(\frac{\sigma_1^2}{\sigma_2^2} \right) + 1 - \frac{1}{\sigma_2^2} (\sigma_1^2 + (\mu_1 - \mu_2)^2) \right), \quad (7)$$

Пусть q - нормальное распределение. Запись нормального распределения:

$$q = N(0,1), \quad (8)$$

Подставим 8 формулу в 7 формулу и получим формула, которая показывает расстояние между выходным распределением и нормальным:

$$KL(p||q) = -\sum_{x \in X} p(x) * \log_e \left(\frac{q(x)}{p(x)} \right), \quad (9)$$

Мы определили две функции, которые составляют нашу общую функцию стоимости.

В результате использования такой функции, наши сематические схожие данные помимо того, что будут размещаться в латентном пространстве близко друг к другу, они будут распределены нормально, то есть $\mu = 0$ и $\sigma = 1$.

На рисунке 10 приведено распределение данных в латентном пространстве при кодировании variational autoencoder.

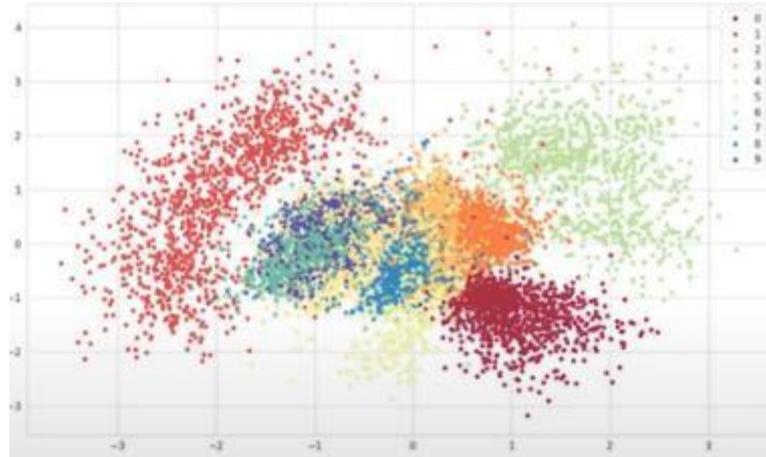


Рисунок 10 – Распределение данным при кодировании variational autoencoder

На рисунке 11 приведена ошибка на val по эпохам.

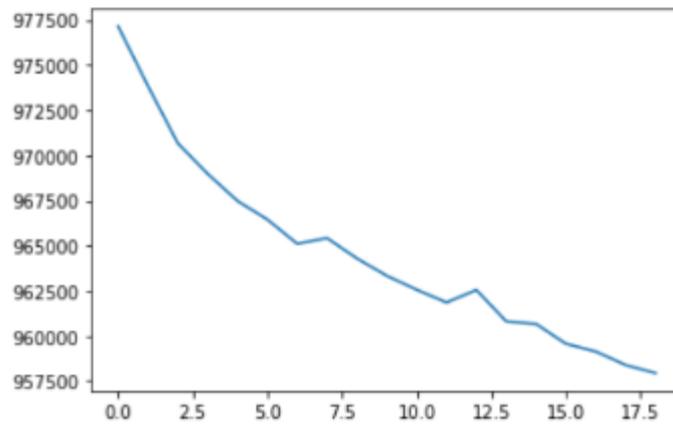


Рисунок 11 – Ошибка на val

Как видим ошибка с каждой эпохой уменьшается.
На рисунке 12 представлена гистограмма, представляющая собой все разности.

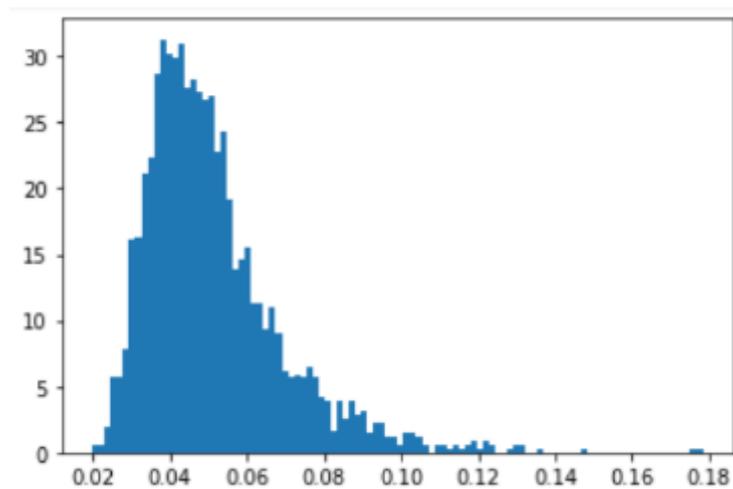


Рисунок 12 – Гистограмма разностей

Порог равен 0.1782.

На рисунке 13 приведены результаты работы программы. Сверху картинки написано предсказание, а снизу правильный ответ.

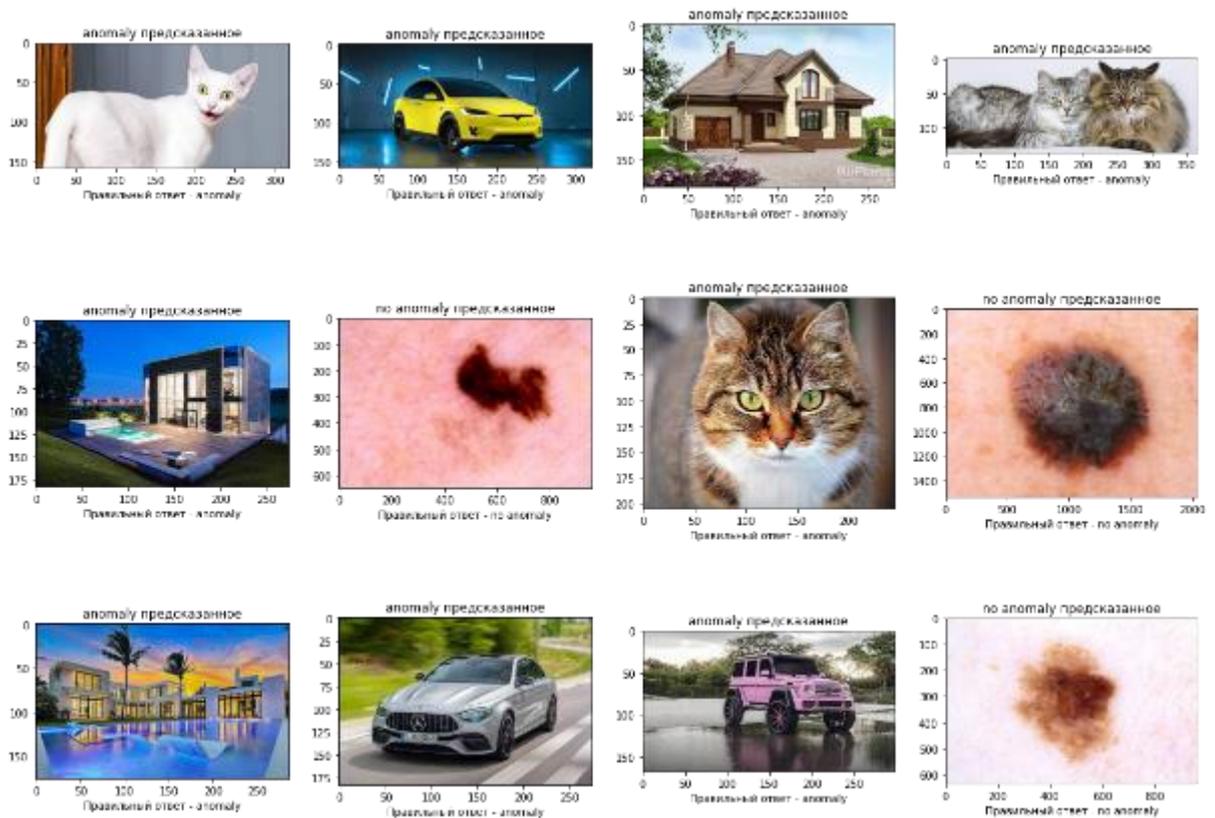


Рисунок 13 – Результаты autoencoder

На рисунке 14 представлены разности между входными и выходными изображениями и метка, является изображением аномалией или нет. Первые три картинки не аномалии, остальные аномалии.

```
{'ISIC_0000142.jpg': ['no anomaly', 0.067003325],
'ISIC_0000165.jpg': ['no anomaly', 0.08873247],
'ISIC_0000175.jpg': ['no anomaly', 0.04902115],
'download (10).jpg': ['anomaly', 0.15084253],
'download (2).jpg': ['anomaly', 0.2628354],
'download (3).jpg': ['anomaly', 0.21156268],
'download (4).jpg': ['anomaly', 0.1980344],
'download (5).jpg': ['anomaly', 0.22648849],
'download (6).jpg': ['anomaly', 0.2169738],
'download (7).jpg': ['anomaly', 0.21334277],
'download (8).jpg': ['anomaly', 0.18039457],
'download (9).jpg': ['anomaly', 0.20018359]}
```

Рисунок 14 – Разницы и метки

Вае хорошо справился с проблемой поиска аномалий. Можно заметить, что тае между родинками низкое, а между аномалиями большое. Ошибок нет. Все работает хорошо.

1.4 Вывод

Для определения аномалий при работе с родинками лучше использовать VAE. Он хорошо представляет изображения родинок в латентном пространстве: изображения похожих родинок располагаются близко друг к другу, также векторы в латентном пространстве распределены нормально. Разность между данными класса на которых обучался низкая, а тех на которых не обучался, высокая.

2 КЛАССИФИКАЦИЯ РОДИНОК

Проблемы:

Родинки друг на друга очень похожи, поэтому обычными нейросетями данную проблему не решить, также в нашем распоряжении маленький датасет.

Гипотеза:

Существует ли архитектура нейросети, которая решает поставленную задачу с хорошим значением точности.

2.1 Анализ входных данных

Данный набор состоит из 2357 изображений злокачественных и доброкачественных онкологических заболеваний, которые были сформированы из Международного Сотрудничества в Области Визуализации Кожи (ISIC). Все изображения были отсортированы в соответствие с классификацией, взятой с ISIC, и все поднаборы были разделены на одинаковое количество изображений, за исключением меланом и родинок, чьи изображения немного преобладают.

Набор данных содержит следующие типы родинок:

- 1 Actinic keratosis;
- 2 Basal cell carcinoma;
- 3 Dermatofibroma;
- 4 Melanoma;
- 5 Nevus;
- 6 Pigmented benign keratosis;
- 7 Seborrheic keratosis;
- 8 Squamous cell carcinoma;
- 9 Vascular lesion;

На рисунке 15 и 16 приведены распределения классов тренировочных данных.

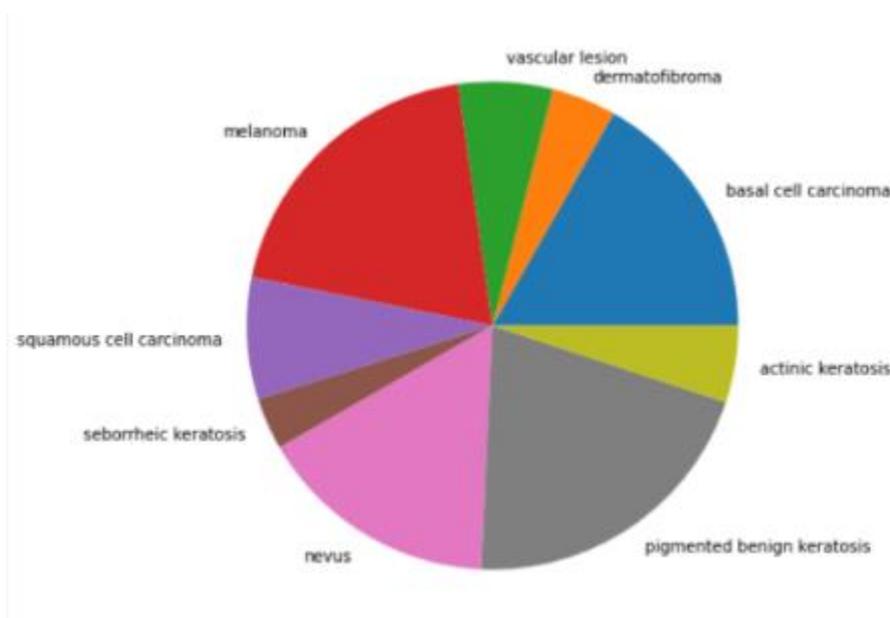


Рисунок 15 – Распределение классов тренировочных данных

```
{'actinic keratosis': 114,  
'basal cell carcinoma': 376,  
'dermatofibroma': 95,  
'melanoma': 438,  
'nevus': 357,  
'pigmented benign keratosis': 462,  
'seborrheic keratosis': 77,  
'squamous cell carcinoma': 181,  
'vascular lesion': 139}
```

Рисунок 16 – Распределение классов тренировочных данных

Как видно данные распределены неравномерно.
На рисунке 17 и 18 приведены распределения классов тестовых данных.

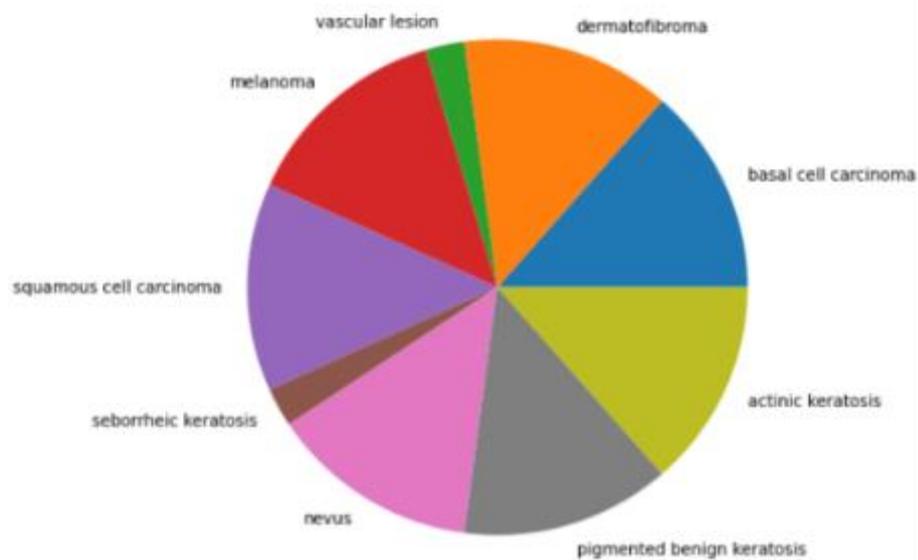


Рисунок 17 – Распределение классов тренировочных данных

```
{'actinic keratosis': 16,  
'basal cell carcinoma': 16,  
'dermatofibroma': 16,  
'melanoma': 16,  
'nevus': 16,  
'pigmented benign keratosis': 16,  
'seborrheic keratosis': 3,  
'squamous cell carcinoma': 16,  
'vascular lesion': 3}
```

Рисунок 18 – Распределение классов тренировочных данных

Как видно данные распределены относительно равномерно.

Данных у нас очень мало, поэтому искусственно увеличим их. Будем использовать аугментацию.

Идея аугментации:

На вход сети подается изображение после к ней последовательно применяются трансформации с разной вероятностью, тем самым нейросеть будет обучаться на разных данных. Трансформации для train:

```
transform_train=A.Compose([
    A.Resize(256,256),
    A.HorizontalFlip(p=0.5),
    A.Rotate(limit=90,p=0.7),
    A.GaussNoise(p=0.5),
    A.RGBShift(p=0.5),
    Normalize(mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225))
])
```

В начале уменьшаем размер изображения, потом зеркально отражаем изображение с вероятностью 0.5, после поворачиваем изображение от 0 до 90 градусов с вероятностью 0.7, добавляем гауссов шум вероятностью 0.5 и меняем каналы вероятностью 0.5. В самом конце нормализуем данные. Такие трансформации, кроме Resize и Normalize, применяются только к train, потому что мы не хотим повредить тестовые данные.

Трансформации для test:

```
transform_val=A.Compose([
    A.Resize(256,256),
    Normalize(mean=(0.485, 0.456, 0.406), std=(0.229, 0.224, 0.225))
])
```

Тестовые данных мы только уменьшаем размер изображения и нормализуем.

Пример аугментации. На рисунке 19 представлена одна и та же родинка, но после применения к ней трансформаций они все выглядят немного по-разному.

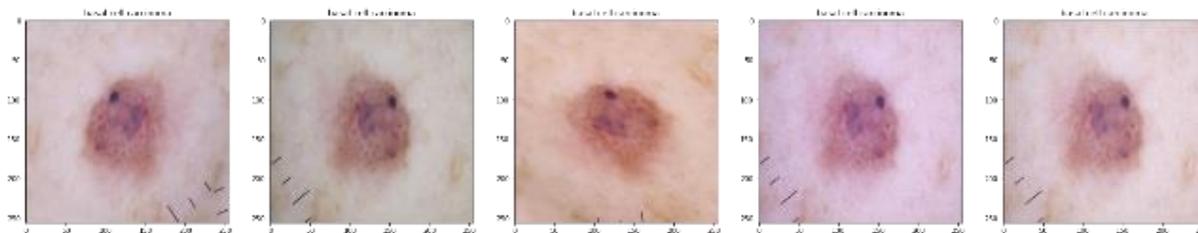


Рисунок 19 – Пример аугментации

В качестве подхода к построению и использованию нейросетей я буду использовать фантьюнинг крупных сверточных нейросетей.

Суть фантьюнига:

Замораживаются верхние (первые) слои сети, нижние до обучаются, так как первые слои выделяют низкоуровневую информацию, а последние выделяют информацию под мою конкретную задачу.

2.2 VGG-16 с заморозкой промежуточных слоев

На рисунке 20 приведена архитектура VGG-16.

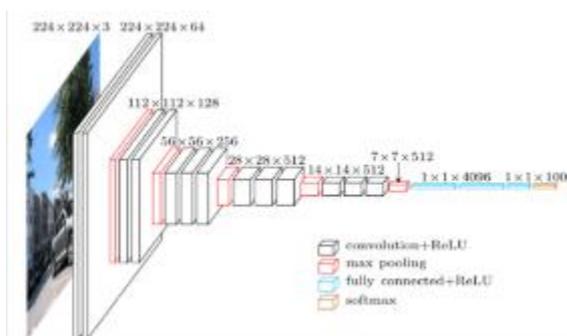


Рисунок 20 – Архитектура VGG-16

VGG-16 была выбрана, потому что она может обеспечить точность тестирования 92,7% в ImageNet, наборе данных, содержащем более 14 миллионов обучающих изображений для 1000 классов объектов. Это одна из топовых моделей конкурса ILSVRC-2014. VGG16 улучшает AlexNet и заменяет большие фильтры последовательностями меньших фильтров 3×3 [2].

Результаты тренировки.

В качестве метрик качества в данной работе используются precision, recall, f1-score, accuracy, macro avg, weighted avg.

На 10 формуле приведена формула precision:

$$precision = \frac{TP}{TP+FP}, \quad (10)$$

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными.

На 11 формуле приведена формула recall:

$$recall = \frac{TP}{TP+FN}, \quad (11)$$

Recall показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм.

На 12 формуле приведена формула f1-score:

$$f1score = (1 + \beta^2) * \frac{precision*recall}{(\beta^2*precision)+recall}, \quad (12)$$

При $\beta^2 = 1$ формула превращается в среднее гармоническое.

На 13 формуле приведена формула accuracy:

$$accuracy = \frac{labels_true_predicted}{all_labels}, \quad (13)$$

где *labels_true_predicted* – количество правильно предсказанных меток, *all_labels* – количество всех меток.

На 14 формуле приведена формула macro avg:

$$accuracy = \frac{1}{N} \sum f1scores, \quad (14)$$

где *N* – количество классов, $\sum f1scores$ – сумма f1-score всех классов.

На 15 формуле приведена формула weighted avg:

$$weighted\ avg = \frac{1}{N} \sum n_i f1scores_i, \quad (15)$$

где *N* – количество классов, *n_i* – количество *i*-ых классов, *f1scores_i* - f1-score *i*-ого класса.

Обучение происходило в штатном режиме. На рисунке 21 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

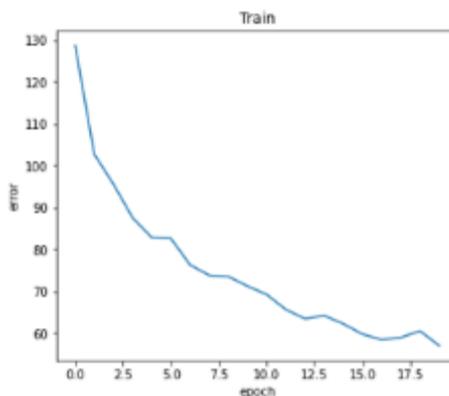


Рисунок 21 – Ошибка VGG-16 на тренировочных данных

На рисунке 22 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

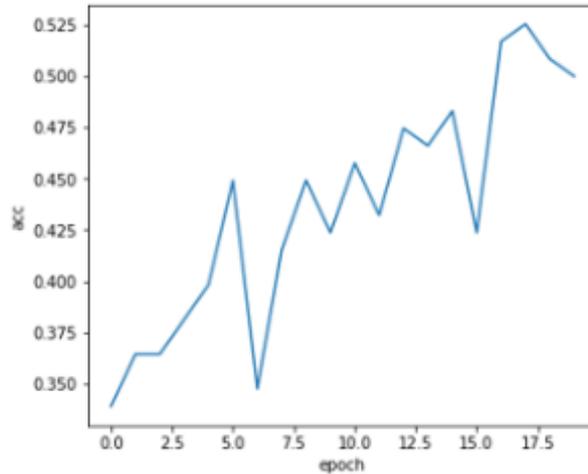


Рисунок 22 – Точность VGG-16 на тестовых данных

На рисунке 23 представлены результаты VGG-16.

	precision	recall	f1-score	support
0	0.70	0.44	0.54	16
1	0.63	0.75	0.69	16
2	1.00	0.38	0.55	16
3	0.10	0.06	0.08	16
4	0.44	0.88	0.58	16
5	0.48	0.88	0.62	16
6	0.00	0.00	0.00	3
7	0.62	0.31	0.42	16
8	0.75	1.00	0.86	3
accuracy			0.53	118
macro avg	0.53	0.52	0.48	118
weighted avg	0.56	0.53	0.49	118

Рисунок 23 – Результаты VGG-16

Результаты получились не очень хорошие. Мы имеем низкие macro-avg, weighted avg, accuracy

На рисунках 24 и 25, приведены результаты работы сверточной нейросети VGG-16. Сверху выводится пересказанное значение, а снизу правильное.

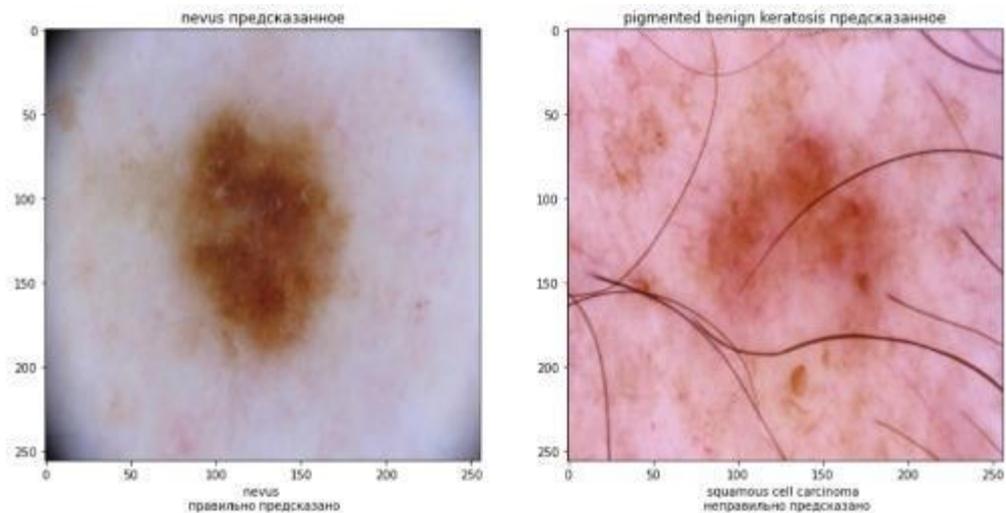


Рисунок 24 – Результаты VGG-16

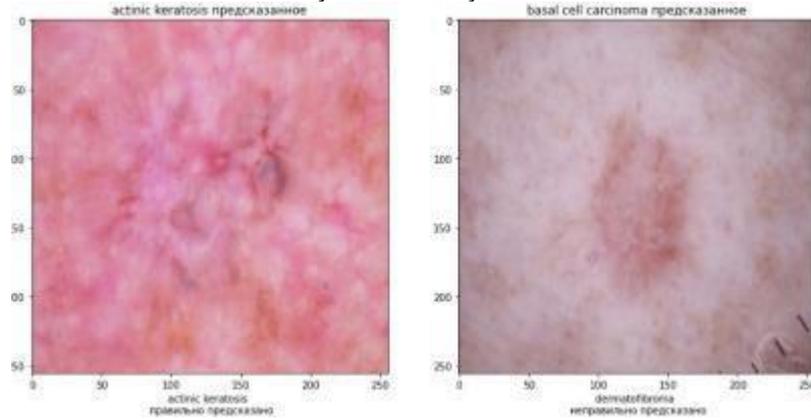


Рисунок 25 – Результаты VGG-16

2.3 VGG-16, без заморозки промежуточных слоев

Можно предположить, что такие не очень хорошие результаты могли быть из-за того, что были заморожены слои свертков, как известно на первых слоях, происходит выделение общих признаков, но это хорошо работает, когда мы работаем с обычными объектами, а не родинками. Родинки похожи одна на другую, поэтому нужно разморозить начальные слои для того, чтобы выделить признаки, которые соответствуют различным типам родинок.

Обучение происходило в штатном режиме. На рисунке 26 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

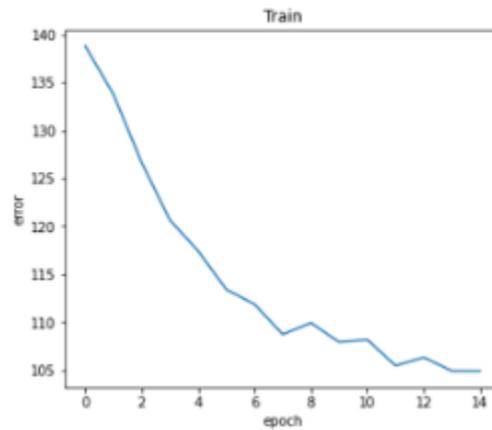


Рисунок 26 – Ошибка VGG-16 без заморозки промежуточных слоев на тренировочных данных

На рисунке 27 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

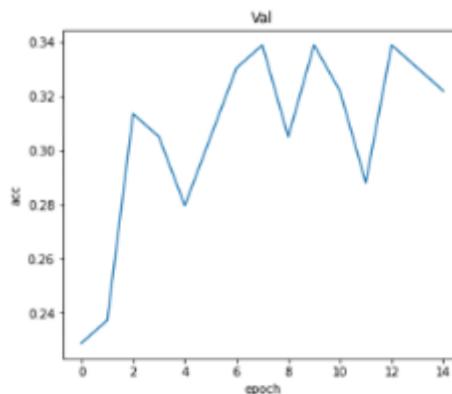


Рисунок 27 – Точность VGG-16 без заморозки промежуточных слоев на тестовых данных

На рисунке 28 представлены результаты VGG-16 без заморозки промежуточных слоев.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	16
1	0.31	0.56	0.40	16
2	0.00	0.00	0.00	16
3	0.11	0.06	0.08	16
4	0.50	0.94	0.65	16
5	0.28	0.81	0.42	16
6	0.00	0.00	0.00	3
7	0.00	0.00	0.00	16
8	0.50	0.67	0.57	3
accuracy			0.34	118
macro avg	0.19	0.34	0.24	118
weighted avg	0.18	0.34	0.22	118

Рисунок 28 – Результаты VGG-16 без заморозки промежуточных слоев

Как видно из результатов, моя гипотеза не подтвердилась и если не замораживать веса, то результаты получатся еще хуже.

На рисунках 29 и 30, приведены результаты работы сверточной нейросети VGG-16 без заморозки промежуточных слоев. Сверху выводиться пересказанное значение, а снизу правильное.

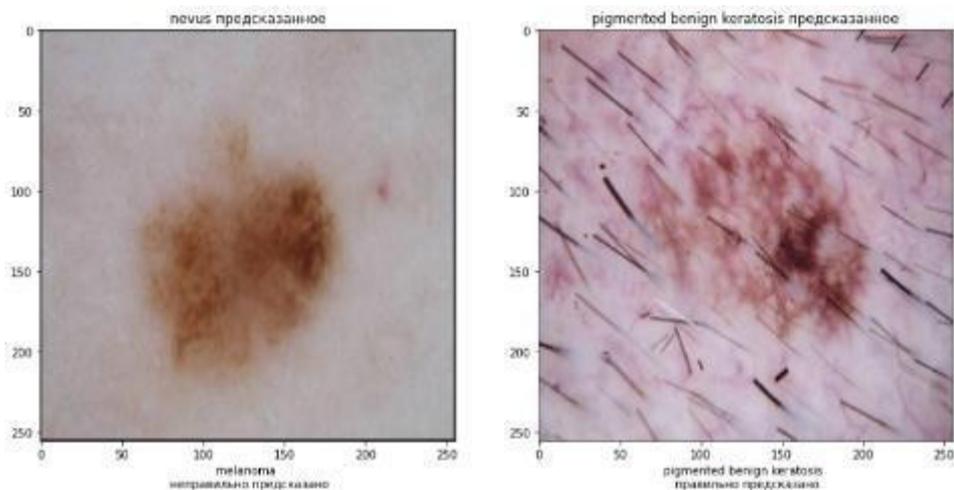


Рисунок 29 – Результаты VGG-16 без заморозки промежуточных слоев

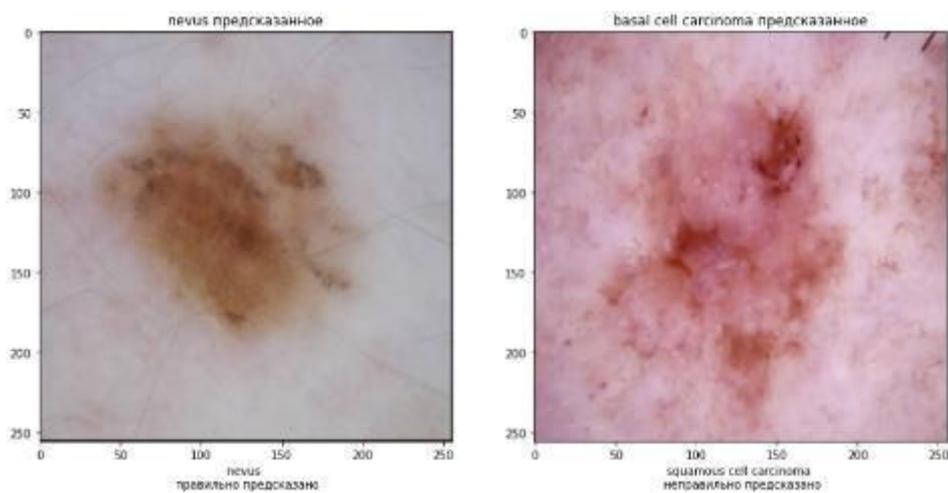


Рисунок 30 – Результаты VGG-16 без заморозки промежуточных слоев

2.4 WIDE_RESNET50_2 с коротким fc

На рисунке 31 приведена архитектура residual network, также приведена обычная сеть. Рисунок показывает различие между архитектурами. Преимуществом такой модели по сравнению с другими является наличие residual block. Сеть, которая состоит из Res Blocks, хороша тем, что она решает проблему с затухающими градиентами. То есть, если сеть очень глубокая, то чтобы найти производную, которая находится где-нибудь в начале, по выходу, нужно сделать очень много перемножений и тем самым эти производные могут быть равны 0, тем градиент не доходит до первых слоев, а с помощью res блоков градиенты будут течь еще и через первые слои и тем самым можно формировать очень глубокие сети.



Рисунок 31 – Архитектура WIDE_RESNET

Обучение происходило в штатном режиме. На рисунке 32 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

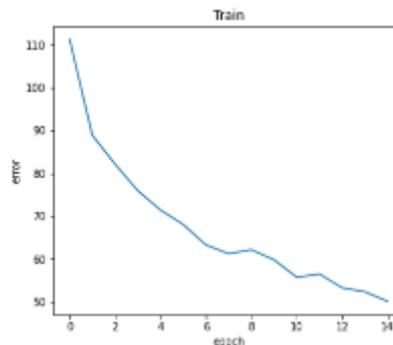


Рисунок 32 – Ошибка WIDE_RESNET50_2

На рисунке 33 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

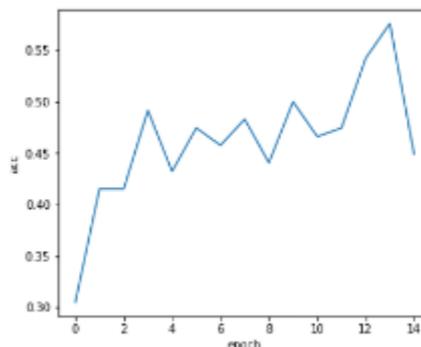


Рисунок 33 – Точность WIDE_RESNET50_2

На рисунке 34 представлены результаты WIDE_RESNET50_2.

	precision	recall	f1-score	support
0	1.00	0.44	0.61	16
1	0.71	0.75	0.73	16
2	1.00	0.50	0.67	16
3	0.20	0.06	0.10	16
4	0.36	1.00	0.53	16
5	0.60	0.94	0.73	16
6	0.00	0.00	0.00	3
7	0.67	0.38	0.48	16
8	1.00	1.00	1.00	3
accuracy			0.58	118
macro avg	0.62	0.56	0.54	118
weighted avg	0.64	0.58	0.55	118

Рисунок 34 – Результаты WIDE_RESNET50_2

Результаты получились лучше, чем при использовании vgg16. Accuracy выше на 5%, macro avg на 6%, weighted avg на 6%.

На рисунках 35 и 36, приведены результаты работы WIDE_RESNET50_2. Сверху выводиться пересказанное значение, а снизу правильное.

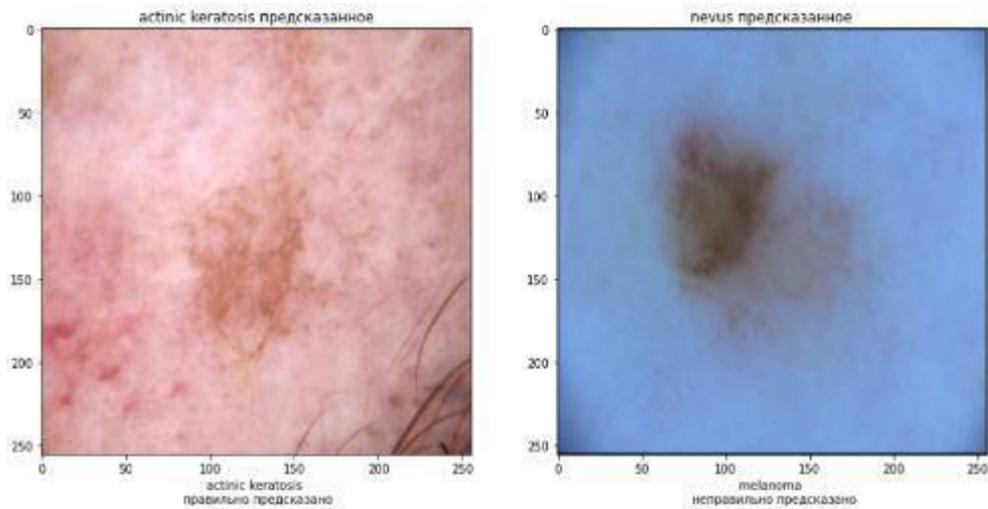


Рисунок 35 – Результаты WIDE_RESNET50_2

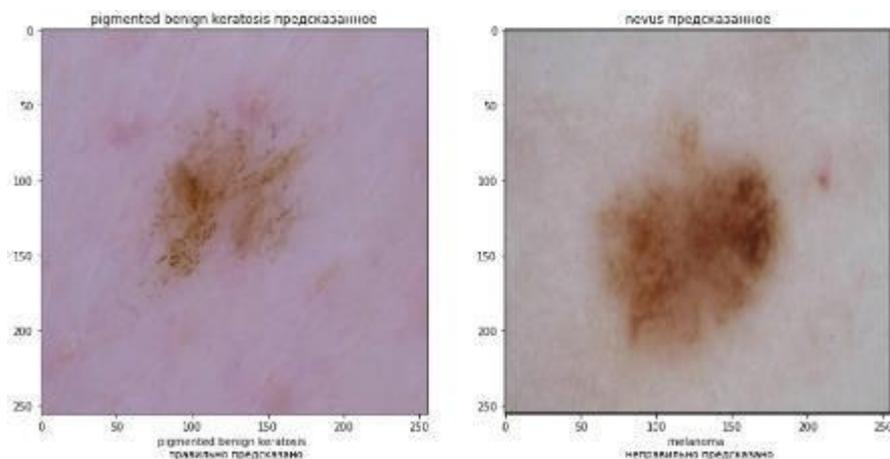


Рисунок 36 – Результаты WIDE_RESNET50_2

2.5 WIDE_RESNET50_2 с длинный fc

В качестве следующей модели возьмем архитектуру WIDE_RESNET50_2, но вместо короткого fc, будем использовать длинный fc.

Обучение происходило в штатном режиме. На рисунке 37 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

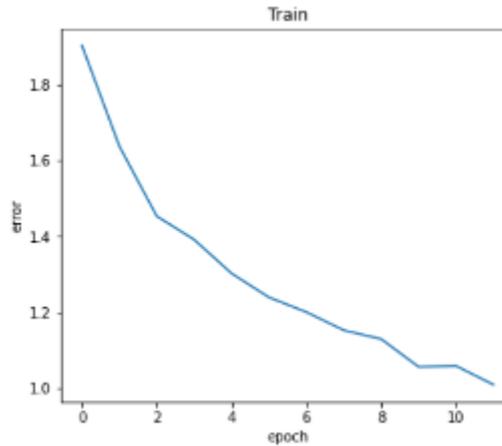


Рисунок 37 – Ошибка WIDE_RESNET50_2 с длинный fc

На рисунке 38 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

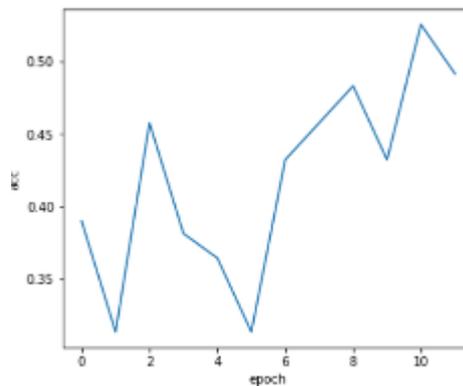


Рисунок 38 – Точность WIDE_RESNET50_2 с длинный fc

На рисунке 39 представлены результаты WIDE_RESNET50_2 с длинный fc.

	precision	recall	f1-score	support
0	1.00	0.25	0.40	16
1	0.80	0.75	0.77	16
2	0.70	0.44	0.54	16
3	0.07	0.06	0.06	16
4	0.48	0.88	0.62	16
5	0.52	0.88	0.65	16
6	0.00	0.00	0.00	3
7	0.50	0.44	0.47	16
8	0.75	1.00	0.86	3
accuracy			0.53	118
macro avg	0.54	0.52	0.49	118
weighted avg	0.57	0.53	0.50	118

Рисунок 39 – Результаты WIDE_RESNET50_2 с длинный fc

Результаты получились хуже, чем при использовании маленького fc. Ассигасы ниже на 5%, масго avg ниже на 5%, weighted avg ниже на 5%.

На рисунках 40 и 41, приведены результаты работы WIDE_RESNET50_2 с длинный fc. Сверху выводиться пересказанное значение, а снизу правильное.

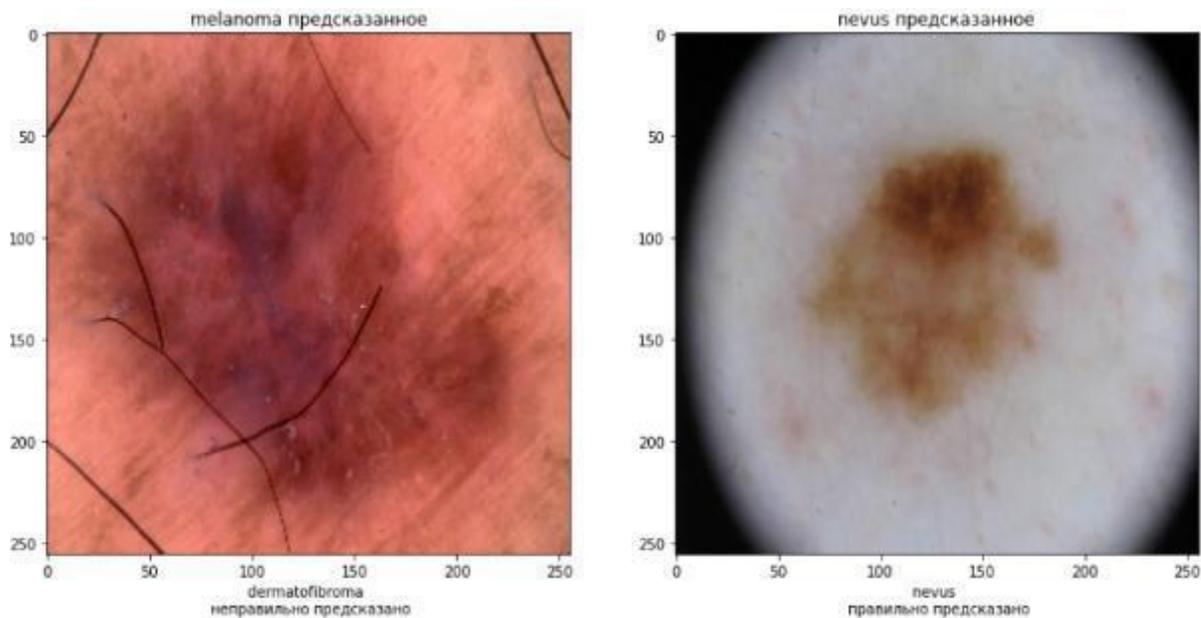


Рисунок 40 – Результаты WIDE_RESNET50_2 с длинный fc

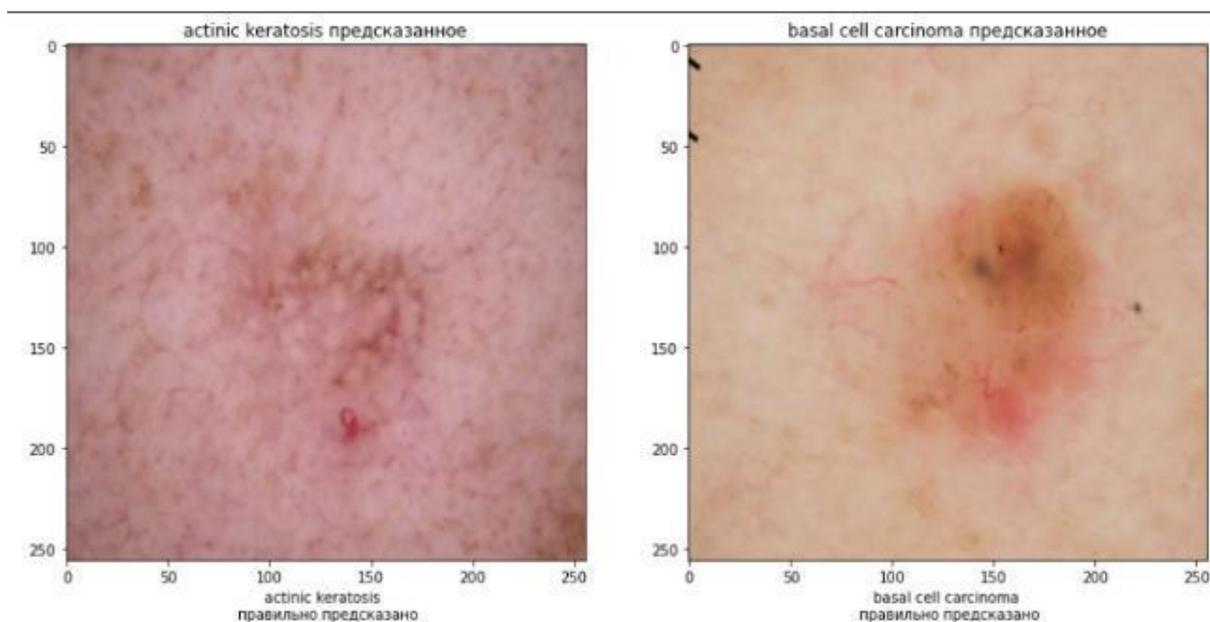


Рисунок 41 – Результаты WIDE_RESNET50_2 с длинный fc

2.6 EfficientNet_V2_L без заморозки промежуточных слоев

В целом модели EfficientNet обеспечивают как более высокую точность, так и лучшую эффективность по сравнению с существующими CNN, уменьшая размер параметров сверточной сети на порядок [3].

На рисунке 42 приведена архитектура EfficientNet_V2_L.

Stage	Operator	Stride	#Channels	#Layers
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	272	15
7	Conv1x1 & Pooling & FC	-	1792	1

Рисунок 42 – Архитектура EfficientNet_V2_L

На рисунке 43 приведены блоки, которые формируют EfficientNet_V2_L.

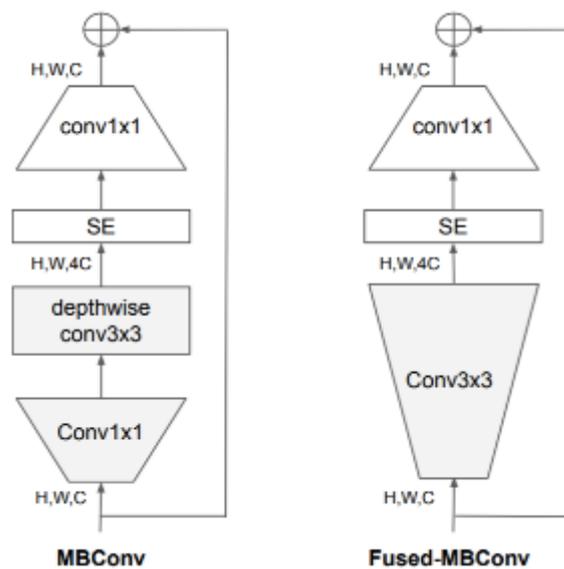


Рисунок 43 – Блоки MBConv и Fused-MBConv

Обучение происходило в штатном режиме. На рисунке 44 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

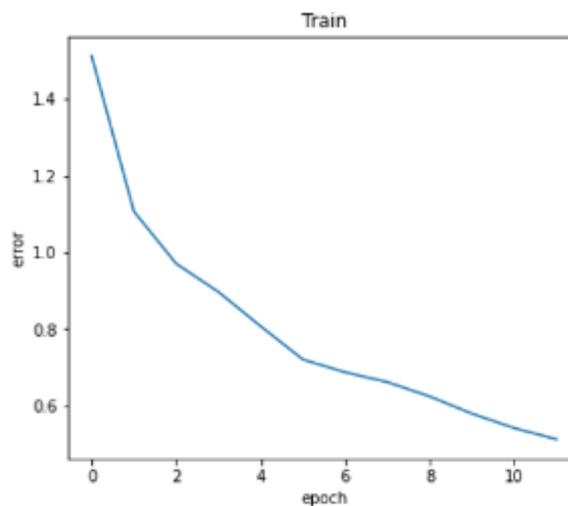


Рисунок 44 – Ошибка EfficientNet_V2_L

На рисунке 45 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

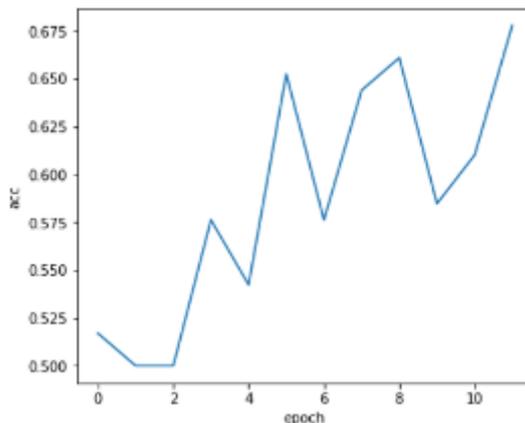


Рисунок 45 – Точность EfficientNet_V2_L

На рисунке 46 представлены результаты EfficientNet_V2_L .

	precision	recall	f1-score	support
0	0.91	0.62	0.74	16
1	0.74	0.88	0.80	16
2	0.91	0.62	0.74	16
3	0.33	0.38	0.35	16
4	0.50	0.94	0.65	16
5	0.93	0.81	0.87	16
6	0.00	0.00	0.00	3
7	0.82	0.56	0.67	16
8	0.75	1.00	0.86	3
accuracy			0.68	118
macro avg	0.65	0.65	0.63	118
weighted avg	0.72	0.68	0.68	118

Рисунок 46 – Результаты EfficientNet_V2_L

Результаты получились лучше, чем при использовании wide resnet50_2. Accuracy выше на 10%, macro avg на 9%, weighted avg на 13%.

На рисунках 47 и 48, приведены результаты работы EfficientNet_V2_L. Сверху выводиться пересказанное значение, а снизу правильное.

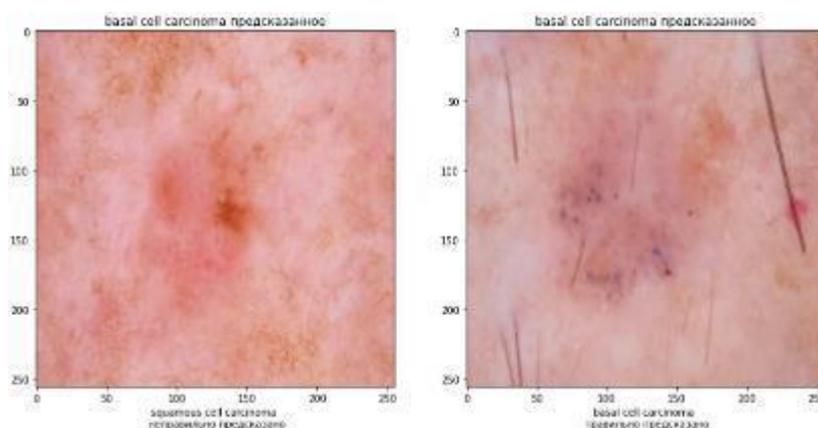


Рисунок 47 – Результаты EfficientNet_V2_L

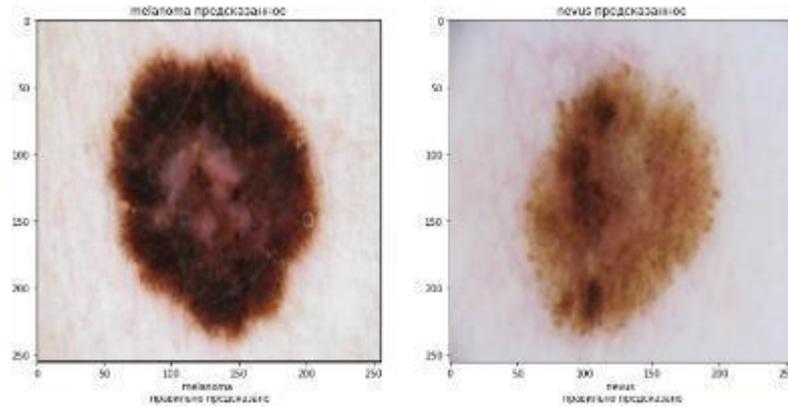


Рисунок 48 – Результаты EfficientNet_V2_L

Результаты получились самые лучшие.

2.7 EfficientNet_V2_L с заморозкой промежуточных слоев

В предыдущем примере после заморозки промежуточных слоев точность улучшилось. Заморозим слои и проверим увеличиться ли точность.

Обучение происходило в штатном режиме. На рисунке 49 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

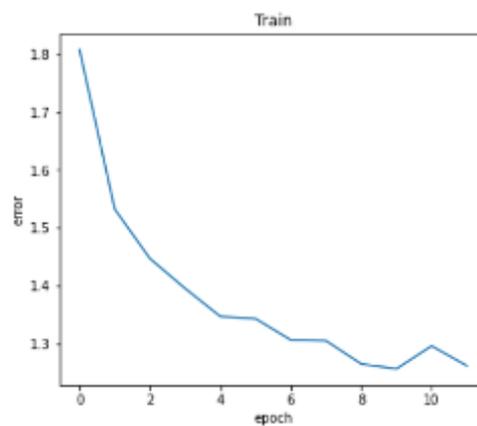


Рисунок 49 – Ошибка EfficientNet_V2_L с заморозкой промежуточных слоев

На рисунке 50 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

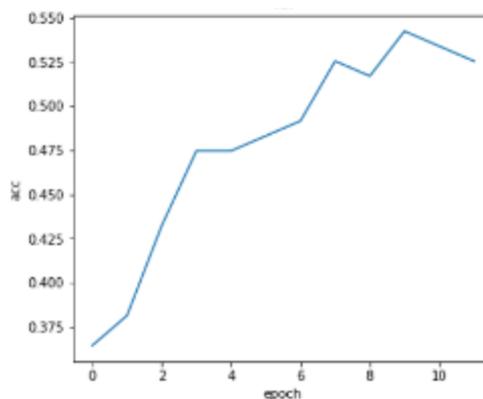


Рисунок 50 – Точность EfficientNet_V2_L с заморозкой промежуточных слоев

На рисунке 51 представлены результаты EfficientNet_V2_L с заморозкой промежуточных слоев.

	precision	recall	f1-score	support
0	1.00	0.25	0.40	16
1	0.65	0.69	0.67	16
2	0.88	0.44	0.58	16
3	0.29	0.25	0.27	16
4	0.48	1.00	0.65	16
5	0.48	0.81	0.60	16
6	0.00	0.00	0.00	3
7	0.60	0.38	0.46	16
8	0.60	1.00	0.75	3
accuracy			0.54	118
macro avg	0.55	0.53	0.49	118
weighted avg	0.61	0.54	0.51	118

Рисунок 51 – Результаты EfficientNet_V2_L с заморозкой промежуточных слоев

После заморозки промежуточных слоев результаты стали хуже. Ассигасу ниже на 14%, масго avg на 14%, weighted avg на 17%. Гипотеза не подтвердилась.

На рисунках 52 и 53, приведены результаты работы EfficientNet_V2_L с заморозкой промежуточных слоев. Сверху выводиться пересказанное значение, а снизу правильное.

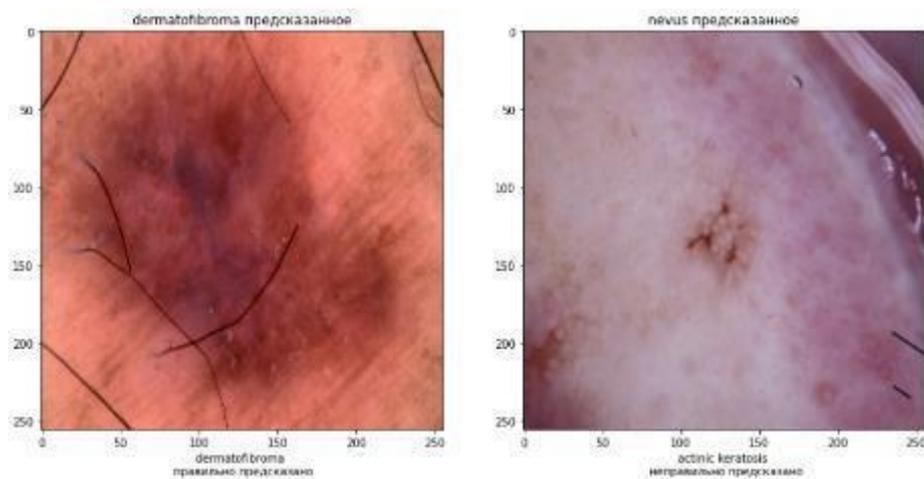


Рисунок 52 – Результаты EfficientNet_V2_L с заморозкой промежуточных слоев

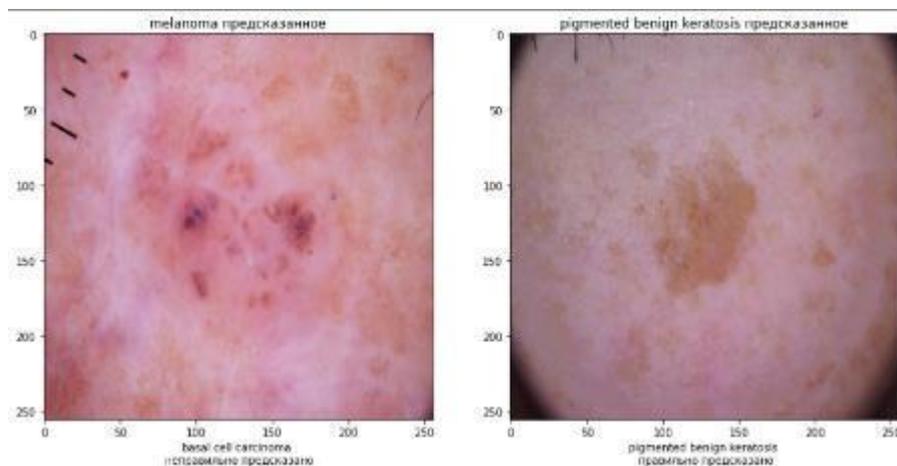


Рисунок 53 – Результаты EfficientNet_V2_L с заморозкой промежуточных слоев

2.8 MOBILENET_V3_SMALL

Специально выбрал маленькую сеть, чтобы она в будущем выступала как модель ученика. На рисунке 54 приведена архитектура MOBILENET:

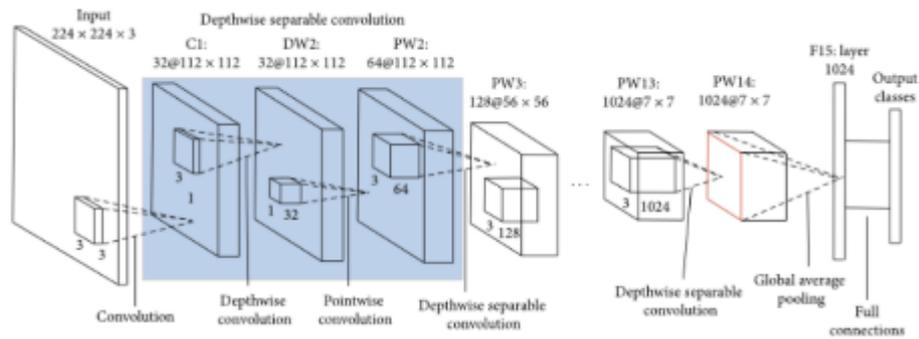


Рисунок 54 – Архитектура MOBILENET

Обучение происходило в штатном режиме. На рисунке 54 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

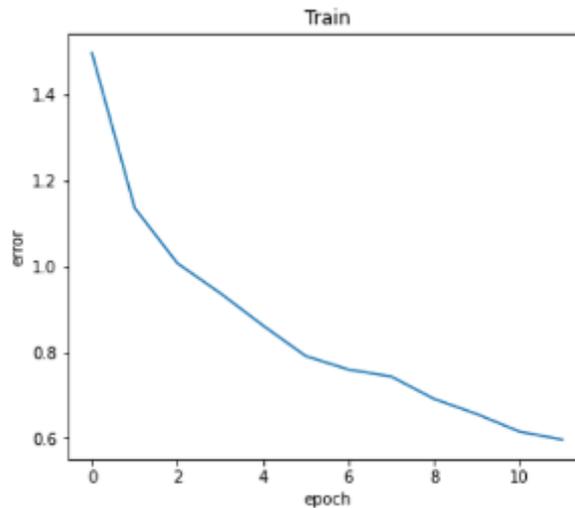


Рисунок 54 – Ошибка MOBILENET

На рисунке 55 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

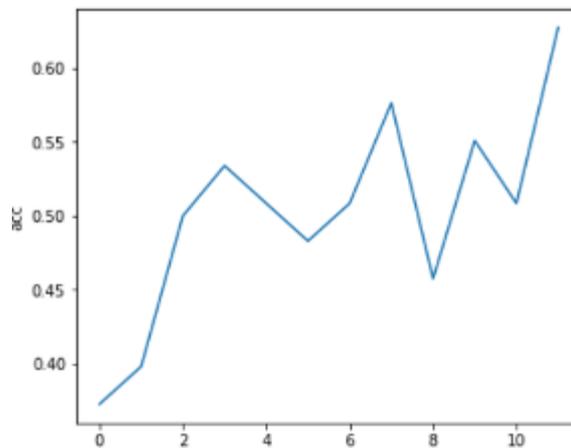


Рисунок 55 – Точность MOBILENET

На рисунке 56 представлены результаты MOBILENET.

	precision	recall	f1-score	support
0	0.65	0.69	0.67	16
1	0.92	0.69	0.79	16
2	0.88	0.44	0.58	16
3	0.33	0.44	0.38	16
4	0.50	0.94	0.65	16
5	0.73	0.69	0.71	16
6	0.00	0.00	0.00	3
7	0.75	0.56	0.64	16
8	1.00	1.00	1.00	3
accuracy			0.63	118
macro avg	0.64	0.60	0.60	118
weighted avg	0.67	0.63	0.62	118

Рисунок 56 – Результаты MOBILENET

Результаты чуть хуже, чем у EfficientNet. Accuracy ниже на 5%, macro avg на 5%, weighted avg на 6%.

На рисунках 57 и 58, приведены результаты работы MOBILENET. Сверху выводиться пересказанное значение, а снизу правильное.

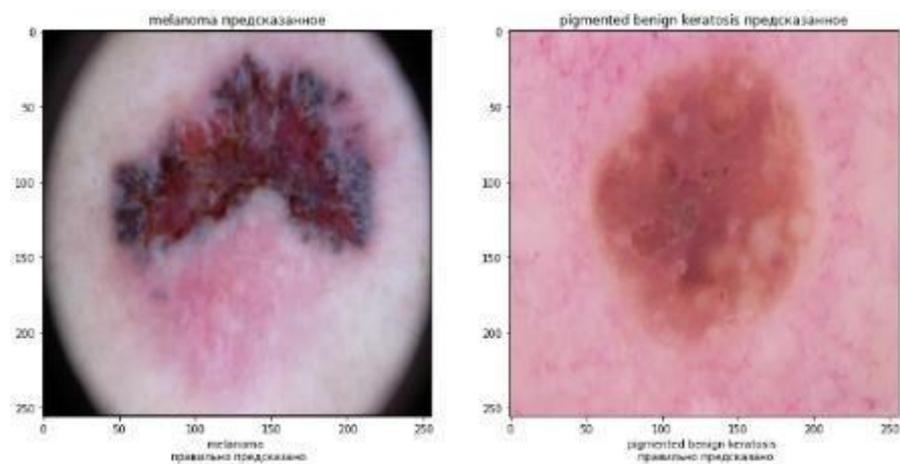


Рисунок 57 – Результаты MOBILENET

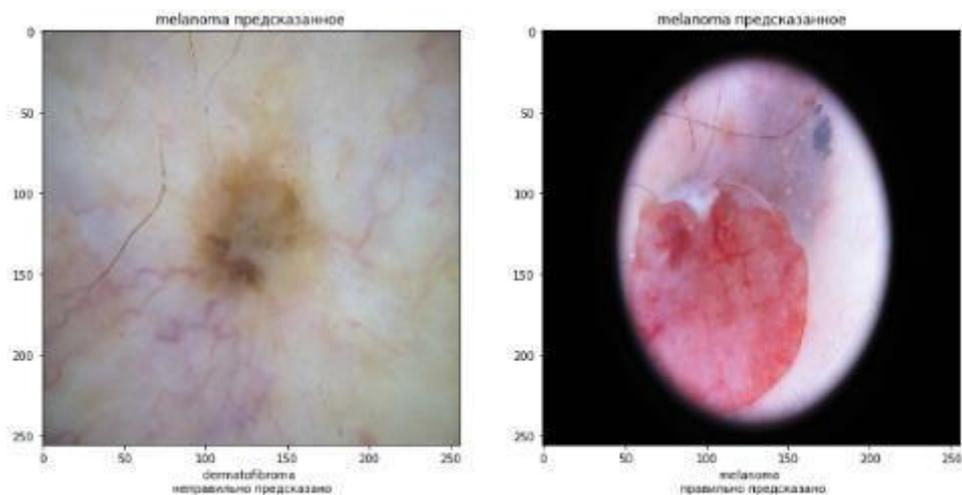


Рисунок 58 – Результаты MOBILENET

2.9 Дистилляция

В реальности очень большие модели, тяжело интегрировать в продакшен, одна из причин — это огромный вес модели. Поэтому можно использовать подход, благодаря которому маленькая модель может получить похожие результаты, как и большая или даже лучше. В начале мы обучаем большую модель на основе наших данных, потом тренируем маленькую модель на основе тех же данных в связке с этой большой сетью и интегрируем в продакшен маленькую модель. В качестве учителя будет выступать EfficientNet_V2_L, а в качестве ученика MOBILENET_V3_SMALL. Размер модели ученика в 5 раз меньше модели учителя, а точность хуже на 5%. На рисунке 59 приведена архитектура дистилляции.

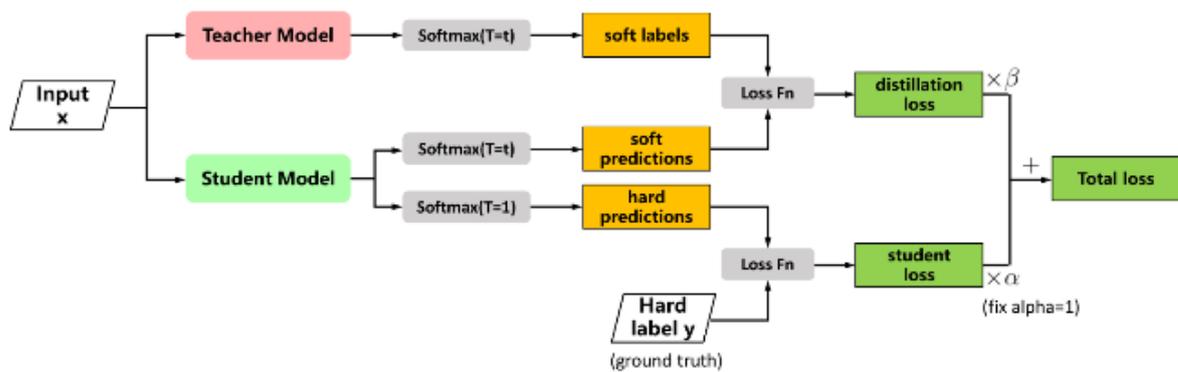


Рисунок 59 – Архитектура дистилляции

Как показано на схеме функция стоимости состоит из суммы СЕ и дивергенции Кульбака-Лейблера умноженные на свои коэффициенты. В СЕ в качестве правильных результатов hard label, то есть one-hot-vector, а в KL подается в качестве правильных результатов soft label, распределение вероятности по классам, модели учителя.

На 16 формуле приведен не обычный softmax, а с сглаживающим параметром:

$$softmax = \frac{e^{\frac{y_i}{T}}}{\sum_{k=1}^n e^{\frac{y_k}{T}}}, \quad (16)$$

где T – температура софтмакса, чем больше, тем более сглаженное распределение. Гиперпараметр, для того чтобы передать как можно больше информации от учителя студенту.

Обучение происходило в штатном режиме. На рисунке 60 показана ошибка на тренировочных данных. Как видно ошибка с каждой эпохой уменьшается.

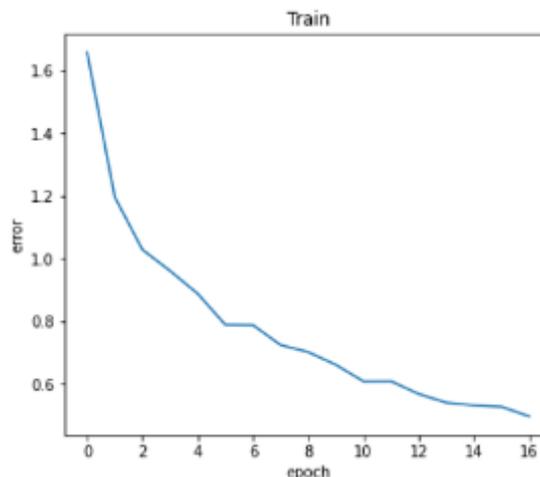


Рисунок 60 – Ошибка дистилляционной модели

На рисунке 61 показана точность на тестовых данных. Как видно точность с каждой эпохой увеличивается.

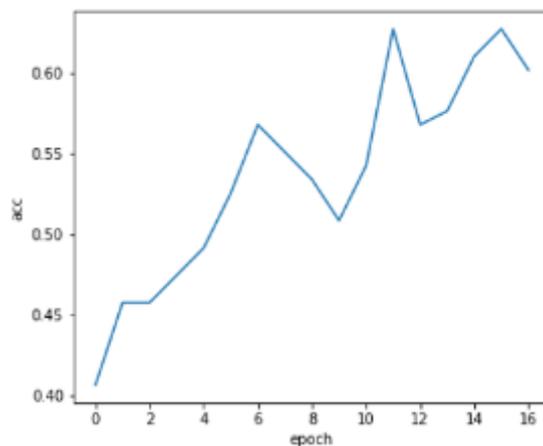


Рисунок 61 – Точность дистилляционной модели

На рисунке 62 представлены результаты дистилляционной модели.

	precision	recall	f1-score	support
0	0.91	0.62	0.74	16
1	0.74	0.88	0.80	16
2	0.91	0.62	0.74	16
3	0.33	0.38	0.35	16
4	0.50	0.94	0.65	16
5	0.93	0.81	0.87	16
6	0.00	0.00	0.00	3
7	0.82	0.56	0.67	16
8	0.75	1.00	0.86	3
accuracy			0.68	118
macro avg	0.65	0.65	0.63	118
weighted avg	0.72	0.68	0.68	118

Рисунок 62 – Результаты дистилляционной модели

К сожалению, не получилось за счет дистилляции улучшить метрики качества.

На рисунках 63 и 64, приведены результаты работы дистилляционной модели. Сверху выводиться пересказанное значение, а снизу правильное.

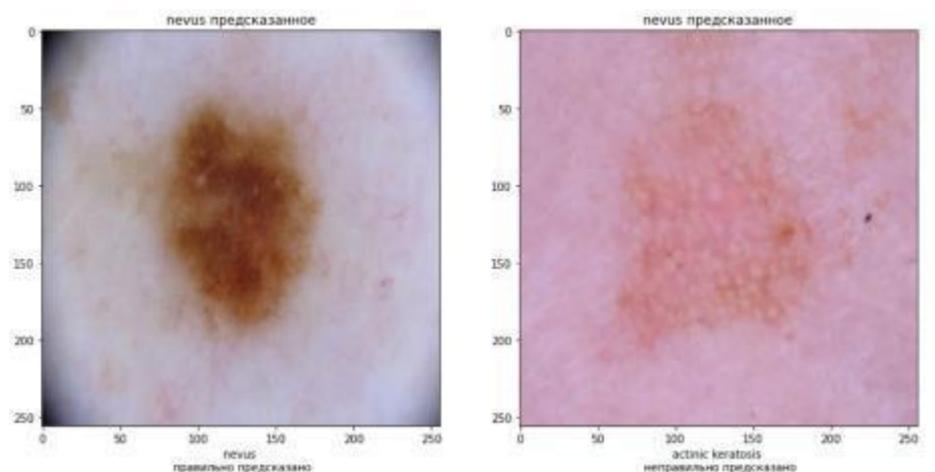


Рисунок 63 – Результаты дистилляционной модели

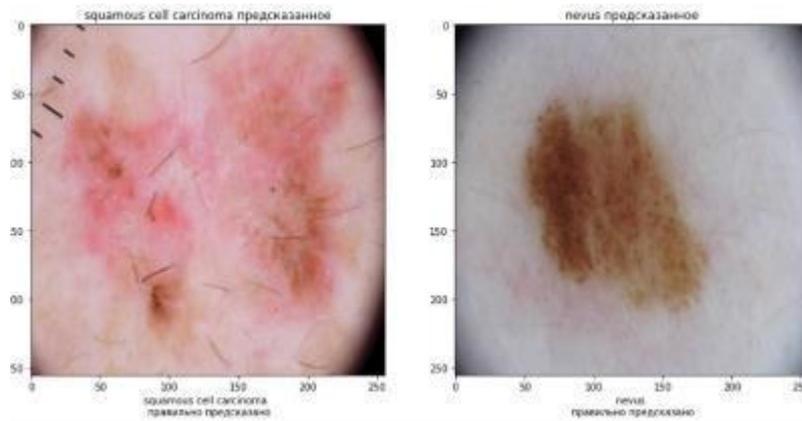


Рисунок 64 – Результаты дистилляционной модели

Вывод

Как видно из результатов самые лучшие оценки получилось у EfficientNet_V2_L accuracy = 0.68, macro avg = 0.63, weighted avg = 0.68, как видно результаты довольно неплохие. Несмотря на то что EfficientNet_V2_L имеет количество параметров во много раз меньше чем vgg16 и wide_resnet50_2, результаты vgg16 и wide_resnet50_2 хуже, могу предположить, что это из-за того, что модель просто переобучается. Также дистилляция к сожалению, не помогла.

3 СЕГМЕНТАЦИЯ

Иногда мало просто знать тип родинки, нужно еще знать ее поражение для того чтобы понять насколько она опасна.

Проблема:

Иногда родинки на изображениях полупрозрачные поэтому может быть тяжело их сегментировать, также оболочки родинок очень похожи между собой.

Анализ датасета

Для решения этой задачи был спользован ISIC Challenge Datasets 2017.

Он в себя включает изображения родинок и бинарные маски.

Train 2000 img

Val 150 img

Test 600 img

Наша задача — это сегментировать родинку на изображении, найти поражение. На рисунке 65 приведены данные, которые нам даны, то есть родинка и ее поражение.

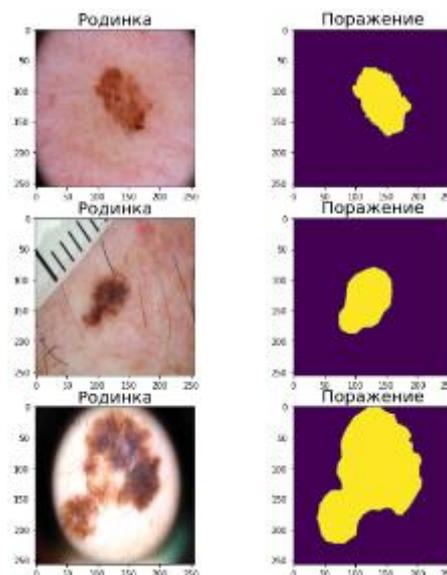


Рисунок 65 – Данные

3.1 Идея моделей сегментации

На вход подается изображение, оно уменьшается в размерности, после из сжатого состояния оно пытается предсказать бинарную маску.

На рисунке 66 представлена общая архитектура сегментационных моделей, которая представляет основную идею.

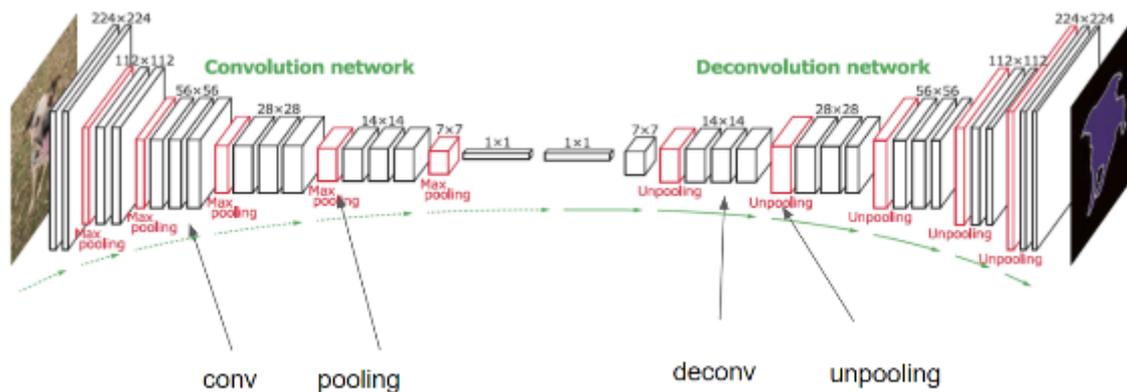


Рисунок 66 – Общая архитектура сегментационных моделей

Для оценки качества модели в данной работе используется IoU – intersection over union. На 17 формуле приведена формула IoU:

$$IoU = \frac{target \cap prediction}{target \cup prediction} \quad (17)$$

где *target* – правильная бинарная маска, *prediction* – предсказанная бинарная маска.

Интуиция - $IoU = 1$, если предсказанная маска полностью совпадает с целевой и $IoU = 0$ если пересечение равно 0.

На рисунке 67 показано, как целевая маска и предсказанное.

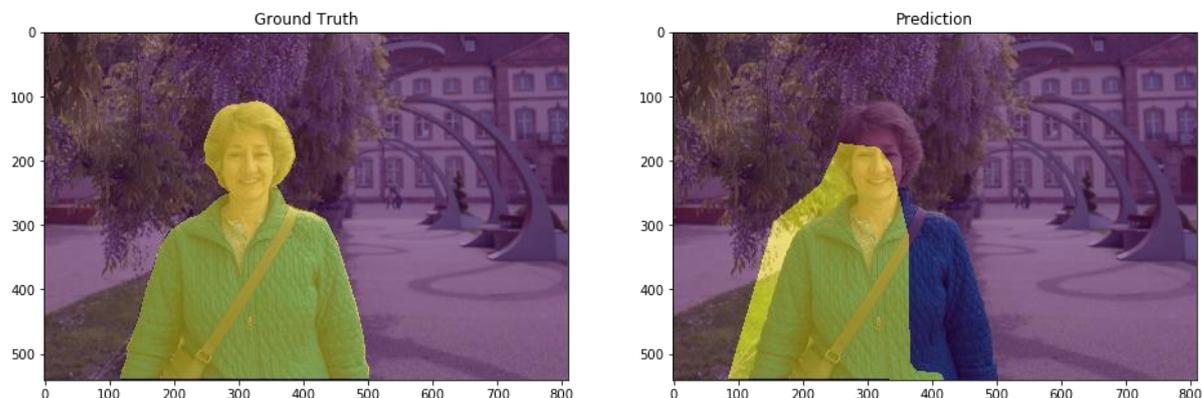


Рисунок 67 – Целевая маска и предсказанная

На рисунке 68 показаны операции над множествами.

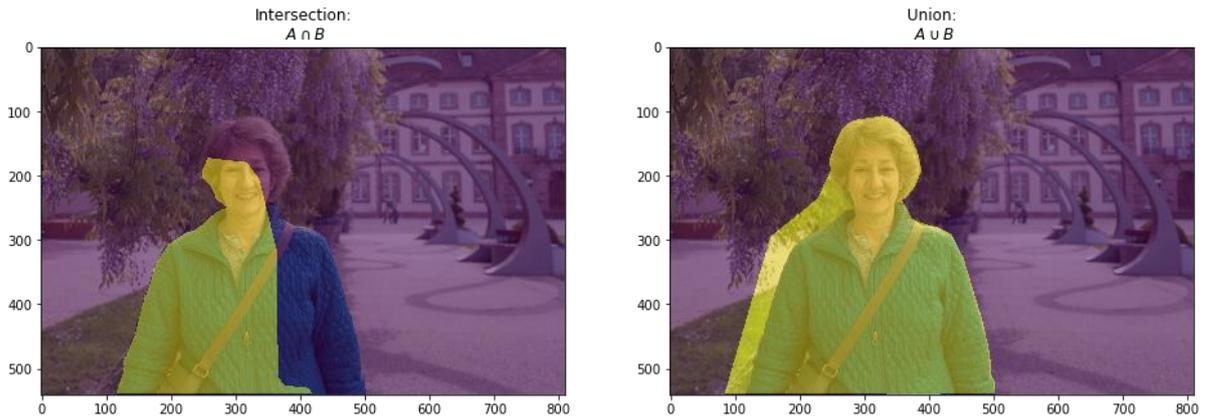


Рисунок 68 – Операции над множествами

У нас проблема бинарной классификации поэтому можно использовать формулу BCEloss в качестве функции стоимости.

На формуле 18 приведена формула BCEloss:

$$BCEloss = -\frac{1}{N} \sum_i^N y_i * \log \log (p(\hat{y}_i)) + (1 - y_i) * \log (1 - p(\hat{y}_i)), \quad (18)$$

где y_i – таргет, а \hat{y}_i – предсказанное.

Однако эта функция стоимости страдает от проблем численной нестабильности. $\lim_{x \rightarrow 0} \log \log (x) = \infty$ приводит к неустойчивости в процессе оптимизации, поэтому будет использоваться точно такая же идейно формула, но в другом представлении BCEWithLogitsLoss.

На формуле 19 приведена формула BCEWithLogitsLoss.

$$BCEWithLogitsLoss = \hat{y}_i - y_i * \hat{y}_i + \log (1 + \exp (-\hat{y}_i)), \quad (19)$$

где y_i – таргет, а \hat{y}_i – предсказанное.

3.2 UNET encoder resnet18

На рисунке 69 представлена архитектура модели UNET.

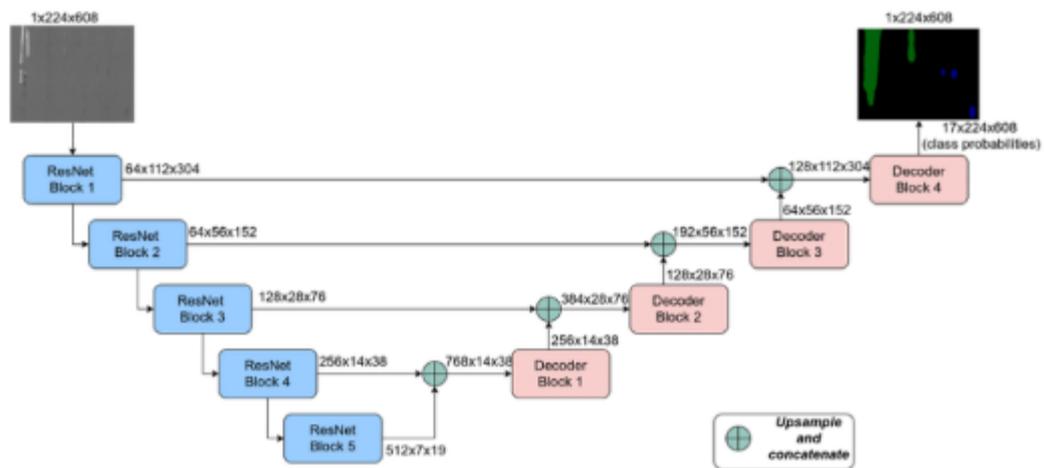


Рисунок 69 – Архитектура модели UNET

Идея:

На encoder уровнях происходит передача карт активации и concat их с картами активациями на decoder уровнях, тем самым передаем информацию на кодировании декодировщику. Также в качестве энкодеров используются resnets, для того чтобы сделать сеть больше добавляем res блоки со skip-connection.

В качестве encoder выбрана resnet18 глубиной 5 и с весами imagenet. Это маленькая модель, она имеет 11М весов.

На рисунке 70 приведены результаты работы UNET(resnet18), как видно ошибка на тренировочных данных падает, а IoU растет. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

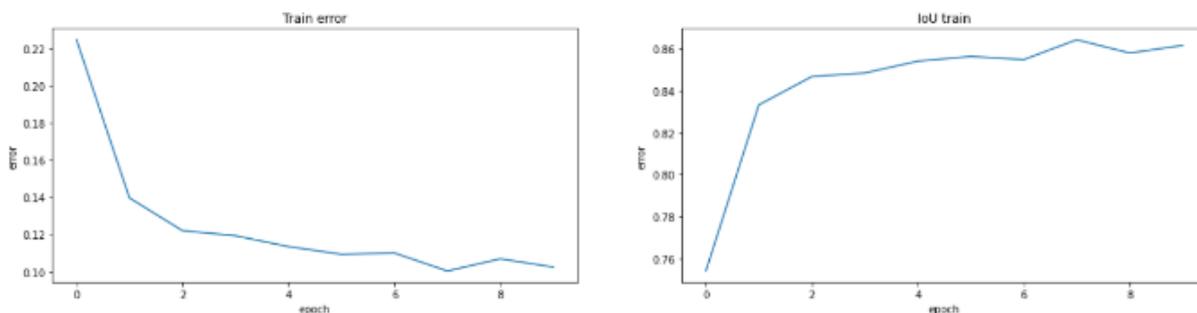


Рисунок 70 – Ошибка UNET(resnet18) на тренировочных данных и IoU

На рисунке 71 приведены результаты работы UNET(resnet18), как видно ошибка на валидационных данных падает, а IoU растет. Первая картинка — это ошибка на валидационных данных, вторая картинка это IoU.

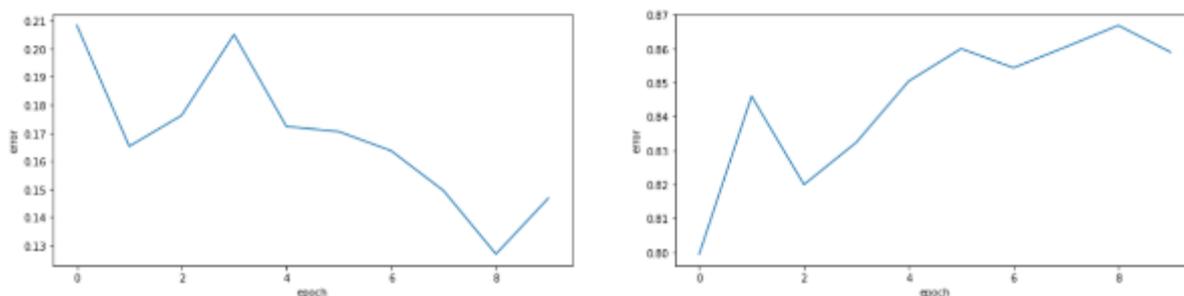


Рисунок 71 – Ошибка UNET(resnet18) на валидационных данных и IoU

Как видно из результатов, ошибка на train и val падает, IoU на train и val увеличивается. Выбор порога.

Нейростеть предсказывает не метки 0,1, а вероятности, поэтому выходное значение нужно обрезать по порогу. В зависимости от порога будет меняться IoU, поэтому нужно выбрать порог, при котором IoU максимальное.

На рисунке 72 праведен график зависимости IoU от порога.

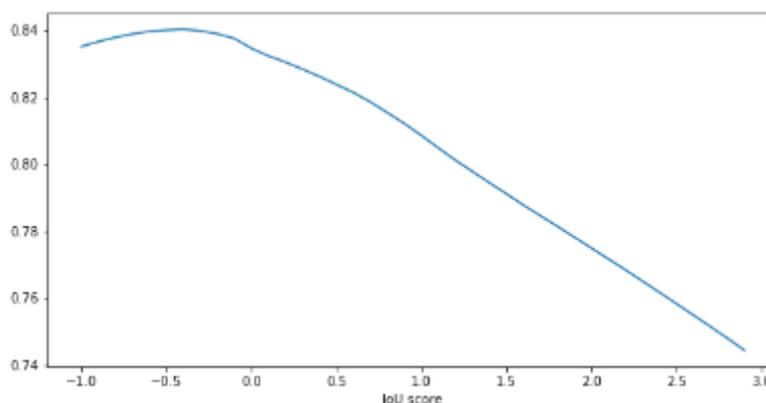


Рисунок 72 – График зависимости IoU от порога UNET(resnet18)

На рисунке 73 праведен порог, при котором достигается наивысший IoU.

```
1 round(best_threshold, 3)
-0.4
```

Рисунок 73 – Наилучший порог для UNET(resnet18)

Лучшая IoU достигается при пороге -0.4 и IoU = 0.84

На рисунке 74 приведены изображение родинки, правильная маска и предсказанная маска. Под предсказанной маской выведена IoU правильной маской и предсказанной.

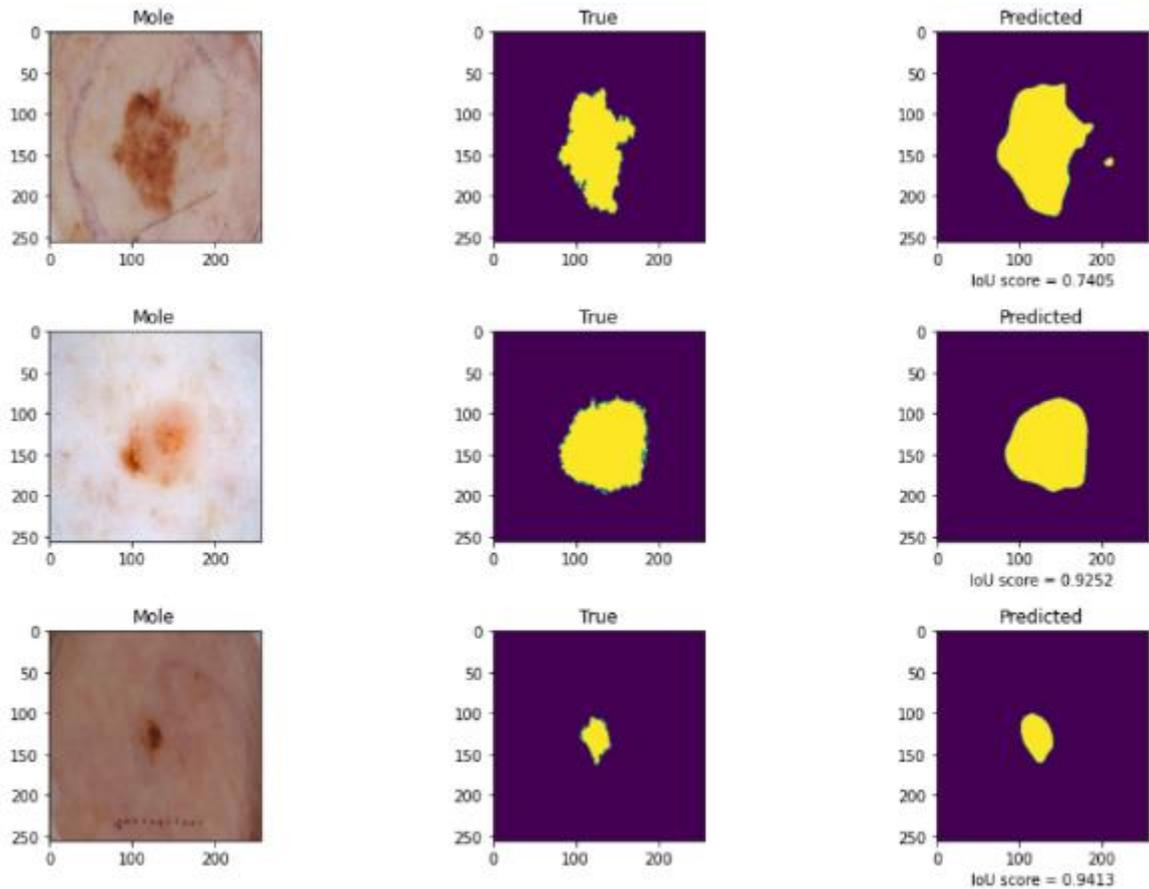


Рисунок 74 – Результаты UNET(resnet18)

3.3 UNET encoder resnet101

Увеличение размера энкодера и декодера может помочь при сегментации, поэтому в качестве encoder выбрана resnet101 глубиной 5 и с весами imagenet. Это достаточно большая модель, она имеет 42M весов в будущем она будет использоваться, как модель учитель при дистилляции.

На рисунке 75 приведены результаты работы UNET(resnet101), как видно ошибка на тренировочных данных падает, а IoU растёт. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

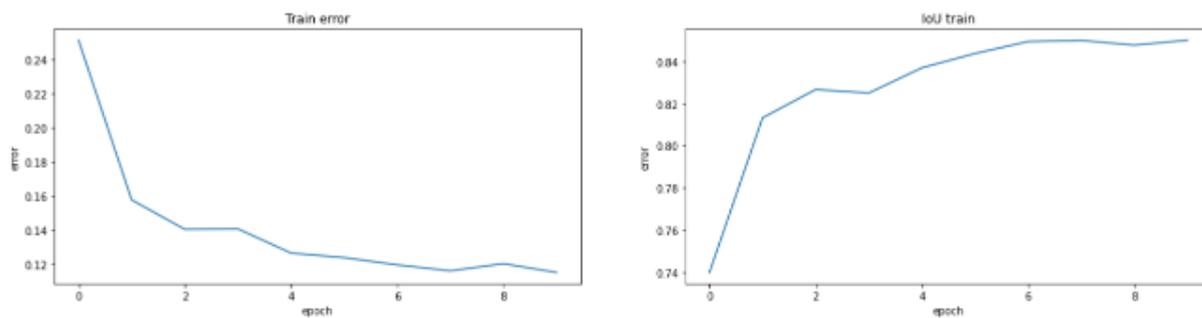


Рисунок 75 – Результаты UNET(resnet101)

На рисунке 76 приведены результаты работы UNET(resnet101), как видно ошибка на валидационных данных падает, а IoU растёт. Первая картинка — это ошибка на валидационных данных, вторая картинка это IoU.

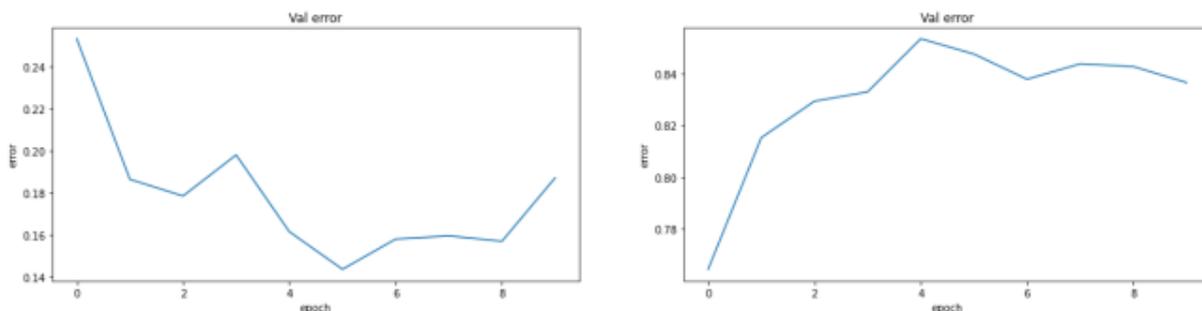


Рисунок 76 – Результаты UNET(resnet101)

Выбор порога

Как видно из результатов, ошибка на train и val падает, IoU на train и val увеличивается. На рисунке 77 праведен график зависимости IoU от порога.

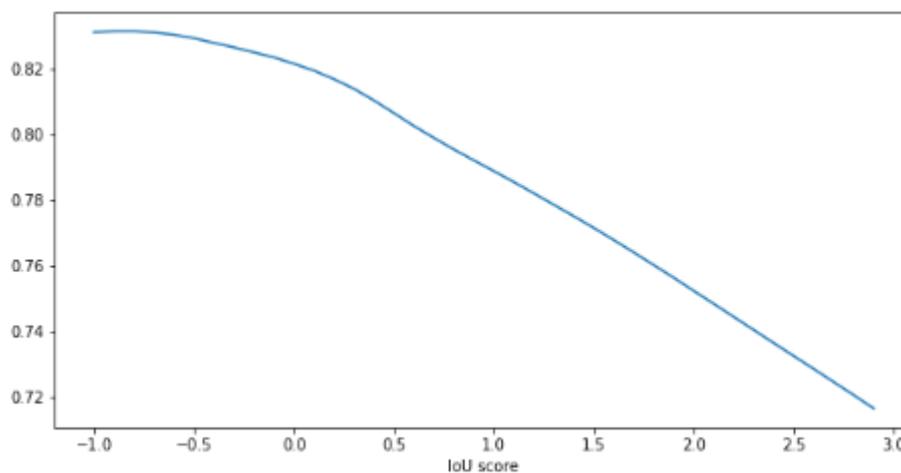


Рисунок 77 – График зависимости IoU от порога UNET(resnet101)

На рисунке 78 праведен порог, при котором достигается наивысший IoU.

```
1 round(best_threshold,3)
-0.8
```

Рисунок 78 – Наилучший порог для UNET(resnet101)

Лучшая IoU при пороге -0.8 IoU = 0.835.

На рисунке 79 приведены изображение родинки, правильная маска и предсказанная маска. Под предсказанной маской выведена IoU правильной маской и предсказанной.

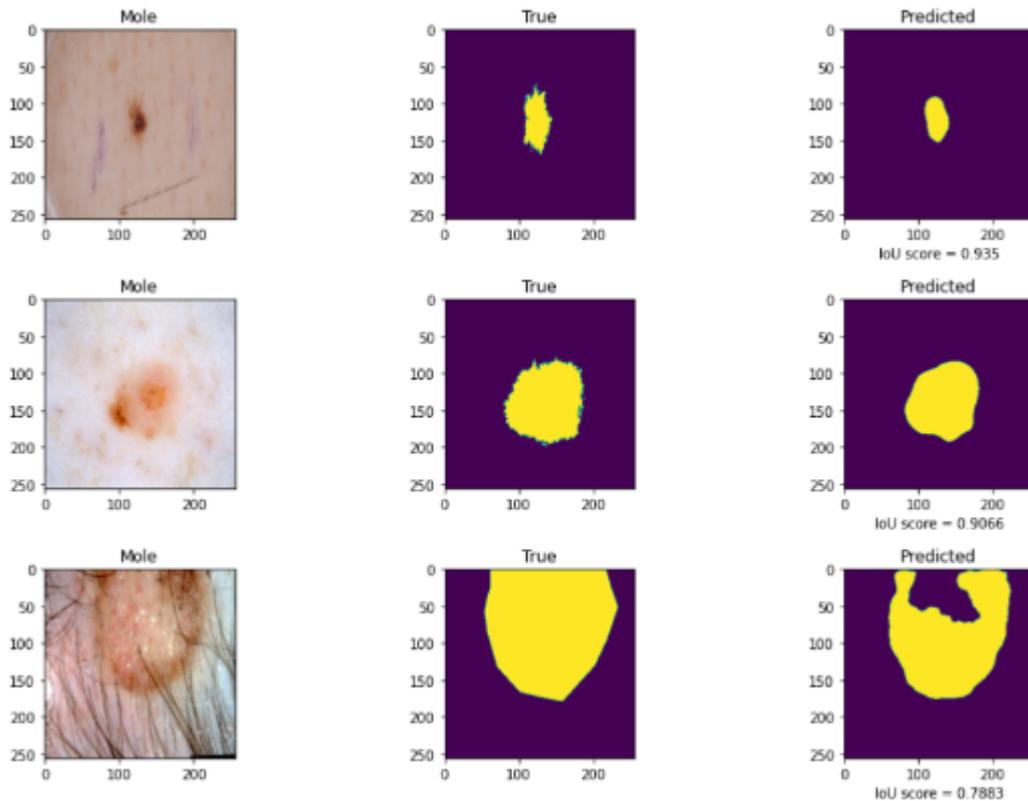


Рисунок 79 – Результаты UNET(resnet101)

Можно увидеть из результатов, что сегментация хорошая.

Показатель IoU меньше чем при использовании resnet18. Увеличение кодировщика и декодировщика не помогло.

3.4 LinkNet

На рисунке 80 приведена архитектура LinkNet.

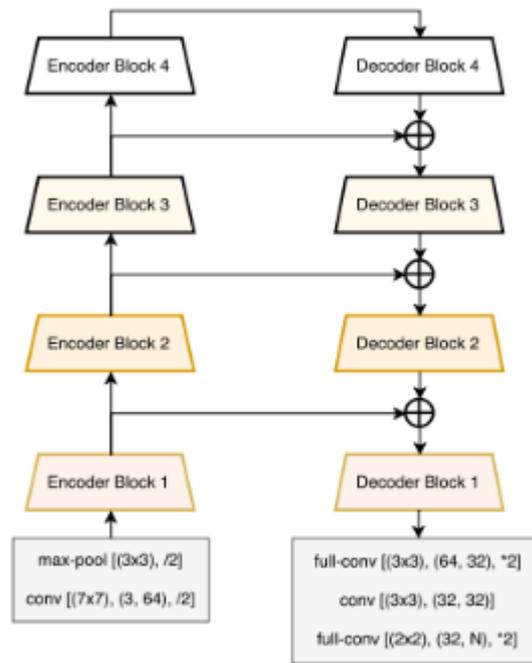


Fig. 1: LinkNet Architecture

Рисунок 80 – Архитектура LinkNet

На рисунке 81 приведена архитектура encoder блока.

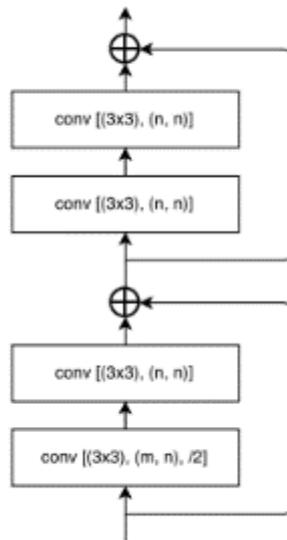


Рисунок 81 – Архитектура encoder блока

На рисунке 82 приведена архитектура decoder блока.

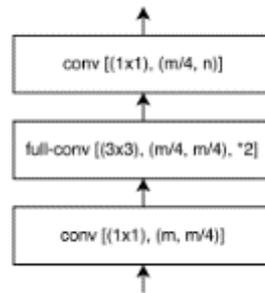


Рисунок 82 – Архитектура decoder блока

Как написано в статье, эта модель маленькая, имеет 11М параметров. Идея была такая, создать маленькую модель, которая по точности не уступала бы моделям с огромным числом параметров, и в будущем для использования ее в продакшене.

На рисунке 83 приведены результаты работы LinkNet, как видно ошибка на тренировочных данных падает, а IoU растет. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

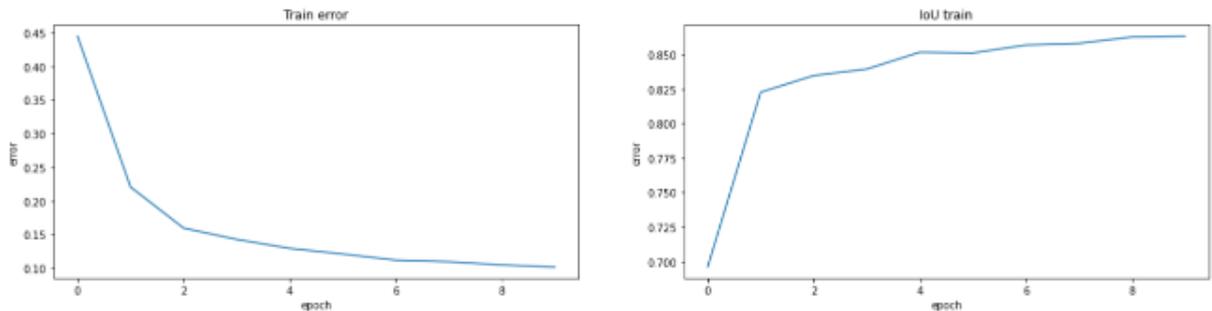


Рисунок 83 – Ошибка LinkNet на тренировочных данных и IoU

На рисунке 84 приведены результаты работы LinkNet, как видно ошибка на валидационных данных падает, а IoU растет. Первая картинка — это ошибка на валидационных данных, вторая картинка это IoU.

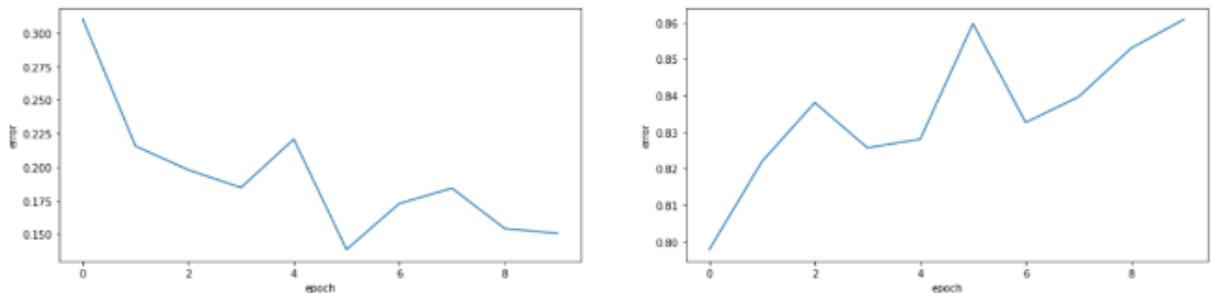


Рисунок 84 – Ошибка LinkNet на валидационных данных и IoU

Как видно из результатов, ошибка на train и val падает, IoU на train и val увеличивается. На рисунке 85 праведен график зависимости IoU от порога.

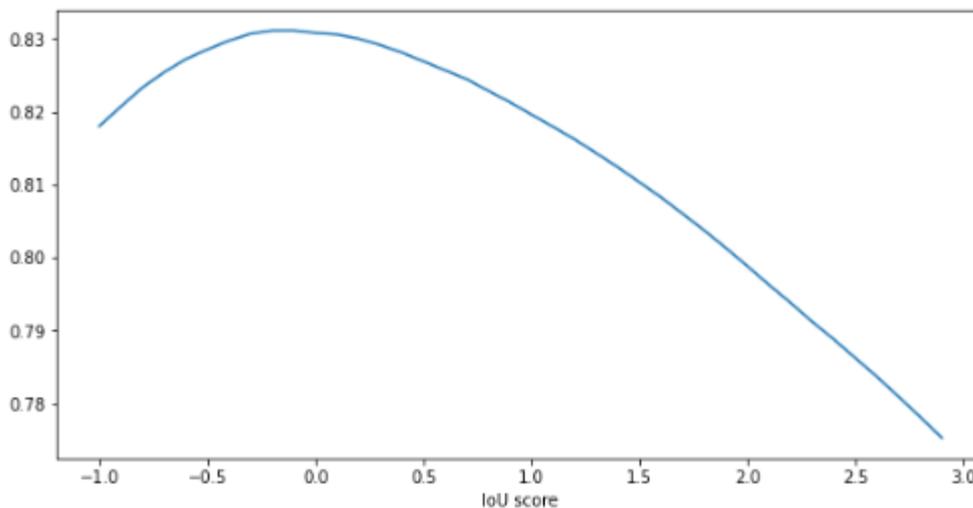


Рисунок 85 – График зависимости IoU от порога LinkNet

На рисунке 86 праведен порог, при котором достигается наивысший IoU.

```
1 round(best_threshold, 3)
-0.1
```

Рисунок 86 – Наилучший порог для LinkNet

Лучший IoU при пороге -0.1, IoU = 0.83.

На рисунке 87 приведены изображение родинки, правильная маска и предсказанная маска. Под предсказанной маской выведена IoU правильной маской и предсказанной.

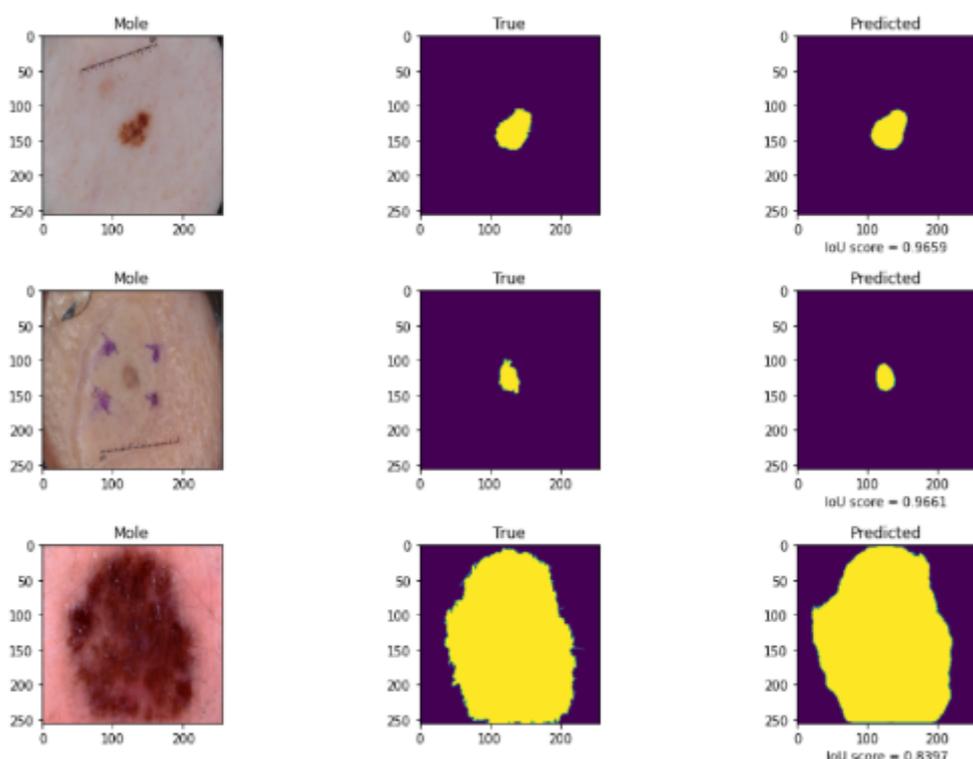


Рисунок 87 – Результаты LinkNet

Можно увидеть из результатов, что сегментация хорошая.

Показатель IoU меньше чем при использовании resnet18 и resnet101. Использование LinkNet не улучшило результаты.

3.5 PSPNet

На рисунке 88 приведена архитектура PSPNet.

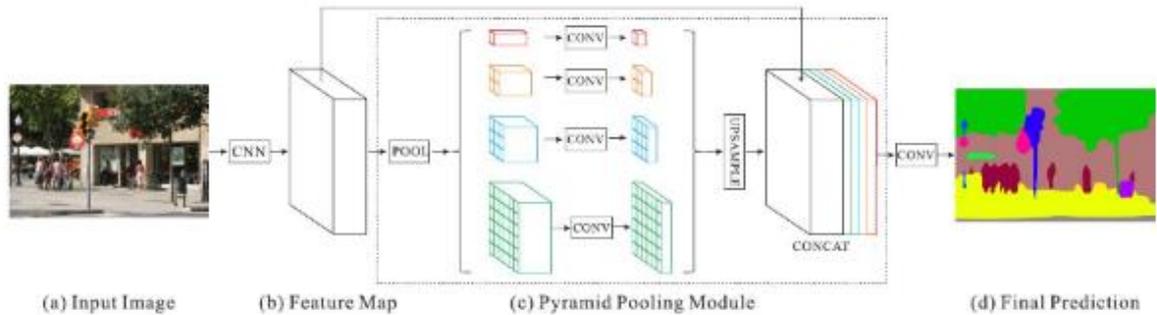


Рисунок 88 – Архитектура PSPNet

На рисунке 89 приведены результаты работы PSPNet, как видно ошибка на тренировочных данных падает, а IoU растет. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

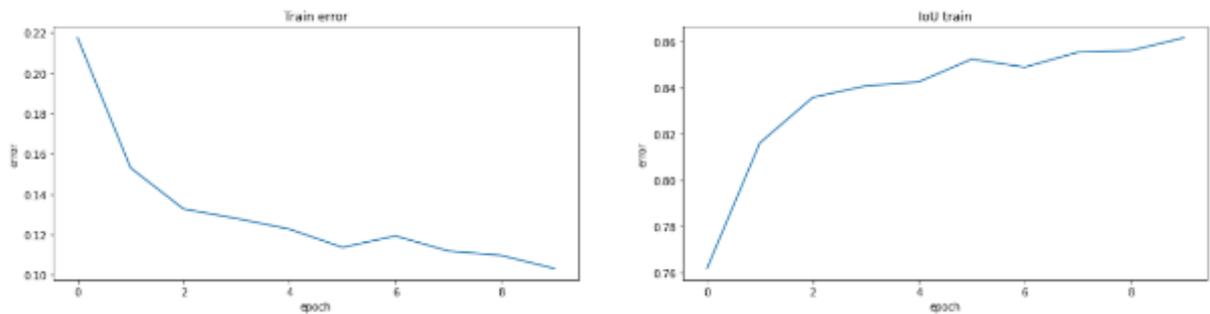


Рисунок 89 – Ошибка PSPNet на тренировочных данных и IoU

На рисунке 90 приведены результаты работы PSPNet, как видно ошибка на валидационных данных падает, а IoU растет. Первая картинка — это ошибка на валидационных данных, вторая картинка это IoU.

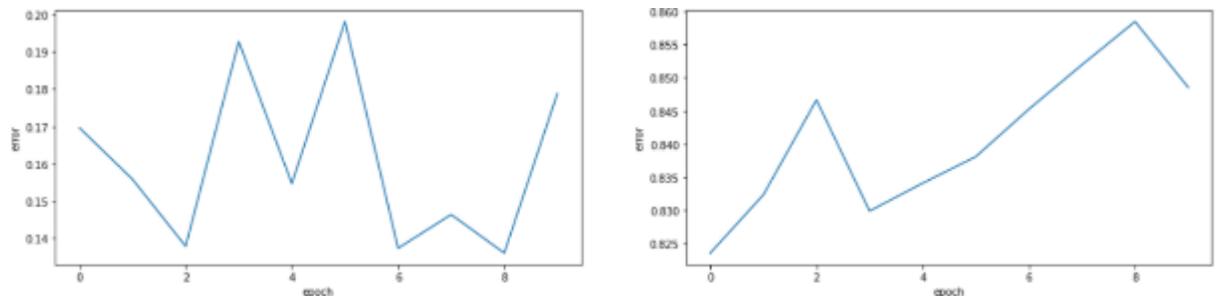


Рисунок 90 – Ошибка PSPNet на валидационных данных и IoU

Как видно из результатов, ошибка на train и val падает, IoU на train и val увеличивается. На рисунке 91 приведен график зависимости IoU от порога.

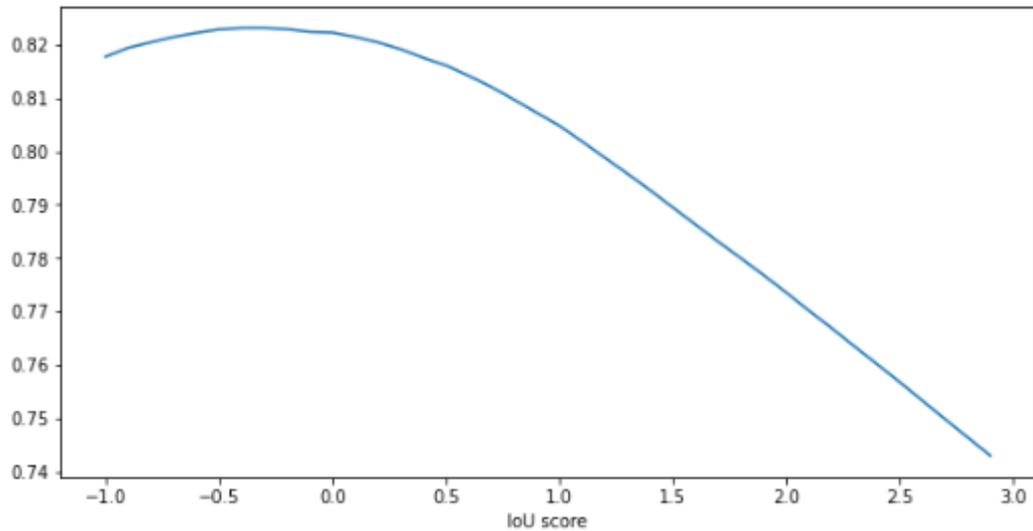


Рисунок 91 – График зависимости IoU от порога PSPNet

На рисунке 92 приведен порог, при котором достигается наивысший IoU.

```
1 round(best_threshold,3)
-0.3
```

Рисунок 92 – Наилучший порог для PSPNet

Лучший IoU при пороге -0.3, IoU равен 0.82.

На рисунке 93 приведены изображение родинки, правильная маска и предсказанная маска. Под предсказанной маской выведена IoU правильной маской и предсказанной.

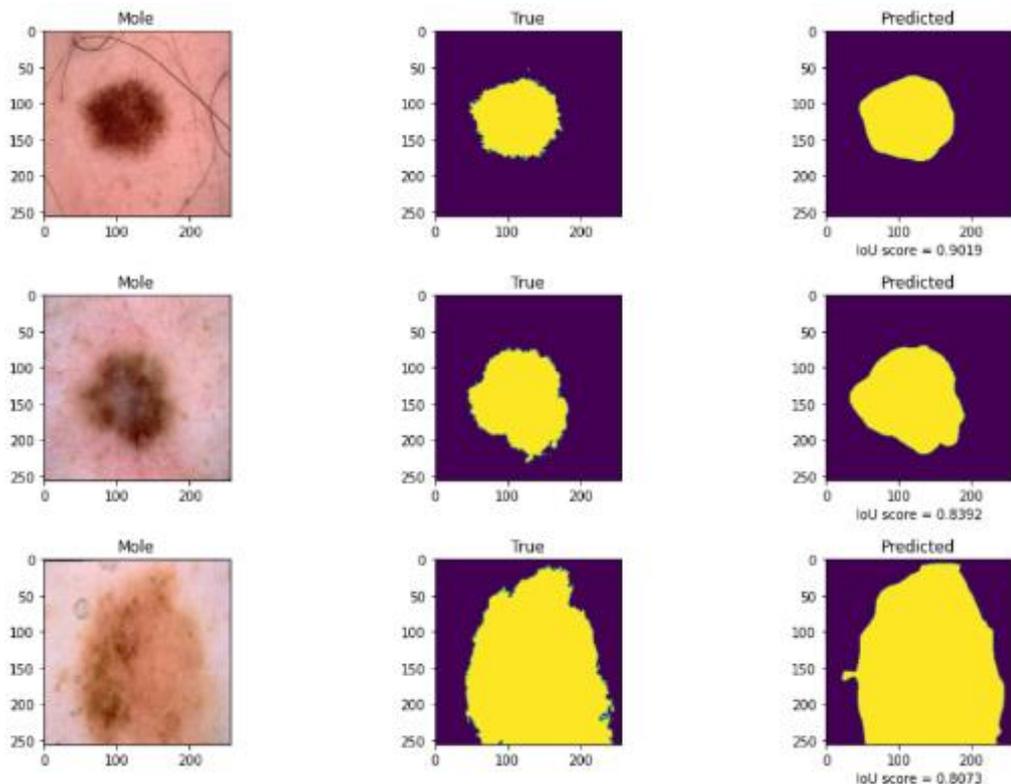


Рисунок 93 – Результаты PSPNet

Можно увидеть из результатов, что сегментация хорошая.
Показатель IoU меньше чем при использовании resnet18 и resnet101. Использование PSPNet не улучшило результаты.

3.6 PAN

На рисунке 94 приведена архитектура PAN.

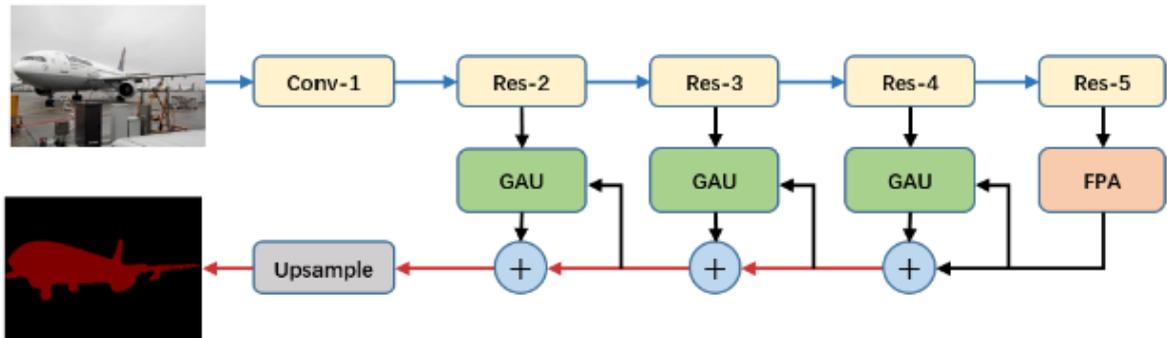


Рисунок 94 – Архитектура PAN

На рисунке 95 приведена архитектура Feature Pyramid Attention (FPA) Module.

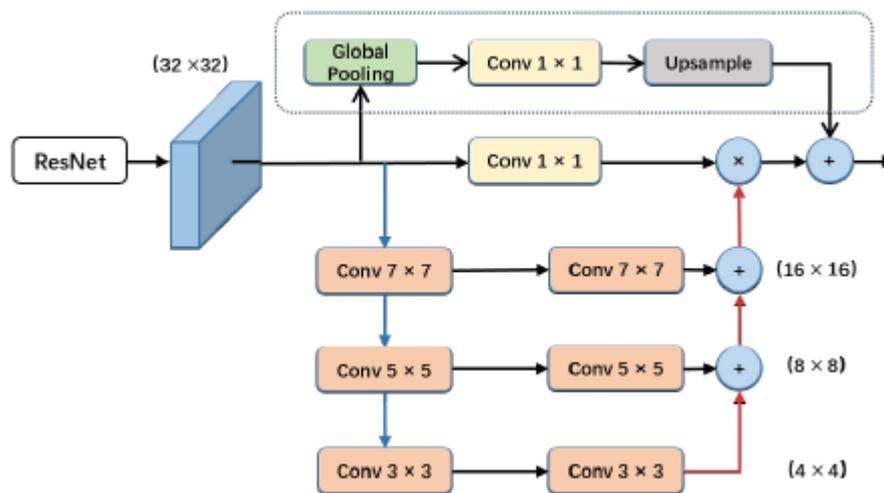


Рисунок 95 – Архитектура FPA

Feature Pyramid Attention (FPA) представлен для создания пространственной пирамидальной структуры внимания на высокоуровневых выходных данных и объединения global pooling для изучения лучшего представления функций [4].

На рисунке 96 приведена архитектура Global Attention Upsample (GAU)

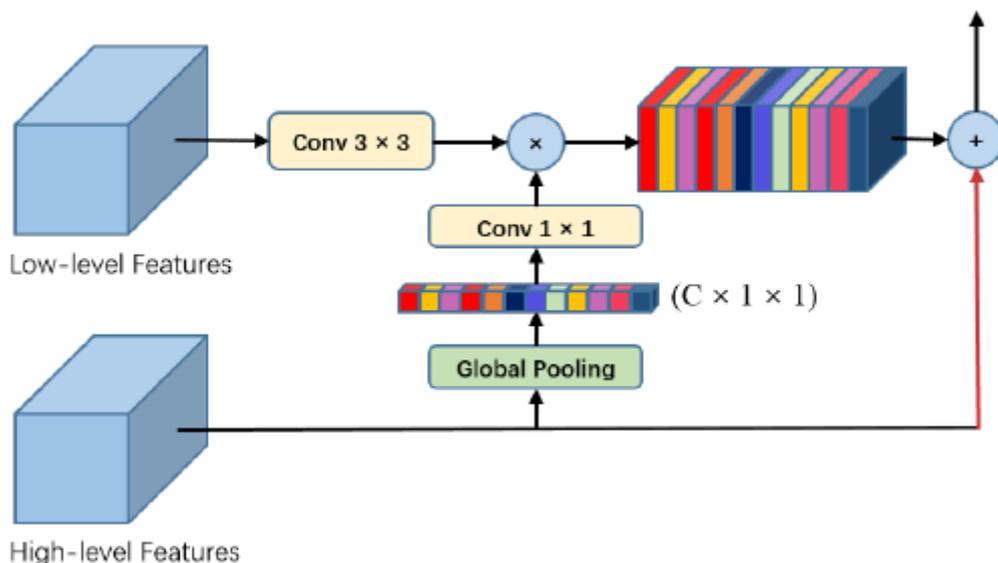


Рисунок 96 – Архитектура GAU

Модуль Global Attention Upsample (GAU) представлен на каждом уровне декодера, чтобы предоставить глобальный контекст в качестве руководства по низкоуровневым функциям для выбора деталей локализации категории [4].

На рисунке 97 приведены результаты работы PAN, как видно ошибка на тренировочных данных падает, а IoU растет. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

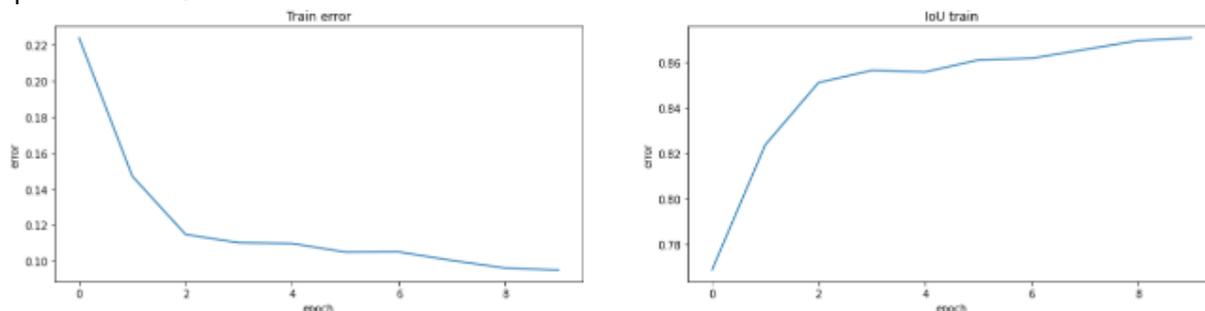


Рисунок 97 – Ошибка PAN на тренировочных данных и IoU

На рисунке 98 приведены результаты работы PAN, как видно ошибка на валидационных данных падает, а IoU растет. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

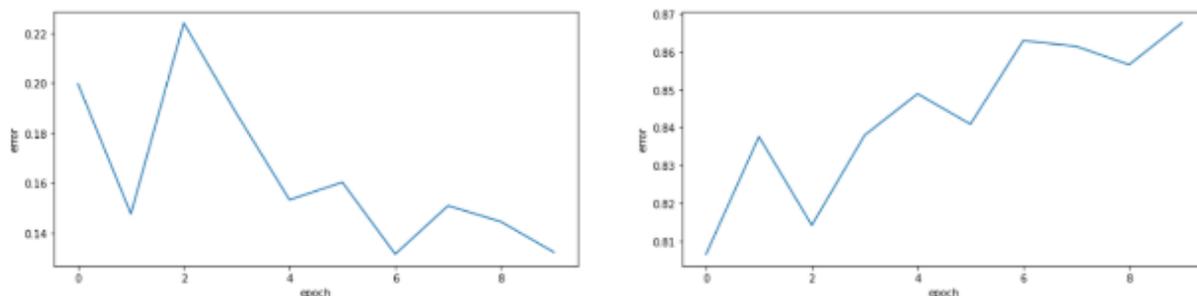


Рисунок 98 – Ошибка PAN на валидационных данных и IoU

Как видно из результатов, ошибка на train и val падает, IoU на train и val увеличивается. На рисунке 99 приведен график зависимости IoU от порога.

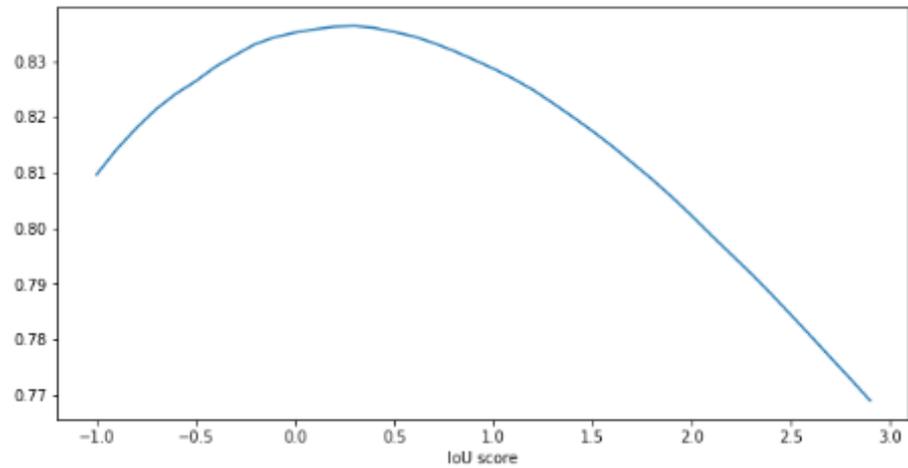


Рисунок 99 – График зависимости IoU от порога PAN

На рисунке 100 праведен порог, при котором достигается наивысший IoU.

```
1 round(best_threshold, 3)
0.3
```

Рисунок 100 – Наилучший порог для PAN

Лучший IoU при пороге 0.3, IoU = 0.84.

На рисунке 101 приведены изображение родинки, правильная маска и предсказанная маска. Под предсказанной маской выведена IoU правильной маской и предсказанной.

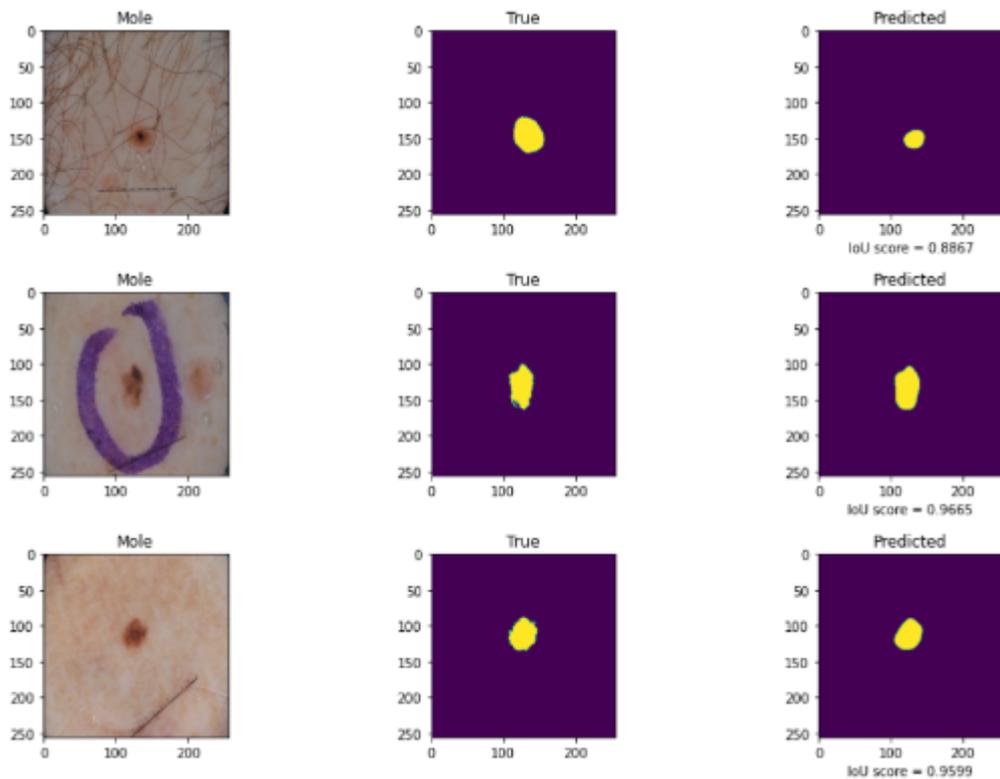


Рисунок 101 – Результаты PAN

Можно увидеть из результатов, что сегментация хорошая
Показатель IoU при использовании PAN лучше, чем при использовании resnet101, PSPNet, LinkNet.

3.7 Дистилляция

В реальности очень большие модели, например, такие как Unet resnet101 (в реальности намного больше) очень тяжело интегрировать в продакшен, одна из причин — это огромный вес модели, число ее параметров. Поэтому придумали подход, благодаря которому маленькая модель может получить похожие результаты, как и большая. В начале мы обучаем большую модель на основе наших данных, потом тренируем маленькую модель на основе тех же данных в связке с этой большой сетью и интегрируем в продакшен маленькую модель. В качестве учителя будет выступать Unet resnet101, а в качестве ученика LinkNet resnet34, потому что она показала самые плохие результаты.

На рисунке 102 приведены результаты работы дистиллированной модели, как видно ошибка на тренировочных данных падает, а IoU растет. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

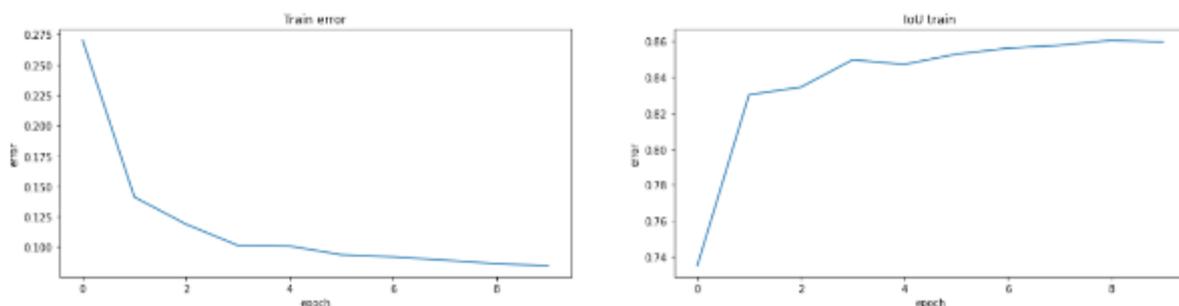


Рисунок 102 – Ошибка дистиллированной модели на тренировочных данных и IoU

На рисунке 103 приведены результаты работы дистиллированной модели, как видно ошибка на валидационных данных падает, а IoU растет. Первая картинка — это ошибка на тренировочных данных, вторая картинка это IoU.

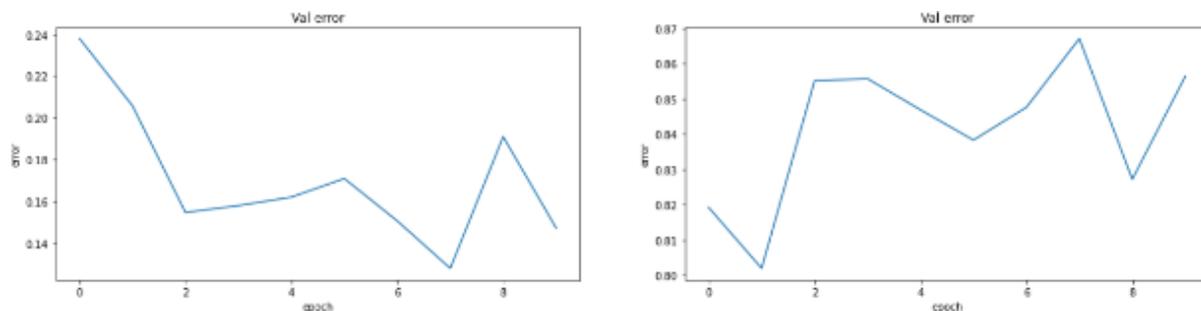


Рисунок 103 – Ошибка дистиллированной модели на валидационных данных и IoU

Как видно из результатов, ошибка на train и val падает, IoU на train и val увеличивается. На рисунке 104 приведен график зависимости IoU от порога.

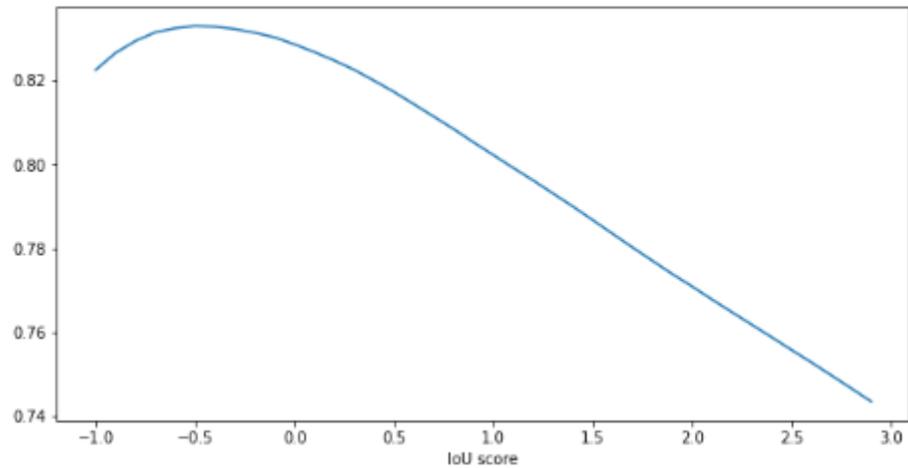


Рисунок 104 – График зависимости IoU от порога дистиллированной модели

На рисунке 105 праведен порог, при котором достигается наивысший IoU.

```
1 round(best_threshold, 3)
-0.5
```

Рисунок 105 – Наилучший порог для дистиллированной модели

Лучшая IoU при пороге -0.5, IoU = 0.84.

После дистилляции качество модели LeakyNet увеличилось на 1%.

На рисунке 106 приведены изображение родинки, правильная маска и предсказанная маска. Под предсказанной маской выведена IoU правильной маской и предсказанной.

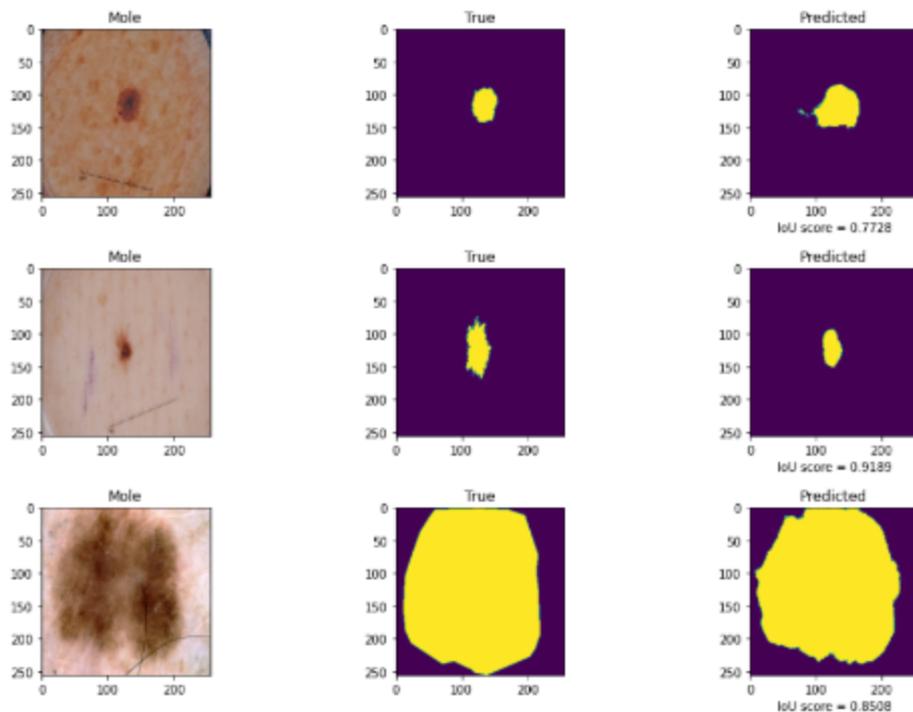


Рисунок 106 – Результаты дистиллированной модели

Можно увидеть из результатов, что сегментация хорошая.

Показатель IoU при использовании дистиллированной модели лучше, чем при использовании resnet101, PSPNet, LinkNet.

3.8 Вывод

Как видно из результатов все модели справились с задачей сегментации хорошо, IoU равна 0.82 - 0.84. Самые лучшие модели это Unet и PAN. После дистилляции качество модели LeakyNet увеличилось на 1%.

4 ДЕТЕКЦИЯ И КЛАССИФИКАЦИЯ ОЖГОВ

После получения ожога очень важно быстро принять меры. В зависимости от степени ожога, его размера и расположения будет зависеть ваши дальнейшие действия.

Проблемы:

Мало данных, ожоги имеют сложную структуру, поэтому классифицировать и детектировать их тяжело.

Гипотеза:

Определить существует модель, которая решает поставленную задачу с должным уровнем точности.

В работе данная задача будет пытаться решаться с помощью библиотеки ultralytics/yolov5, будут исследованы модели yolov5, которые лучше всего решают поставленную задачу.

Анализ данных Skin Burn Dataset

Этот набор данных состоит примерно из 1300 изображений из Интернета с разной степенью ожога кожи.

Этикетки представлены в формате YOLO с классами 0,1,2, обозначающими ожог первой степени, ожог второй степени и ожог третьей степени соответственно.

На рисунке 107 представлены данные с которыми мы работаем.

0 0.7366771159874608 0.6360759493670886 0.1755485893416928 0.18354430379746836

Рисунок 107 – Входные данные

Первое число — это метка, второе и третья это центр бокса, четвертое и пятое – это ширина и высота бокса.

На рисунке 108 представлены данные которые мы получаем, это координаты бокса и метка класса.



Рисунок 108 – Входные данные

4.1 Идея YOLOV5

Для решения задачи использовалась YOLOV5. Саму модель и все алгоритмы я взял с репозитория ultralytics/yolov5 [5].

На рисунке 109 представлена архитектура YOLOV5.

Overview of YOLOv5

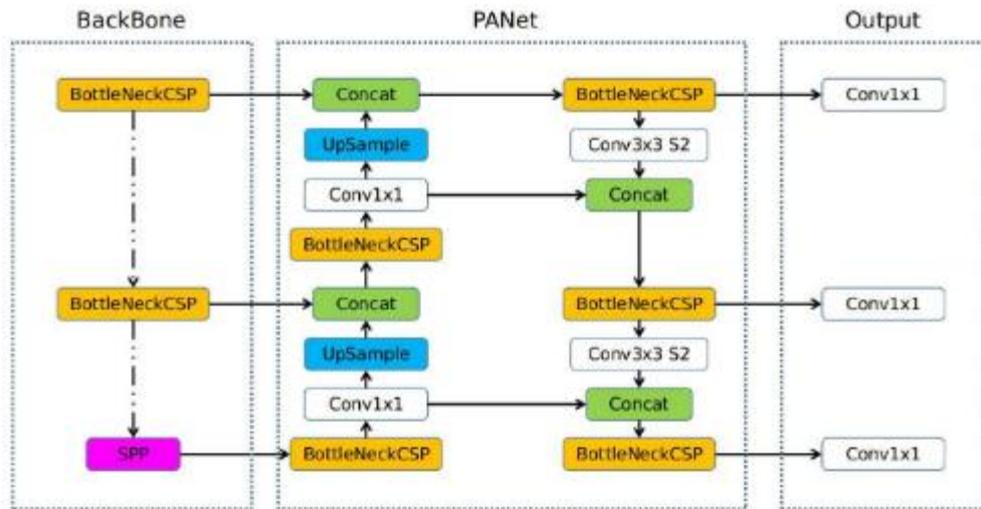


Рисунок 109 – Архитектура YOLOv5

YOLO v5 требует, чтобы набор данных был в формате *даркнет*. Вот схема того, как это выглядит:

- 1 Один текстовый файл с метками на изображение.
 - 2 Одна строка на объект.
 - 3 Каждая строка содержит: class_index bbox_x_center bbox_y_center bbox_width bbox_height.
 - 4 Координаты бокса должны быть нормализованы между 0 и 1.
- На рисунке 110 формат в котором должны быть расположены данные.

```

...
burns
├── images
│   ├── train
│   └── val
├── labels
│   ├── train
│   └── val
...
    
```

Рисунок 110 – Расположение данных

На рисунке 111 приведены модели, доступные в ultralytics/yolov5. В своей работе я буду использовать вариации моделей YOLOv5s, YOLOv5m, YOLOv5x.

Nano	Small	Medium	Large	XLarge
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
4 MB _{FP16}	14 MB _{FP16}	41 MB _{FP16}	89 MB _{FP16}	166 MB _{FP16}
6.3 ms _{V100}	6.4 ms _{V100}	8.2 ms _{V100}	10.1 ms _{V100}	12.1 ms _{V100}
28.4 mAP _{COCO}	37.2 mAP _{COCO}	45.2 mAP _{COCO}	48.8 mAP _{COCO}	50.7 mAP _{COCO}

Рисунок 111 – Доступные модели

На формуле 20 приведена YOLOLoss:

$$YOLOLoss = lcalization_loss + confidence_loss + classification_loss, \quad (20)$$

где $lcalization_loss$ – штрафует за неправильные координаты бокса, $confidence_loss$ – штрафует за неправильную уверенность в том, что в боксе находится объект, $classification_loss$ - штрафует за неправильную классификацию объекта в боксе.

На формуле 21 приведена $lcalization_loss$:

$$lcalization_loss = a_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \left[(x_{i,j} - \hat{x}_{i,j})^2 + (y_{i,j} - \hat{y}_{i,j})^2 \right] + a_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \left[(\sqrt{w_{i,j}} - \sqrt{\hat{w}_{i,j}})^2 + (\sqrt{h_{i,j}} - \sqrt{\hat{h}_{i,j}})^2 \right], \quad (21)$$

где $x_{i,j}, y_{i,j}$ – координаты центра цели, $w_{i,j}$ – ширина цели, $h_{i,j}$ – высота цели, $\hat{x}_{i,j}, \hat{y}_{i,j}$ – координаты центра предсказанное, $\hat{w}_{i,j}$ – ширина цели, $\hat{h}_{i,j}$ – высота цели.

На формуле 22 приведена $confidence_loss$:

$$confidence_loss = \sum_{i=0}^{s^2} \sum_{j=0}^B \left[(c_{i,j} - \hat{c}_{i,j})^2 \right] + a_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B \left[(c_{i,j} - \hat{c}_{i,j})^2 \right] \quad (22)$$

где $c_{i,j}$ – целевая вероятность того что в боксе находится объект, $\hat{c}_{i,j}$ – тестовая вероятность того что в боксе находится объект.

На формуле 23 приведена $classification_loss$:

$$classification_loss = \sum_{i=0}^{s^2} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (23)$$

где $p_i(c)$ – целевая вероятность того что объект принадлежит классу c , $\hat{p}_i(c)$ – целевая вероятность того что объект принадлежит классу c .

Метрики качества:

1 box_loss — потеря регрессии ограничивающего прямоугольника (среднеквадратичная ошибка).

2 obj_loss — уверенность в наличии объекта есть потеря объектности (Binary Cross Entropy).

3 cls_loss — потеря классификации (Cross Entropy).

Точность измеряет, насколько верны предсказания bbox. (True positives / (True positives + False positives)), и Recall измеряет, какая часть истинных bbox была правильно предсказана (True positives / (True positives + False negatives)). 'mAP_0.5' средняя средняя точность (mAP) at IoU (Intersection over Union) порог 0.5. 'mAP_0.5:0.95' средний mAP по разным IoU порогам, начиная от 0.5 до 0.95.

Анализ входных данных

На рисунке 112 приведена диаграмма распределения классов.

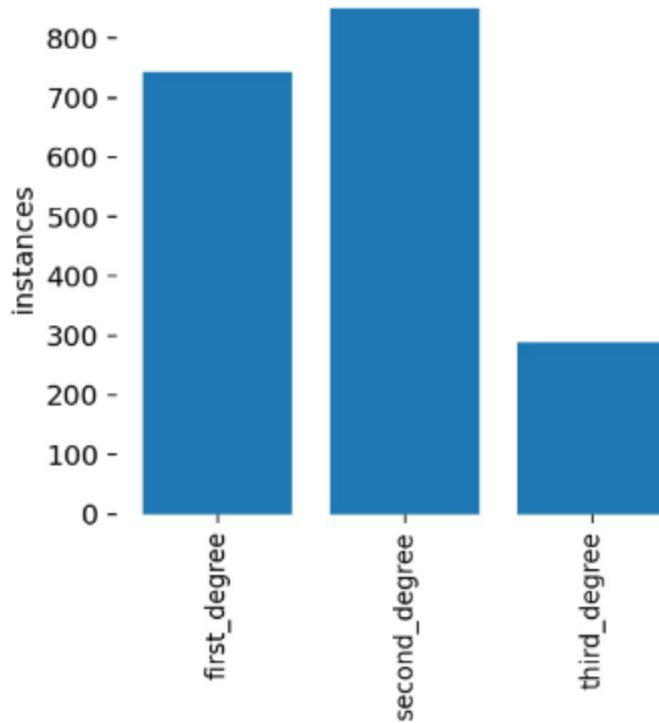


Рисунок 112 – Распределение данных

Как можно заметить данных ожогов 3-ей степени меньше всего, больше всего 2-ой степени. Данных мало особенно для данного алгоритма. В документации написано, что на каждый класс должно быть 1500 изображений.

4.2 Yolov5s(640x640)

На рисунке 113 приведены гиперпараметры Yolov5x(256x256) модели: размер изображения к которому мы приводим все наши изображения, количество батчей, количество эпох.

```
python train.py --img 640 --batch 4 --epochs 30 \
  --data ./data/burns.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt \
  --name yolov5s_burns --cache
```

Рисунок 113 – Задание модели Yolov5x(256x256)

На рисунке 114 приведены ошибки и точности на каждой эпохе на тренировочных и валидационных данных модели Yolov5s(640x640). Что означает каждая метрика описано выше.

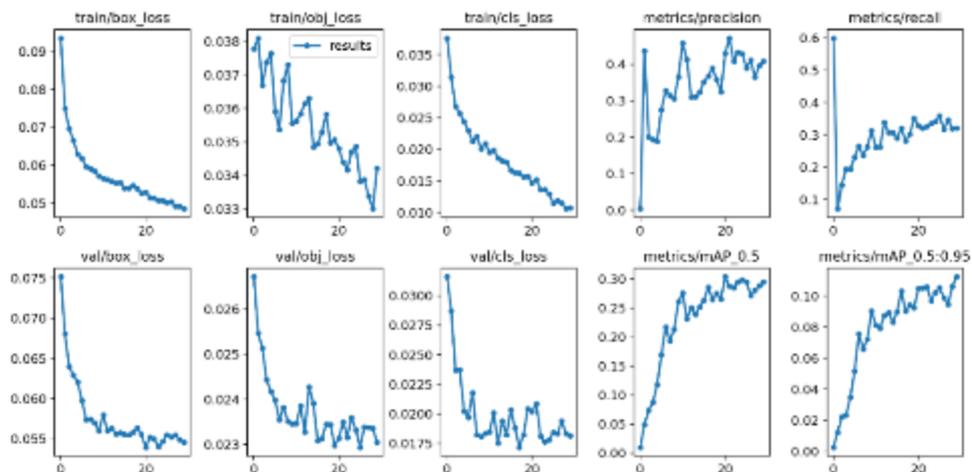


Рисунок 114 – Ошибки и точности модели YOLOv5s(640x640)

Как видно из графиков box_loss, obj_loss, cls_loss на train и val падают, а precision, recall, mAP растут.

Несмотря на хороший тренд, результаты получились плохими, низкая точность и полнота, возможно это происходит из-за очень маленького размера выборки.

На рисунке 115 представлена зависимость F1 score от уверенности модели YOLOv5s(640x640)

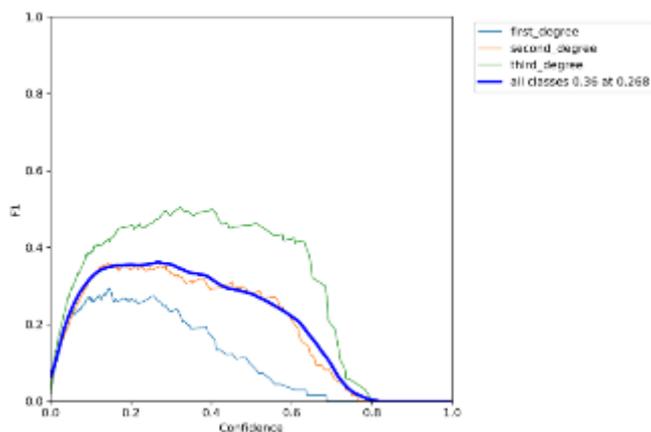


Рисунок 115 – Зависимость F1 score от уверенности модели YOLOv5s(640x640)

В среднем лучший F1 score достигается при 0.19 confidence.

На рисунке 116 представлена зависимость Precision от Recall модели YOLOv5s(640x640).

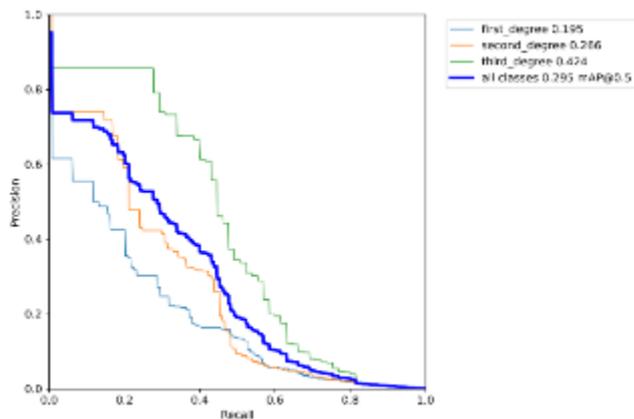


Рисунок 116 – Зависимость Precision от Recall модели YOLOv5s(640x640)

MAP низкая, но лучшая среди всех.

На рисунках рисунок 117 и 118 представлены результаты работы нейросети. Левая картинка реальная метка и бокс, правая – предсказанное. (При обучении первой нейросети были использованы метки 0,1,2, а реальные метки идут в формате 1,2,3, поэтому для получения правильного класса предсказания нужно прибавить 1 к классу, который предсказал алгоритм. Для все остальных нейросетей это исправлено).

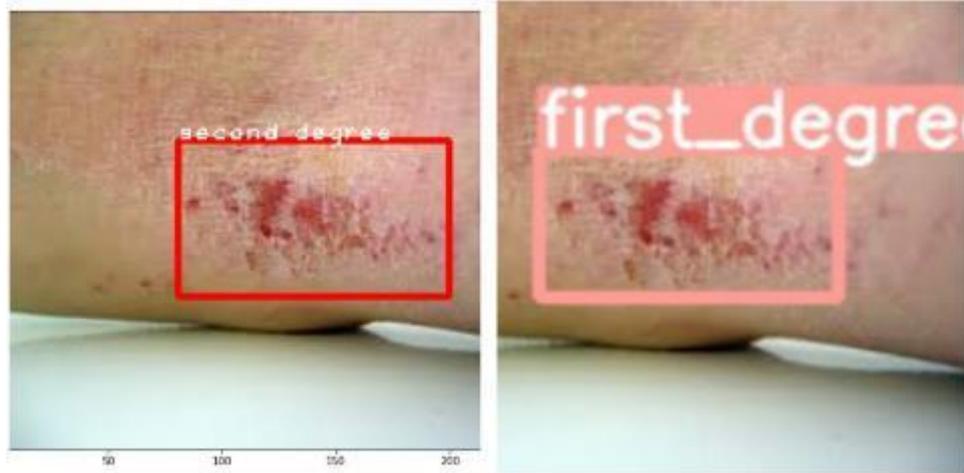


Рисунок 117 – Результаты Yolov5s(640x640)



Рисунок 118 – Результаты Yolov5s(640x640)

4.3 Yolov5x(256x256)

На рисунке 119 приведены гиперпараметры Yolov5x(256x256) модели: размер изображения к которому мы приводим все наши изображения, количество батчей, количество эпох.

```
!python train.py --img 256 --batch 64 --epochs 30 \
  --data ./data/burns.yaml --cfg ./models/yolov5x.yaml --weights yolov5x.pt \
  --name yolov5x_burns --cache
```

Рисунок 119 – Задание модели Yolov5x(256x256)

На рисунке 120 приведены ошибки и точности на каждой эпохе на тренировочных и валидационных данных модели Yolov5x(256x256). Что означает каждая метрика описано выше.

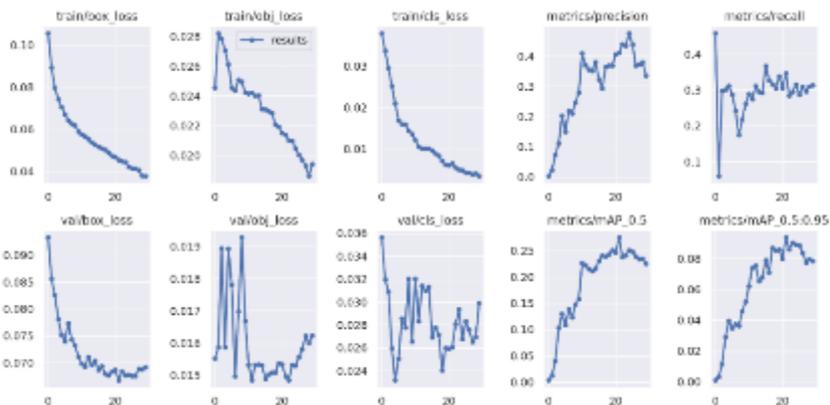


Рисунок 120 – Ошибки и точности модели Yolov5x(256x256)

Как видно из графиков box_loss, obj_loss, cls_loss на train и val падают, а precision, recall, mAP растут.

Несмотря на хороший тренд, результаты получились плохими, низкая точность и полнота, возможно это происходит из-за очень маленького размера выборки.

Результаты еще хуже, чем на прошлом алгоритме.

На рисунке 121 представлена зависимость F1 score от уверенности модели Yolov5x(256x256).

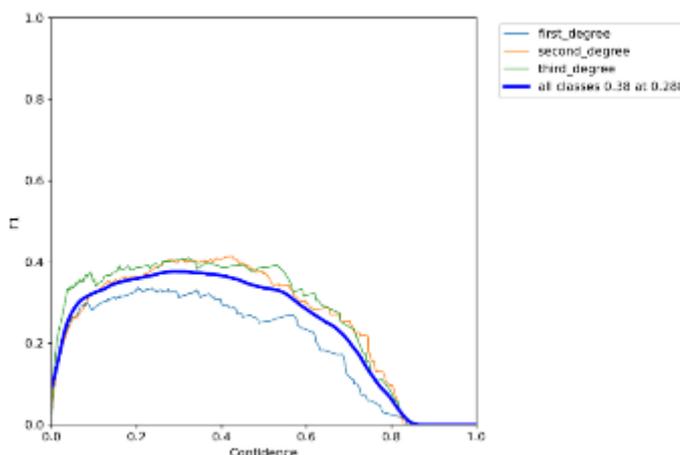


Рисунок 121 – Зависимость F1 score от уверенности модели Yolov5x(256x256)

В среднем лучший F1 score достигается при 0.3 confident.

На рисунке 122 представлена зависимость Precision от Recall модели Yolov5x(256x256).

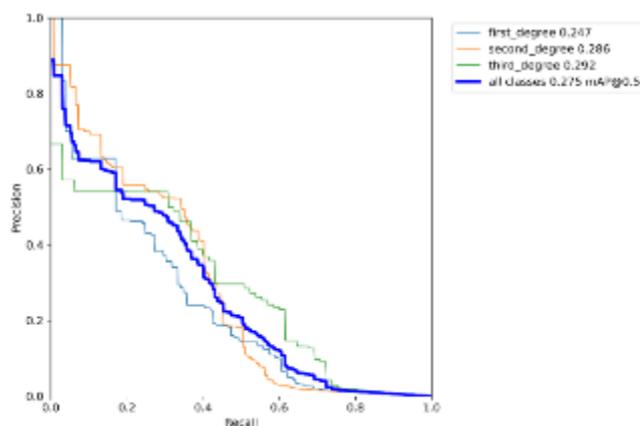


Рисунок 122 – Зависимость Precision от Recall модели Yolov5x(256x256)

mAP низкая.

На рисунках рисунок 123 и 124 представлены результаты работы нейросети. Левая картинка реальная метка и бокс, правая – предсказанное.



Рисунок 123 – Результаты Yolov5x(256x256)



Рисунок 124 – Результаты Yolov5x(256x256)

4.4 Yolov5x(640x640)

На рисунке 125 приведены гиперпараметры Yolov5x(640x640) модели: размер изображения к которому мы приводим все наши изображения, количество батчей, количество эпох.

```
!python train.py --img 640 --batch 16 --epochs 30 \
  --data ./data/burns.yaml --cfg ./models/yolov5x.yaml --weights yolov5x.pt \
  --name yolov5x burns --cache
```

Рисунок 125 – Задание модели Yolov5x(640x640)

На рисунке 126 приведены ошибки и точности на каждой эпохе на тренировочных и валидационных данных модели Yolov5x(640x640). Что означает каждая метрика описано выше.

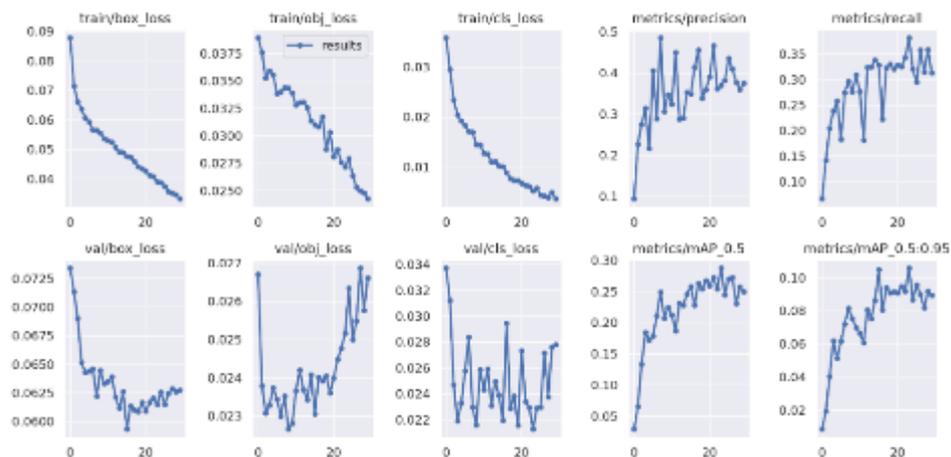


Рисунок 126 – Ошибки и точности модели Yolov5x(640x640)

Как видно из графиков box_loss, obj_loss, cls_loss на train и val падают, а precision, recall, mAP растут.

Несмотря на хороший тренд, результаты получились плохими, низкая точность и полнота, возможно это происходит из-за очень маленького размера выборки.

Результаты получились самые лучшие.

На рисунке 127 представлена зависимость F1 score от уверенности модели Yolov5x(640x640).

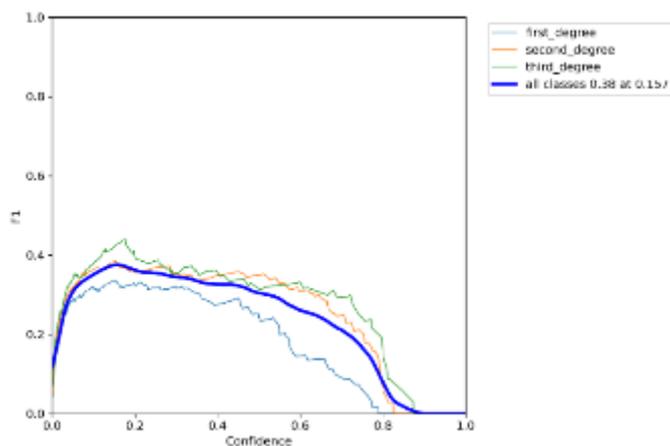


Рисунок 127 – Зависимость F1 score от уверенности модели Yolov5x(640x640)

В среднем лучший F1 score достигается при 0.19 confident.

На рисунке 128 представлена зависимость Precision от Recall модели Yolov5x(640x640).

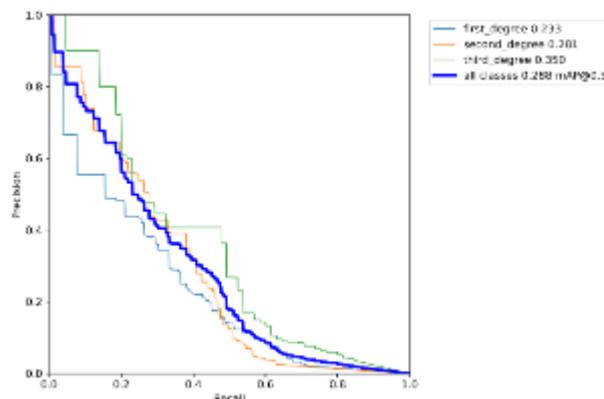


Рисунок 128 – Зависимость Precision от Recall модели Yolov5x(640x640)

mAP низкая.

На рисунках рисунок 129 и 130 представлены результаты работы нейросети. Левая картинка реальная метка и бокс, правая – предсказанное.

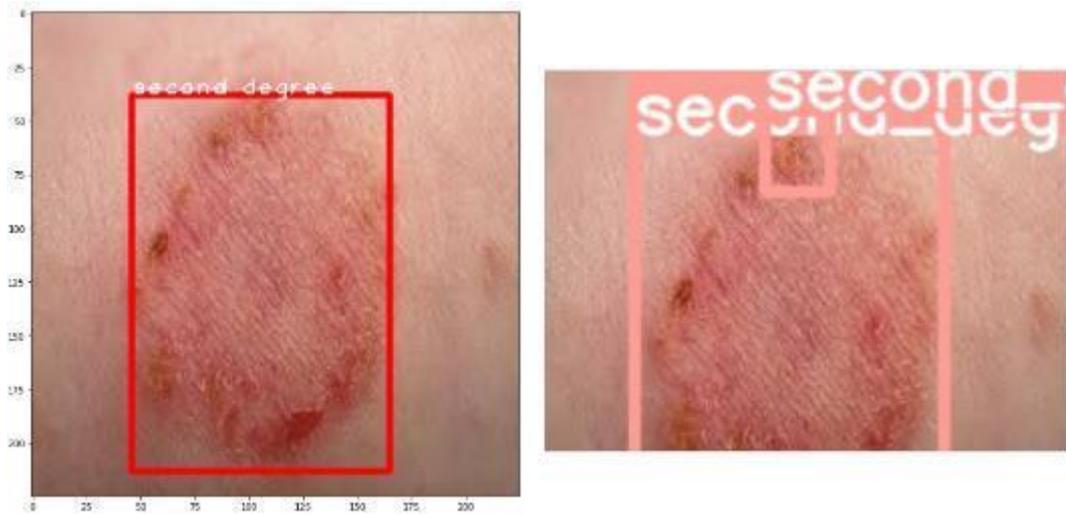


Рисунок 129 – Результаты Yolov5x(640x640)

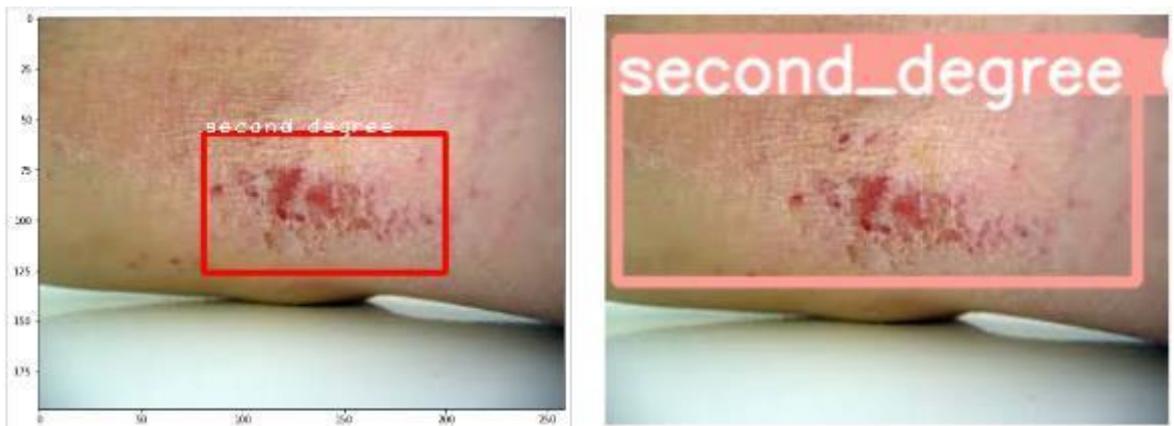


Рисунок 130 – Результаты Yolov5x(640x640)

4.5 Yolov5m(640x640)

На рисунке 131 приведены гиперпараметры Yolov5m(640x640) модели: размер изображения к которому мы приводим все наши изображения, количество батчей, количество эпох.

```
!python train.py --img 640 --batch 32 --epochs 45 \
--data ./data/burns.yaml --cfg ./models/yolov5m.yaml --weights yolov5m.pt \
--name yolov5m_burns --cache
```

Рисунок 131 – Задание модели Yolov5m(640x640)

На рисунке 132 приведены ошибки и точности на каждой эпохе на тренировочных и валидационных данных модели Yolov5m(640x640). Что означает каждая метрика описано выше.

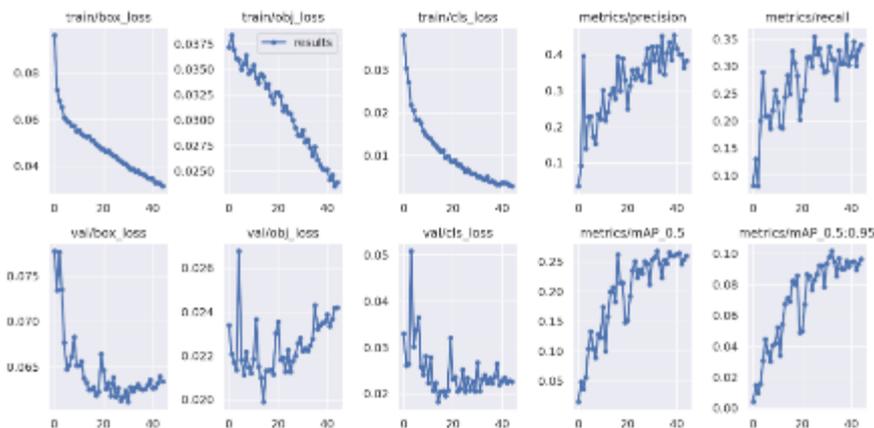


Рисунок 132 – Ошибки и точности модели Yolov5m(640x640)

Как видно из графиков box_loss, obj_loss, cls_loss на train и val падают, а precision, recall, mAP растут.

Несмотря на хороший тренд, результаты получились плохими, низкая точность и полнота, возможно это происходит из-за очень маленького размера выборки.

На рисунке 133 представлена зависимость F1 score от уверенности модели Yolov5m(640x640).

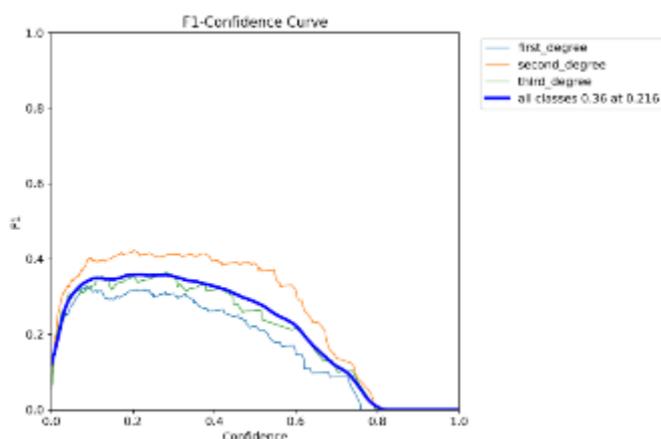


Рисунок 133 – Зависимость F1 score от уверенности модели Yolov5m(640x640)

В среднем лучший F1 score достигается при 0.2 confident.

На рисунке 134 представлена зависимость Precision от Recall модели Yolov5m(640x640).

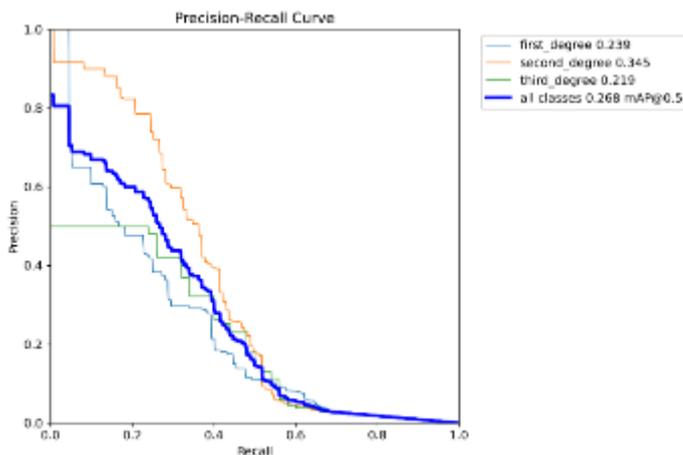


Рисунок 134 – Зависимость Precision от Recall модели Yolov5m(640x640)

mAP низкая.

На рисунках рисунок 135 и 136 представлены результаты работы нейросети. Левая картинка реальная метка и бокс, правая – предсказанное.



Рисунок 135 – Результаты Yolov5m(640x640)



Рисунок 136 – Результаты Yolov5m(640x640)

4.6 Вывод

Самые лучшие оценки показал Yolov5x(640x640). По правилам на каждый класс должно быть 1500 изображений у меня было намного меньше, эпох должно было быть 300 у меня было 30, batchsize должен был быть большим так как small batch sizes produce poor batchnorm statistics у меня для Yolov5x(640x640) было 16, это мало, но больше на видеокарту не влезло, Yolov5x(256x256) было 64.

ВЫВОД ПО НАУЧНОЙ РАБОТЕ

В результате работы можно заключить, что поставленные задачи были выполнены. Алгоритм со 100% вероятностью определяет аномалии, используя vae. Также с помощью EfficientNet_V2_L удалось достичь accuracy = 0.68, macro avg = 0.63, weighted avg = 0.68. Это довольно неплохо. С помощью Unet и PAN удалось достичь точности в 0.84. Результат хороший. Yolov5x(640x640) решает задачу классификации и детекции лучше всех, среди всех моделей из ultralytics, однако f1-score = 0.38, precision = 0.45, recall = 0.35 и mAP = 0.288. Это низки показатели. Это произошло из-а того, что очень мало было данных.

Список источников

- 1 ResNet-18 convolutional neural [Электронный ресурс]. – Режим доступа : https://www.mathworks.com/help/deeplearning/ref/resnet18.html#mw_591a2746-7267-4890-8390-87ae4dc7204c_sep_mw_6dc28e13-2f10-44a4-9632-9b8d43b376fe –Дата доступа: 21.11.2022.
- 2 VGG Very Deep Convolutional Networks (VGGNet) – What you need to know [Электронный ресурс]. – Режим доступа : <https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/#:~:text=The%20VGG16%20model%20achieves%20almost,models%20submitted%20to%20ILSVRC%2D2014.> –Дата доступа: 25.11.2022.
- 3 EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling [Электронный ресурс]. – Режим доступа: <https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy>

[and.html#:~:text=EfficientNet%20Performance&text=In%20general%2C%20the%20EfficientNet%20models,by%20an%20order%20of%20magnitude](#) – Дата доступа: 22.11.2022.

4 Review — PAN: Pyramid Attention Network for Semantic Segmentation (Semantic Segmentation) [Электронный ресурс]. – Режим доступа : <https://medium.com/mlearning-ai/review-pan-pyramid-attention-network-for-semantic-segmentation-semantic-segmentation-8d94101ba24a> – Дата доступа: 25.11.2022.

5 YOLOv5 [Электронный ресурс]. – Режим доступа: <https://github.com/ultralytics/yolov5> – Дата доступа: 27.11.2022.

UDC

A MEDICAL APPLICATION FOR CLASSIFYING SPOTS ON THE SKIN USING NEURAL NETWORKS

Yermalovich D.S., Vladymtsev V.D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Annotation. With the help of machine learning, a medical application has been developed that allows people to instantly determine the type and lesion of a mole, as well as determine the degree and location of the burn. Also, for more accurate classification and segmentation, an algorithm was developed that checks the input data for anomalies.

Keywords. Machine learning, mathematics, computer vision, classification, detection, segmentation, anomalies, data analysis.

УДК 004.891.2

ОГРАНИЧЕНИЯ РОСТА МОЩНОСТЕЙ НЕЙРОННЫХ СЕТЕЙ: ФИЗИЧЕСКИЕ И ЭНЕРГЕТИЧЕСКИЕ АСПЕКТЫ

*Касьян В. А., студент гр.253501, Ахметов Р. Я., студент гр.253502,
Сенько Н. С., студент гр.253502, Внук О.М., магистрант гр. 225941, Владимирцев В. Д.,
ассистент каф. Информатики*

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Владимирцев В. Д. – ассистент каф. Информатики

Аннотация. В данной научной работе рассматриваются ограничения роста мощностей нейронных сетей, вызванные физическими и энергетическими ограничениями современных компьютеров. Проводится анализ влияния энергопотребления, роста стоимости и роста производительности компьютеров на развитие нейронных сетей, а также предлагаются пути оптимизации и возможные решения для снижения энергоемкости и стоимости обучения нейронных сетей.

Ключевые слова: Нейронные сети, ограничения роста, мощность, обучение нейронных сетей, энергоэффективность, архитектура нейронных сетей, вычислительные ресурсы, технологические ограничения, энергопотребление оптимизация нейронных сетей, производительность.

Появление нейронных сетей стало одним из наиболее важных достижений в области искусственного интеллекта за последнее время. Нейронные сети - это алгоритм машинного обучения, который имитирует структуру человеческого мозга при выявлении закономерностей в данных. Они продемонстрировали замечательную эффективность в самых разных приложениях, таких как распознавание изображений и речи, обработка естественного языка и автономное вождение.

Со временем, развитие микроэлектронной промышленности привело к значительному увеличению количества транзисторов на интегральных схемах. Этот рост количества транзисторов был сопровождается их уменьшением и повышением тактовой частоты. Закон Мура, сформулированный в 1965 году Гордоном Муром, предсказывал, что количество транзисторов на интегральной схеме будет удваиваться примерно каждые два года. В течение нескольких десятилетий этот прогноз оставался достаточно точным.

Большие модели — это нейронные сети с большим количеством параметров. Из-за своих размеров эти модели требуют значительного объема вычислительных ресурсов для обучения и оптимизации, что делает вычислительную мощность решающим фактором при работе с ними. На заре разработки нейронных сетей обучение модели было трудоемким и ресурсоемким процессом, что делало его непрактичным для большинства исследователей. Однако с появлением параллельных вычислений и использованием графических процессоров (GPU) обучение моделей стало значительно быстрее. Это привело к тому, что все больше исследователей смогли присоединиться к этой области и расширили сферу своих применений. Возросшая доступность больших моделей также привела к более сложным и детальным исследованиям, при этом такие модели тестируются и дорабатываются. В конечном счете это привело к повышению производительности и более точным прогнозам в различных областях, включая обработку естественного языка, распознавание изображений и речи.

Тем не менее, существуют физические и энергетические ограничения, связанные с дальнейшим ростом мощности нейронных сетей. Как уже упоминалось ранее, закон Мура сталкивается с проблемами, вызванными уменьшением размеров транзисторов и увеличением тактовой частоты. Эти ограничения влияют на производительность и энергопотребление вычислительных систем, используемых для обучения и работы с нейронными сетями.

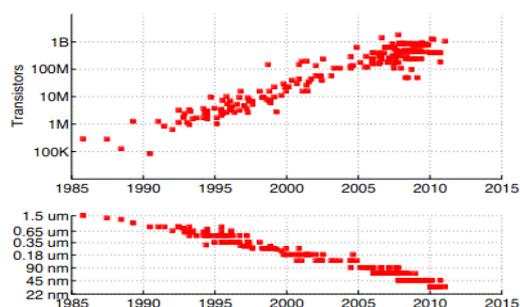


Рисунок 1 – Рост количества транзисторов и их уменьшение с течением времени

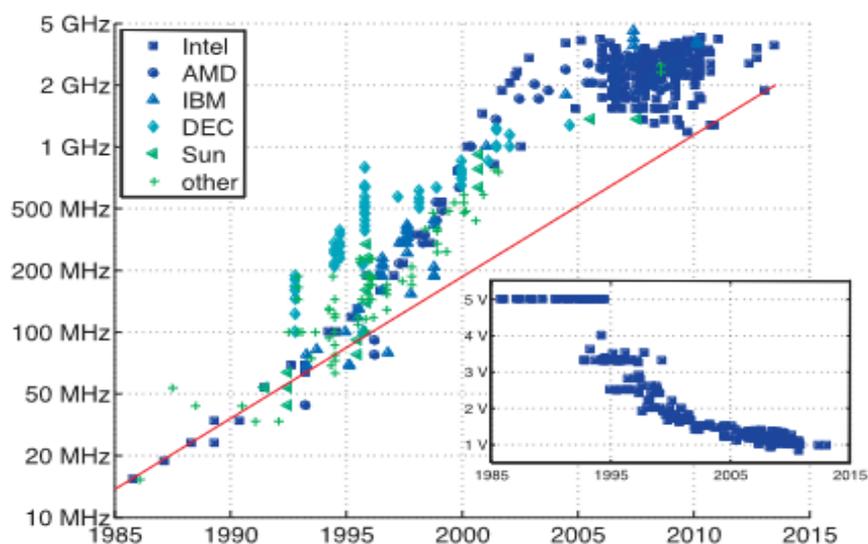


Рисунок 2 – Увеличение тактовой частоты с течением времени

Для преодоления этих проблем, исследователи и разработчики активно изучают новые подходы и технологии, такие как квантовые вычисления, нейроморфные компьютеры и специализированные архитектуры для обработки нейронных сетей, например, Tensor Processing Units (TPU) от Google. Кроме того, исследуются методы оптимизации архитектуры нейронных сетей, снижения их энергопотребления и увеличения эффективности обучения.

Вместе с тем, разработка алгоритмов машинного обучения, требующих меньше вычислительных ресурсов и энергии, также является важным направлением исследований в области нейронных сетей. Прогресс в этой области позволит сократить затраты на обучение моделей и сделать их доступными для широкого круга пользователей, что в свою очередь может способствовать дальнейшему расширению применения нейронных сетей в различных отраслях.

Использование графических процессоров (GPU) в машинном обучении стало крупным достижением за последние годы. Обеспечивая высокую степень распараллеливания вычислений, графические процессоры могут обрабатывать большие объемы данных и выполнять множество вычислений одновременно, что делает их хорошо подходящими для задач с высокой вычислительной нагрузкой, таких как обучение нейронных сетей.

На заре вычислений на графических процессорах исследователи часто использовали игровые графические процессоры, предназначенные для потребителей, для обучения своих моделей. Однако по мере того, как нейронные сети становились все крупнее и сложнее, для удовлетворения спроса на более быстрые и мощные вычисления стало необходимо специализированное оборудование, такое как тензорные процессоры Google (TPU). Это открыло новую эру в машинном обучении, характеризующуюся разработкой специализированных аппаратных и программных средств, предназначенных для полного использования мощности графических процессоров и других высокопроизводительных вычислительных ресурсов.

Со временем вычислительная мощность, необходимая для обучения сложных нейронных сетей, возросла. Это обусловлено необходимостью достижения самых современных результатов в области обработки естественного языка, компьютерного зрения и распознавания речи. Более

крупные модели требуют больше энергии для обучения, но прогресс в алгоритмах и растущая доступность данных позволили исследователям продолжать раздвигать границы возможного. Однако сам объем требуемой вычислительной мощности может стать проблемой для многих организаций. Чтобы решить эту проблему, необходимы дальнейшие исследования для разработки более эффективных алгоритмов, которые позволяют получать самые современные результаты с меньшими затратами энергии. Кроме того, были бы полезны более доступные вычислительные ресурсы, такие как облачные сервисы или специализированное оборудование.

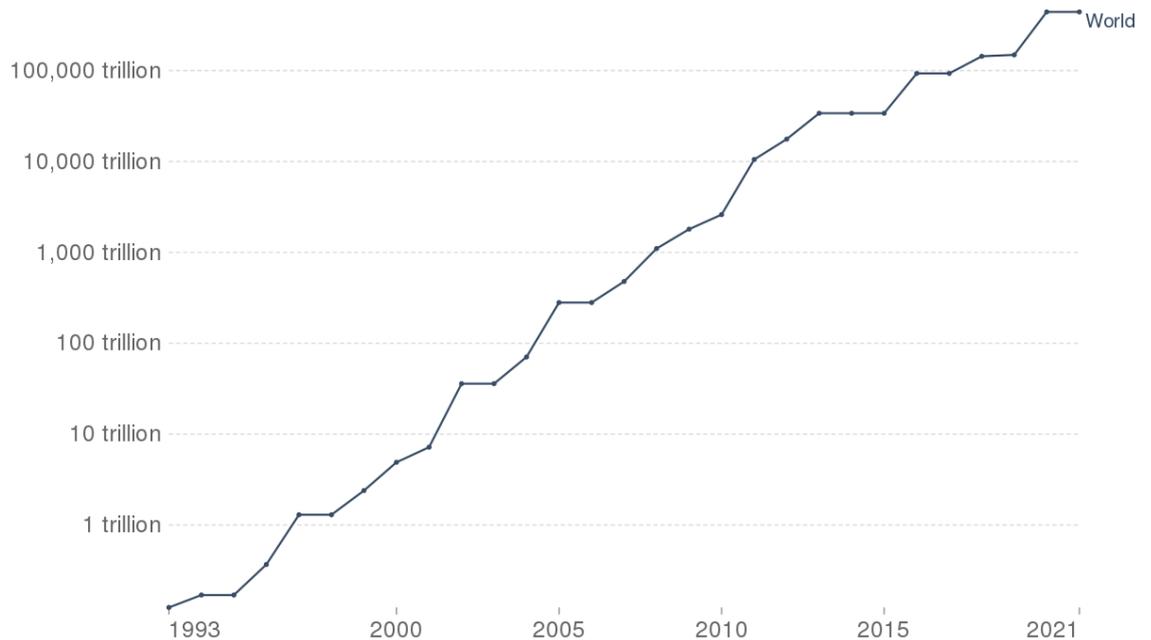


Рисунок 3 – Количество операций с плавающей точкой в секунду, которые могут выполнить суперкомпьютеры

Чтобы дать некоторое представление о количестве вычислительной мощности, необходимой для обучения нейронных сетей, давайте рассмотрим модель BERT в качестве примера. BERT (Bidirectional Encoder Representations от Transformers) - это модель обработки естественного языка с 340 миллионами параметров. Google обучал BERT, используя кластер из 4096 процессоров в течение четырех дней. Это служит ярким примером того, как вычислительная мощность, необходимая для обучения моделей, значительно увеличилась в масштабе.

Чтобы привести более свежий пример, GPT-3 (Generative Pre-trained Transformer 3), модель обработки естественного языка, представленная OpenAI в 2020 году, имеет ошеломляющие 175 миллиардов параметров, что делает ее одной из крупнейших моделей на сегодняшний день. Обучение такой масштабной модели требует значительного объема вычислительной мощности и ресурсов, что делает ее доступной только для организаций, обладающих необходимой инфраструктурой. Поскольку нейронные сети продолжают увеличиваться в размерах и усложняться, важно учитывать последствия их энергопотребления и изучать способы сделать их более устойчивыми и эффективными.

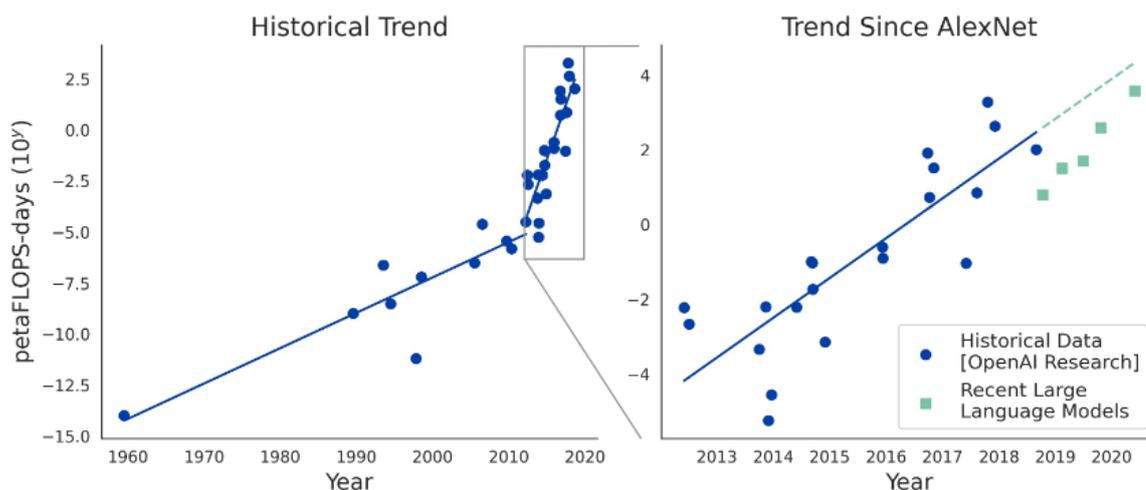


Рисунок 4 – Увеличение количества вычислений для обучения нейронных сетей с течением времени

Тенденция к созданию более крупных моделей в машинном обучении привела к значительному увеличению объема вычислительной мощности, необходимой для обучения этих моделей. Развитие параллельных вычислений и использование графических процессоров сделали возможным обучение моделей, которые когда-то считались невозможными. Поскольку нейронные сети продолжают развиваться, мы можем ожидать, что спрос на вычислительную мощность будет продолжать расти. Это поднимает важные вопросы о воздействии машинного обучения на окружающую среду и необходимости разработки более энергоэффективного оборудования.

Также стоит упомянуть проблему чрезмерного энергопотребления нейронных сетей. Нейронные сети, особенно крупные модели, требуют значительного количества энергии для обучения и функционирования, что может негативно сказаться на окружающей среде и экономической составляющей их использования. С 2000 года стоимость обучения нейронных сетей резко выросла, и этот рост продолжается до сих пор. Несмотря на то, что нейронные сети были изобретены еще в 1950-х годах, они долгое время не были особенно популярны из-за ограничений вычислительной техники и недостаточно большого количества данных для обучения. Однако с развитием технологий и появлением больших объемов данных, нейронные сети стали все более популярными, и сегодня они используются в самых разных областях: от распознавания речи до анализа изображений и машинного перевода.

Одним из главных факторов, определяющих стоимость обучения нейронных сетей, является объем данных, необходимых для обучения. С увеличением объема данных растет и стоимость их сбора, обработки и хранения. Кроме того, обучение нейронной сети требует высокопроизводительных компьютеров с большим количеством вычислительных ресурсов, которые также являются дорогостоящими.

Еще одним фактором, влияющим на стоимость обучения нейронных сетей, является квалификация специалистов, занимающихся этой работой. Обучение нейронных сетей требует высокой квалификации и опыта в области алгоритмов машинного обучения и вычислительной техники, что в свою очередь ведет к высоким затратам на оплату труда таких специалистов.

Стоимость обучения нейронных сетей также зависит от сложности задачи, которую необходимо решить. Некоторые задачи, такие как распознавание речи или анализ изображений, требуют большого количества данных и сложных алгоритмов обработки, что ведет к более высокой стоимости обучения.

Несмотря на рост стоимости обучения нейронных сетей, их популярность продолжает расти, поскольку они являются мощным инструментом для решения самых разных задач. Более того, с развитием технологий и увеличением количества данных, точность и эффективность нейронных сетей продолжает увеличиваться, что делает их еще более привлекательными для использования в различных областях.

Вместе с тем, рост стоимости обучения нейронных сетей также стимулирует развитие новых методов и технологий, направленных на уменьшение затрат на обучение. Например, в последние годы активно развивается область автоматизированного машинного обучения (AutoML), которая позволяет автоматически настраивать и оптимизировать нейронные сети без необходимости вручную настраивать их параметры. Это позволяет сократить время и затраты на обучение и делает использование нейронных сетей более доступным для широкого круга пользователей.

Таким образом, хотя стоимость обучения нейронных сетей продолжает расти с 2000 года, их популярность и значимость не уменьшаются. С появлением новых технологий и методов,

направленных на сокращение затрат на обучение, они становятся все более доступными для использования в различных областях и могут стать ключевым инструментом в решении сложных задач.

Потребляемая мощность стала тем фактором, который заставляет разработчиков суперкомпьютеров переосмысливать их архитектуру. Поскольку отдельные узлы суперкомпьютера потребляют все больше электроэнергии и выделяют все больше тепла, их необходимо разносить в пространстве и интенсивно охлаждать. Без использования экзотических систем охлаждения перегрев приведет к тому, что суперкомпьютеры просто не смогут выполнять нужные исследователям приложения. К сожалению, стоимость построения нестандартных охлаждающих систем может быть вполне сопоставима со стоимостью самого суперкомпьютера, а их обслуживание обойдется еще дороже.

Система	Число процессоров	Показатели устойчивости и готовности
ASC Q	8192	Среднее время между перерывами в работе: 6,5 часа, 114 незапланированных случаев выхода из строя в течение месяца. Источники сбоев: система хранения, процессоры, оперативная память
ASC White	8192	Среднее время между отказами: 5 часов (2001) и 40 часов (2003). Источники сбоев: система хранения, процессоры, оборудование третьих фирм.
PSC Lemieux	3016	Среднее время между перерывами в работе: 9,7 часа. Уровень готовности: 98,33%
Google (предположительно)	450000	600 перезагрузок в день; 2-3% оборудования, подлежащего замене в течение года. Источники сбоев: система хранения и память. Уровень готовности: ~100%
Источник информации: D.A. Reed		

Рисунок 5 – таблица зависимости количества сбоев в работе суперкомпьютера от числа процессоров

Как показано во врезке «Энергетическая эффективность Green Destiny», суперкомпьютер с низким энергопотреблением оказался удивительно устойчивым в работе. За два года не было зарегистрировано ни одного отказа. Согласно результатам опроса, опубликованным в 2001 году компанией Contingency Planning Research, стоимость часа простоя такого компьютера варьируется от 90 тыс. долл. при выполнении им операций, связанных с организацией продаж по каталогу, до 6,5 млн долл. при выполнении брокерских операций. Нет никаких гарантий того, что суперкомпьютер никогда не сломается, и это наглядно проиллюстрировано в таблице. А также общая стоимость владения такой техникой значительно превышает первоначальную стоимость ее приобретения.

Машинное обучение находится на пути к тому, чтобы потреблять всю поставляемую энергию, и такая модель является дорогостоящей, неэффективной и неустойчивой.

В значительной степени это объясняется тем, что данная область является новой, чрезвычайно интересной и быстро развивающейся. Она разрабатывается для того, чтобы открыть новые горизонты в плане точности или возможностей. Сегодня это означает большие модели и большие обучающие наборы, что требует экспоненциального роста вычислительных возможностей и потребления огромного количества энергии в центрах обработки данных как для обучения, так и для выводов. Кроме того, умные устройства начинают появляться повсюду.

Но цифры потребляемой мощности начинают пугать людей. На недавней конференции по автоматизации проектирования технический директор AMD Марк Пейпермастер представил слайд, показывающий энергопотребление систем ML в сравнении с мировым производством энергии:

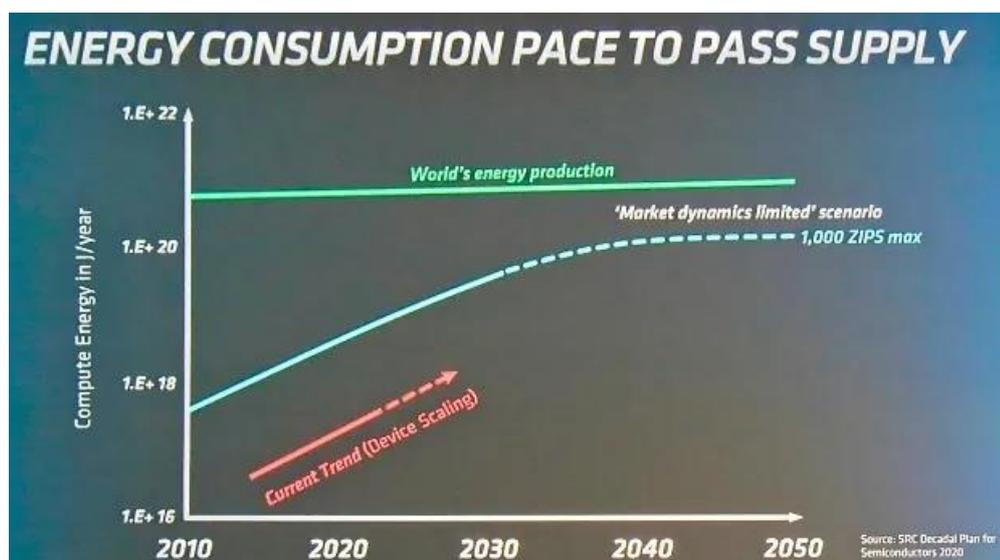


Рисунок 6 – слайд с презентации Марка Пейпермастера

Компания Papermaster не единственная, кто бьет тревогу. "Мы забыли, что движущей силой инноваций в течение последних 100 лет была эффективность", - говорит Стив Тейг, генеральный директор Perceive. "Именно это привело к появлению закона Мура. Сейчас мы живем в эпоху анти эффективности".

А Аарт де Геус, председатель совета директоров и генеральный директор компании Synopsys, от имени планеты Земля призвал сделать что-то с этим. "Тот, у кого есть мозги, чтобы понять, должен иметь сердце, чтобы помочь".

Почему потребление энергии растет так быстро? "Вычислительные потребности нейронных сетей ненасытны", - говорит Ян Братт, научный сотрудник и старший директор по технологиям компании Arm. Чем больше сеть, тем лучше результаты и тем больше проблем вы можете решить". Энергопотребление пропорционально размеру сети. Поэтому энергоэффективный вывод абсолютно необходим для внедрения все более сложных нейронных сетей и расширенных сценариев использования, таких как приложения для работы с голосом и зрением в реальном времени".

К сожалению, не все заботятся об эффективности. "Если посмотреть на то, что пытаются сделать компании-гиперскейлеры, то они пытаются получить более качественное и точное распознавание голоса, речи, рекомендательные движки", - говорит Тим Векслинг, старший вице-президент по продуктам и развитию бизнеса компании Mythic. "Это денежный вопрос. Чем выше точность, тем больше клиентов они могут обслужить и получить большую прибыль". Если посмотреть на обучение в центре обработки данных и вывод этих очень больших моделей NLP, то именно там потребляется много энергии. И я не знаю, есть ли реальная мотивация для оптимизации энергопотребления в этих приложениях".

Но некоторым людям не все равно. "Существует некоторое коммерческое давление, направленное на снижение углеродного воздействия этих компаний, не прямое денежное, а скорее на то, что потребитель примет только углеродно-нейтральное решение", - говорит Александр Уэйкфилд, ученый из Synopsys. "Это давление со стороны "зеленой" энергетики, и если один из этих поставщиков заявит, что он нейтрален к выбросам углекислого газа, то больше людей, скорее всего, будут использовать его".

Но не вся энергия потребляется в области облачных вычислений. Растет число умных периферийных устройств, которые также вносят свой вклад в эту проблему. "Существуют миллиарды устройств, составляющих IoT, и в какой-то момент в недалеком будущем они будут потреблять больше энергии, чем мы производим в мире", - говорит Марси Вайнштейн, директор по стратегическому и техническому маркетингу компании Aspinity. "Они потребляют энергию для сбора, передачи и выполнения любых действий с данными, которые они собирают".

С увеличением мощности и сложности современных нейронных сетей, важность энергоэффективности и оптимизации становится все более актуальной. Решение экологических и экономических проблем, связанных с ростом энергопотребления нейронных сетей, требует совместных усилий со стороны исследователей, инженеров и промышленности. Одним из направлений оптимизации является проработка новых архитектур нейронных сетей, которые могут обеспечить более эффективное использование вычислительных ресурсов и снижение энергопотребления. Например, применение разреженных нейронных сетей или использование квантовых компьютеров для обработки нейросетевых моделей может значительно уменьшить

затраты на энергию. Кроме того, исследователи активно разрабатывают методы сжатия и ускорения нейронных сетей, которые позволяют уменьшить размер моделей и, таким образом, снизить потребление ресурсов. Это включает техники, такие как кластеризация весов, прореживание и квантизация. Важным аспектом является также разработка и применение более энергоэффективного оборудования, такого как специализированные архитектуры для обработки нейронных сетей (ASIC), которые могут существенно снизить энергопотребление по сравнению с традиционными графическими процессорами (GPU). С другой стороны, увеличение стоимости нейронных сетей может стать препятствием для их широкого внедрения, особенно для малых и средних предприятий. В этом контексте важно разработать стратегии и программы, направленные на увеличение доступности нейросетевых технологий для всех участников рынка, включая предоставление обучающих материалов, программ финансирования и поддержки разработчиков. Исследования в области энергоэффективных нейронных сетей и оптимизации их стоимости являются важными шагами для обеспечения долгосрочной устойчивости и широкого распространения нейронных сетей в различных отраслях и приложениях. При этом необходимо продолжать разрабатывать и внедрять новые методы обучения, которые будут использовать меньше вычислительных ресурсов и энергии. Это может включать использование переноса обучения с подкреплением, мета обучения и других подходов, которые могут сократить время и стоимость обучения нейросетевых моделей. Создание открытых исходных кодов и общедоступных ресурсов, таких как предварительно обученные модели, базы данных и инструменты для работы с нейронными сетями, также будет способствовать доступности нейросетевых технологий для широкого круга пользователей. Это может стимулировать инновации и разработку новых решений на основе нейронных сетей, которые будут учитывать экологические и экономические ограничения.

В заключение, преодоление физических и энергетических ограничений роста мощности нейронных сетей потребует совместных усилий со стороны исследователей, инженеров, промышленности и законодателей. Разработка новых алгоритмов, технологий и аппаратных решений, а также учет экологических и экономических факторов, будет ключом к долгосрочной устойчивости и успешному развитию этой перспективной области искусственного интеллекта.

Список использованных источников:

1. Medium[Электронный ресурс]. – Режим доступа: URL:<https://medium.com/illumination/gpt-3-vs-gpt-4-987872f48ecf> (дата обращения: 11.04.2023).
2. Medium[Электронный ресурс]. – Режим доступа: URL:<https://medium.com/illumination/meta-llama-vs-chatgpt-a-detailed-comparison-9794ccedd41c> (дата обращения: 11.04.2023).
3. Stadtherr M. A. *High performance computing: Are we just getting wrong answer faster? //CAST division awards banquet, Miami Beach, Florida.* – 1998.
4. Lohn A., Musser M. *AI and Compute: How Much Longer Can Computing Power Drive Artificial Intelligence Progress //Center for Security and Emerging Technology.* <https://doi.org/10.51593>. – 2022.
5. Ву-Чун Фен, Кирк Камерон *Green500: рейтинг энергетической эффективности* [Электронный ресурс]. // Открытые системы: изд. научн. Журн. – Режим доступа: URL:<https://www.osp.ru/os/2008/01/4839411?ysclid=ffil0b172k217199330> (дата обращения: 11.04.2023).
6. Markov I. L. *Limits on fundamental limits to computation //Nature.* – 2014. – Т. 512. – №. 7513. – С. 147-154.

UDC 004.891.2

LIMITATIONS OF NEURAL NETWORK CAPACITY GROWTH: PHYSICAL AND ENERGY ASPECTS

Kasyan V. A, Akhmetov R. Y, Senko N. S

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V. D. – Assistant of the Department of Informatics

Annotation. This scientific work explores the limitations of neural network power growth caused by the physical and energy limitations of modern computers. The analysis of the influence of power consumption, cost growth and growth of computer performance on the development of neural networks is carried out, and optimization ways and possible solutions to reduce the power consumption and cost of training of neural networks are proposed.

Keywords. Neural networks, growth constraints, power, neural network training, energy efficiency, neural network architecture, computing resources, technological constraints, energy consumption optimization of neural networks, performance.

АНАЛИЗ ЭВОЛЮЦИИ NFT ТЕХНОЛОГИИ И ЕЁ ПЕРСПЕКТИВЫ

*Азаров Е. А., студент гр. 253501, Горзун А. В., студент гр. 253501,
Шумский Д. С., студент гр. 253501*

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Стройникова Е. Д. – ст. преп. кафедры информатики

Аннотация. Статья "Эволюция NFT технологии и её перспективы" рассматривает развитие технологии NFT, которая используется для создания уникальных цифровых активов и их продажи на рынке. Мы анализируем историю и развитие NFT, начиная с первых экспериментов с использованием блокчейн технологии и заканчивая сегодняшними трендами.

Статья также исследует перспективы использования NFT в различных отраслях, таких как искусство, образование, музыка и многое другое. Мы обсуждаем, как NFT могут помочь художникам и музыкантам получить больше дохода от своих творческих работ и как они могут изменить способ взаимодействия людей с культурными ценностями.

В целом, статья представляет обзор развития NFT технологии и её перспективы в различных отраслях, а также анализирует возможные проблемы и решения, связанные с этой технологией.

Ключевые слова. NFT, криптовалюта, блокчейн, цифровые активы, децентрализация, искусство, коллекционирование, технологические инновации, интернет, экономика, правовые аспекты, дизайн, цифровые права, цифровая валюта, уникальность, прозрачность, авторские права, социальная значимость, криптовалютные рынки, экологические аспекты, цифровая аутентичность, применение NFT в различных отраслях.

Введение

Non-Fungible Token (NFT) – это цифровой невзаимозаменяемый токен, который существует в сети блокчейн – децентрализованной базе данных. Bitcoin, например, является взаимозаменяемым, это означает, что любые две единицы криптовалюты идентичны, тогда как NFT - единственный в своём роде.

NFT используются в разных сферах и могут представлять различные цифровые активы, такие как музыка, изображения, видео, предметы для видеоигр, виртуальная недвижимость, коллекционные карточки и многое другое.

История

Ранние эксперименты (2012–2017 года): идеи, стоящие за NFT, впервые появились с созданием Colored Coins на блокчейне Bitcoin. Colored Coins представляли собой маркеры, привязанные к определенным транзакциям Bitcoin и предназначенные для представления активов вне криптовалюты. В 2014 году был создан проект Counterparty, который использовал блокчейн Bitcoin для создания уникальных токенов и их обмена.

CryptoPunks и CryptoKitties (2017 год): в июне 2017 года компания Larva Labs выпустила проект CryptoPunks, состоящий из 10 000 уникальных цифровых персонажей, каждый из которых был представлен в виде NFT на Ethereum. В ноябре 2017 года появились CryptoKitties, которые стали первым вирусным проектом с использованием NFT. Пользователи могли покупать, продавать и разводить уникальных цифровых котят.

Развитие платформ и стандартов (2018–2020): стандарт ERC-721 для Ethereum стал основным стандартом для создания NFT. Он обеспечивал единый интерфейс для управления и передачи уникальных токенов. В то же время появилось множество платформ, таких как Rarible, OpenSea, SuperRare и другие, которые позволяли всем желающим покупать, создавать и продавать NFT-искусство, коллекционные предметы и другие вещи.

Экспансия и основное применение (2021 год): в 2021 году произошел бурный рост популярности и стоимости NFT. Знаменитости, художники и спортивные организации начали использовать NFT для продажи цифрового искусства, музыки и видео и других товаров. Проект NBA Top Shot, например, представил коллекционные карточки с видео-моментами из баскетбольных матчей в форме NFT.

Повседневное использование и интеграция (2021–2023): в это время NFT стали интегрироваться в различные области повседневной жизни и бизнеса. Некоторые примеры включают:

- Виртуальную реальность и метавселенную: NFT используются для представления цифровых объектов, земельных участков и аватаров в виртуальных мирах, таких как Somnium Space и The Sandbox и другие.
- Игры: во многих видеоиграх NFT используются для создания уникальных игровых предметов, персонажей и внутриигровой валюты.
- Цифровую моду и дизайн: NFT позволяют дизайнерам и брендам продавать уникальную цифровую одежду, которая может быть использована в метавселенной или в соцсетях.

- Финансовые услуги и страхование: NFT могут быть использованы для представления и обмена активами, такими как акции, облигации, недвижимость и произведения искусства, а также для создания страховых полисов и других финансовых инструментов.

Принцип работы NFT

Понимание NFT с технологической точки зрения требует знания нескольких основных концепций:

- Реестры, обычно блокчейны, которые служат общей и надежной записью информации. Другими словами, они служат «слоем доверия» и «единым источником правды».
- Криптографические ключи, которые передают контроль над NFT и обеспечивают основные функции, такие как шифрование и аутентификация.
- Идентификаторы токенов используются для различения NFT в экземпляре смарт-контракта в блокчейне. Каждый идентификатор токена связан с уникальным адресом (устанавливая инфраструктуру открытого ключа) и с метаданными. Таким образом, разные стороны могут легко находить друг друга и взаимодействовать друг с другом, а также извлекать выгоду из уникальных свойств блокчейнов, таких, как неизменность.
- Метаданные, которыми может быть что угодно (например, произведение цифрового искусства, цифровое представление физического товара). Важно отметить, что метаданные могут храниться как в цепочке, так и вне ее.
- Смарт-контракты, которые можно рассматривать как программы или приложения, работающие на блокчейне и отвечающие за «чеканку» (создание) NFT.
- Кошельки, которые используются для хранения ключей (управления ключами) или потенциально даже метаданных. Кроме того, они позволяют управлять и обмениваться NFT с помощью простых в использовании приложений.

Процесс создания NFT включает в себя преобразование изображений, видео и других цифровых файлов в криптоактивы на блокчейне. На данный момент существует несколько блокчейнов с инструментами для создания токенов, таких как TRON, EOS и т. д. Немаловажную роль в создании NFT играет смарт-контракт, он включает в себя набор функций, необходимых для создания, передачи токенов и управления ими. Смарт-контракт обязан придерживаться стандартов, таких как ERC-721, и других. Имея контракт, необходимо разместить его в блокчейне и создать токен.

Блокчейн привнес взаимозаменяемость в цифровое пространство и позволил обменивать стоимость на аналогичную в форме криптовалют, а теперь обеспечивает невзаимозаменяемость в цифровом пространстве в форме NFT.

Именно неизменность, надежность, прозрачность и децентрализация блокчейна создают новые возможности для беспрепятственной торговли, обмена и совместного использования цифровых представлений любых видов активов, и для многих предприятий это, вероятно, будет стимулировать новые инновации на основе блокчейна.

В отличие от стандартов для взаимозаменяемых токенов, ERC-721 предлагает возможность включать расширенные метаданные об элементе, такие как историческое право собственности. Это также позволяет каждому токеноу иметь отдельные свойства, которые отличают его от других токенов. Эти свойства подробно описаны в файле вне цепочки, который доступен через метаданные токена.

Этот стандарт в настоящее время стимулирует дальнейшие инновации для создания системы правил, библиотек и соглашений для цифровых сертификатов на блокчейне и облегчит компаниям изучение и использование потенциала NFT.

Анализ популярности NFT

После волны популярности NFT в середине 2021 – середине 2022 года кажется, что все про него забыли и никому он не нужен. Рассмотрим статистику запросов “NFT” в Google.

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

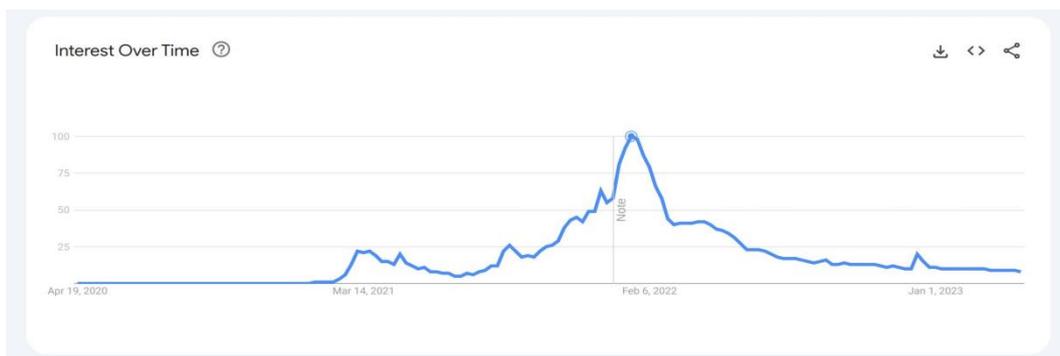


Рисунок 1 – График запросов слова “NFT” в поисковике Google

Заметим, что самый пик был в феврале 2022 года и дальше все пошло на спад. Также был маленький скачок популярности в январе 2023 года.

Также приведем статистику, основанную на еженедельном объеме торгов NFT в долларах США за последние 2 года.

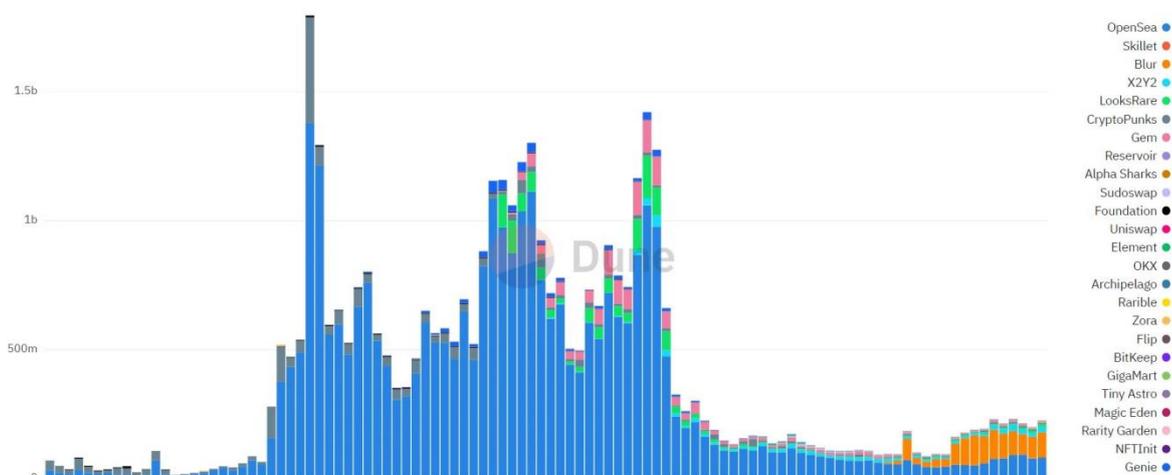


Рисунок 2 – График еженедельного объема торгов NFT в долларах США за последние 2 года.

Тут также видно большая популярность NFT в середине 2021 до середины 2022 года. И маленький скачок в январе 2023 года.

Но есть еще одна занимательная статистика. Она показывает еженедельное количество сделок NFT за последние 2 года.

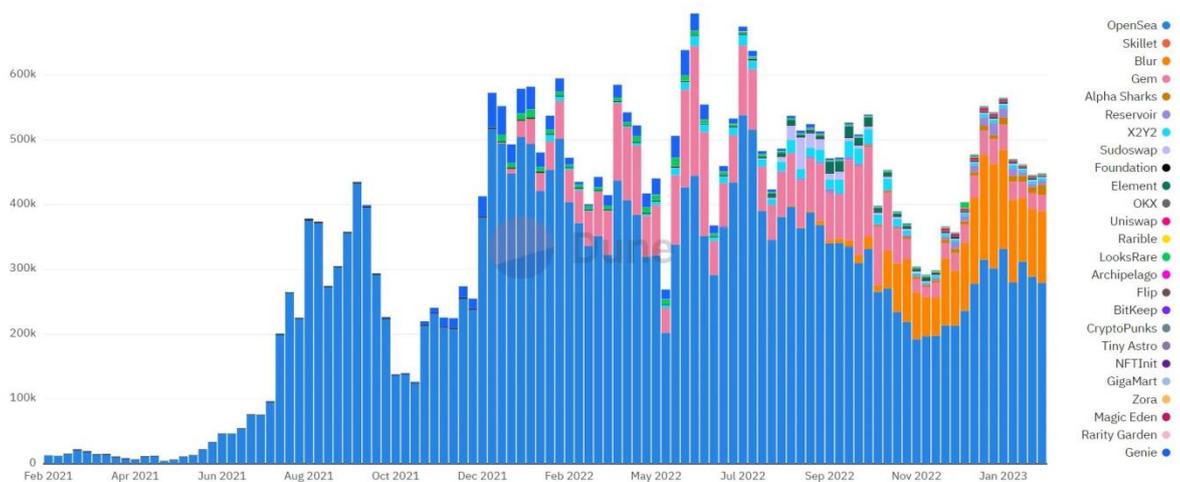


Рисунок 3 – График еженедельного количества сделок NFT за последние 2 года.

Можно увидеть, что количество операций с января 2021 по сегодняшний день было примерно на одном уровне (400 тысяч). На основе количества сделок можно понять, что NFT до сих пор используется по всему миру и не забывается обществом и то, что технология прочно закрепилась

на цифровом рынке и не собирается покидать его. А следовательно оно повлияет на развитие WEB 3.0

Перспективы развития NFT технологии

NFT можно применить в различных сферах повседневной жизни:

- Децентрализованные финансы (DeFi): NFT могут быть применены в DeFi-сфере для создания новых финансовых продуктов, таких как займы, обеспеченные цифровыми активами, или инвестиции в токенизированные активы.
- Образование и профессиональные сертификаты: NFT могут использоваться для представления образовательных достижений или профессиональных сертификатов, упрощая процесс проверки квалификации и истории обучения.
- Личная идентификация и управление данными: NFT могут быть использованы для создания децентрализованных систем идентификации, где пользователи смогут контролировать свои личные данные и предоставлять доступ к ним только выбранным сторонам.
- Защита от фальсификации на различных голосованиях: каждому участнику голосование выдается уникальный NFT. Это поможет предотвратить многократное участие в выборе одного и того же участника, так как каждый NFT будет связан с уникальным идентификатором.
- Управление событиями и билетами: NFT могут быть использованы для создания уникальных и неподдельных билетов на мероприятия, которые могут быть легко проверены, перепроданы или переданы другим лицам.
- Лояльность и бонусные программы: компании могут использовать NFT для создания индивидуальных и уникальных программ лояльности, предоставляя клиентам возможность собирать, обменивать и покупать награды при помощи специальных токенов.
- Развитие децентрализованных социальных сетей: NFT могут стать основой для новых поколений социальных сетей, где пользователи будут взаимодействовать, создавать и обмениваться уникальным контентом, при этом контролируя и монетизируя свои данные и активы.
- Реклама и маркетинг: бренды могут использовать NFT для создания уникальных цифровых предметов, которые предложены в качестве подарков, бонусов или эксклюзивных предложений, усиливая взаимодействие с потребителями и лояльность бренда.

Заключение

В мире, который становится все более цифровым, необходимо представлять собственность и управление ею цифровым способом. NFT позволяют это осуществить. В перспективе NFT технология будет продолжать эволюционировать, предлагая больше новых возможностей. Вероятно расширение применения NFT за пределы цифровых активов и вовлечение новых сфер, таких как образование, наука и медицина. Таким образом, NFT являются мощным и многообещающим инструментом, способным изменить подход к управлению и обмену цифровыми активами, а также создать новые возможности для индустрий и пользователей.

Список использованных источников:

1. Seoyoung K., Tiana L. *NFTs For Dummies [Текст] / Seoyoung K., Tiana L. — 1. : For Dummies, 2021 — 272 с.*
2. Роман С. *Что такое NFT простыми словами / Роман С. [Электронный ресурс] // R-Media : [сайт]. — URL: <https://www.raiffeisen-media.ru/money/cto-takoe-nft-prostymi-slovami/>*
3. *Introduction to NFTs for Identity. / [Электронный ресурс] // Walt.id : [сайт]. — URL: <https://walt.id/white-paper/nfts-for-identity>*
4. *Ren & Heinrich NFT Market Analysis 2023 / Ren & Heinrich [Электронный ресурс] // Medium : [сайт]. — URL: <https://ren-heinrich.medium.com/nft-markets-analysis-2023-d88e48f008d8>*

EVOLUTION OF NFT TECHNOLOGY AND ITS PROSPECTS

Harhun A. V., Shumskiy D. S., Azarov E. A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Stroinikova E. D. – Senior Lecturer in Department of Informatics

Annotation. The article "Evolution of NFT technology and its prospects" examines the development of NFT technology, which is used to create unique digital assets and sell them on the market. We analyze the history and development of NFT, starting with the first experiments using blockchain technology and ending with today's trends.

The article also explores the prospects of using NFT in various industries, such as art, education, music and much more. We discuss how NFTs can help artists and musicians earn more income from their creative work and how they can change the way people interact with cultural values.

In general, the article provides an overview of the development of NFT technology and its prospects in various industries, as well as analyzes possible problems and solutions related to this technology.

Keywords. NFT, cryptocurrencies, blockchain, digital assets, decentralization, art, collectibles, technological innovations, Internet, economics, legal aspects, design, digital rights, digital currency, uniqueness, transparency, copyrights, social significance, cryptocurrency markets, environmental aspects, digital authenticity, application of NFT in various industries.

УДК

ВЛИЯНИЕ GPT НА СПОРТИВНОЕ ПРОГРАММИРОВАНИЕ И ОБУЧЕНИЕ

*Тимошевич К. С., студент гр.253503, Сорока Д. Ф., студент гр.253503,
Владыцев В. Д., ассистент каф. Информатики*

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Владыцев В. Д. – ассистент каф. информатики

Аннотация. В данной научной работе будет рассмотрена модель искусственного интеллекта GPT и ее версии, развитие данной платформы, положительные и отрицательные стороны ее использования в качестве учебного помощника и инструмента для решения задач, отношение преподавателей к применению этой технологии учащимися и студентами, а также влияние данной платформы на сферу программирования в современных реалиях. Подробно будет рассмотрено влияние GPT на олимпиадное движение, в частности спортивное программирование, процесс обучения в учреждениях среднего образования и технических ВУЗах.

Ключевые слова: GPT, ChatGPT, спортивное программирование, олимпиадные задачи, искусственный интеллект, обучение, ICPC.

В современном мире технологии играют все большую роль в нашей жизни. Каждый день происходят новые открытия в области информационных технологий и разработок, которые делают нашу жизнь более комфортной. Одним из таких достижений является чат GPT.

GPT (Generative Pre-training Transformer) — это модель искусственного интеллекта, разработанная OpenAI, которая обучается на большом наборе текстовых данных. Существует несколько версий GPT, включая GPT, GPT-2, GPT-3 и GPT-4. Основное различие между этими версиями — объем данных, на которых она обучалась.

ChatGPT стал незаменимым инструментом для участников спортивного программирования, благодаря следующим возможностям: способность анализировать огромные объемы данных, производить сложные вычисления, оптимизировать уже написанный код, решать несложные задачи. Более того, участники, овладевшие искусством работы с ChatGPT превосходили тех, кто данную платформу не использовал.

Одной из самых знаменитых участниц международных соревнований в сфере олимпиадного программирования стала Аиша, принимавшая участие в десятках мероприятий подобного рода. Девушке удалось разработать свой собственный алгоритм для работы с ChatGPT, используя который, ей удавалось решать трудные задачи на олимпиадах.

Целью этой талантливой программистки стало участие в International Collegiate Programming Contest или ICPC. Данная олимпиада состоит из задач разной сложности. Некоторые могут быть решены с помощью несложных алгоритмов, а некоторые требуют передовых методов и многолетний опыт.

Во время соревнования Аиша использовала ChatGPT для поиска наиболее оптимизированного и быстрого решения, анализа данных. Это позволило ей и ее команде решить одну из самых сложных задач на этом соревновании, а решение, предоставленное ими, являлось гораздо быстрее и оптимизированнее, чем решения, когда-либо предоставленные другими.

Количество команд, использующих ChatGPT на олимпиадах по спортивному программированию, со временем начало возрастать. Это оказало сильное влияние как на сложность предоставляемых задач, так и на развитие платформы GPT: решения, предоставляемые чатом, стали более эффективными.

К основным преимуществам платформы также можно отнести точность в анализе данных, уменьшение количества ошибок в решениях, предоставляемых GPT, по сравнению с предыдущими версиями, легкодоступность и простота в использовании платформы для людей из разных стран, возможность использования ChatGPT для обучения: изучение оптимальных алгоритмов для решения задач разных типов, структур данных, возможность платформы генерировать задачи для тренировки студентов, искать нужную информацию, что заметно сократит время, потраченное на ее поиски. Однако у задач, сгенерированных платформой также есть и свои недостатки: решения не всегда удовлетворяют требованиям и выдают ожидаемый результат.

Пример решения чатом одной из олимпиадных задач.

Задача А. Дорога в Студсовет

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Егор проходит отбор в студсовет ФКСиС. В студсовете состоит $n - 1$ человек. Егор условно пронумеровал их от 2 до n . Председатель студсовета имеет номер n . Чтобы попасть в студсовет, Егор должен набрать как можно больше баллов. i -й студсоветчик начисляет Егору a_i баллов за выполненное задание. А также некоторые из студсоветчиков могут направить Егора к коллегам, чтобы он продолжил выполнять задание у кого-нибудь из них. Так, если студсоветчик с номером k может направить Егора к коллегам, он называет ему два числа: l_k и r_k . Это означает, что Егор может выбрать любого студсоветчика с номером $q \in [l_k, r_k]$ и пойти к нему. Без направления Егор идти не может.

Себя Егор обозначил номером 1.

Помогите Егору просчитать свои шансы. Посчитайте максимальное количество баллов, которое Егор может набрать. Если Егор не сможет добраться до председателя студсовета, выведите «No» (Без кавычек).

Рисунок 1 – Условие олимпиадной задачи

```
#include <iostream>
#include <vector>
#include <queue>
using namespace std;
const int MAXN = 1e5 + 5;
int n, m;
int a[MAXN];
vector<pair<int,int>> adj[MAXN];
bool can_reach(int s, int t) {
    vector<bool> vis(n+1, false);
    queue<int> q;
    q.push(s);
    vis[s] = true;
    while (!q.empty()) {
        int u = q.front();
        q.pop();
        if (u == t) {
            return true;
        }
        for (auto [v, w] : adj[u]) {
            if (!vis[v]) {
                vis[v] = true;
                q.push(v);
            }
        }
    }
    return false;
}

int main() {
    cin >> n >> m;
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    for (int i = 1; i <= m; i++) {
        int k, l, r;
        cin >> k >> l >> r;
        for (int j = l; j <= r; j++) {
            adj[k].push_back({j, a[j]});
        }
    }
    int s = 1, t = n;
    if (!can_reach(s, t)) {
        cout << "No\n";
        return 0;
    }
    vector<int> dist(n+1, -1e9);
    dist[s] = a[s];
    priority_queue<pair<int,int>> pq;
    pq.push({dist[s], s});
    while (!pq.empty()) {
        auto [d, u] = pq.top();
```

```

pq.pop();
d = -d;
if (d > dist[u]) {
    continue;
}
for (auto [v, w] : adj[u]) {
    if (dist[v] < dist[u] + w) {
        dist[v] = dist[u] + w;
        pq.push({-dist[v], v});
    }
}
}
cout << dist[t] << "\n";
return 0;
}

```

#	Date	Participant	Problem	Compiler	Verdict
25291	3 minutes ago	Upsolving: [BSUIR] WomenMoment	A. Дорога в Студсовет	C++20 (gcc-12.2.1)	MLE 9

Рисунок 2 – Результат отправки решения, сгенерированного ChatGPT

Как видно на втором скриншоте, код, предоставленный GPT, не является оптимизированным, так как программа смогла пройти только часть тестов, предоставленных платформой BSUIR Open.

Также к недостаткам ChatGPT можно отнести вероятность отсутствия желания у разработчиков развивать собственную креативность при решении задач, так как частое использование данной платформы может привести к тому, что программисты перестанут самостоятельно решать задачи, излишне полагаясь на GPT; хотя платформа и является одним из лучших инструментов для работы, она все же остается ограниченной в поиске инновационных и оптимизированных решений.

Для того, чтобы разобраться, как именно ChatGPT повлиял на сферу образования, для начала хотелось бы рассмотреть процесс развития платформы более подробно.

В 2020 году профессором Университета Северной Каролины Полом Файфе был проведен эксперимент с использованием ChatGPT. Целью эксперимента было научить студентов «критической грамотности в сфере данных». Студенты должны были написать эссе с помощью чата GPT-2. Текст, предложенный GPT, было разрешено копировать в любом объеме. Около 87 процентов участвовавших студентов заявили, что платформа мешала им работать: им казалось, что алгоритм подрывает их креативность, невозможно было понять, какие части были им опущены, многие факты, предоставленные чатом, были из несуществующих источников.

На данный момент ChatGPT получил бан на самой крупной платформе для разработчиков StackOverflow за многочисленные ошибки при ответах на вопросы. Платформа StackOverflow опирается на пользователей, которые решают, являются ли ответы, предоставленные другими разработчиками, точными, однако большая часть ответов, которые были сгенерированы ChatGPT, не являлись верными, поэтому сообществом StackOverflow было принято решение о временном запрете на публикацию ответов, предоставленных ChatGPT. Ввиду отсутствия верификации с авторитетными источниками и какой-либо подтвержденной базой знаний модель может очень подробно и серьезно отвечать на совершенно бессмысленные вопросы, не оценивая их реалистичность, что еще раз ставит под сомнение правильность предоставляемых платформой ответов.

14 марта 2023 года была выпущена усовершенствованная версия GPT-4. Она лучше распознает контекст и точнее придерживается заданной ей стилистики ответов. По мнению ее создателей, GPT-4 стала креативнее, гибче подстраивается под пользователя и эффективнее работает с «тонкими сценариями». Еще одним важным преимуществом 4-й версии нейросети стала ее усовершенствованная способность сдавать экзамены и тесты по разным предметам. Она показала отличные результаты по ряду дисциплин, ощутимо превзойдя показатели своей предшественницы GPT-3.5, не говоря уже об успехах среднестатистического человека.

Также стоило бы отметить успехи ChatGPT в сфере программирования. СМИ сообщили, что данная платформа успешно прошла собеседование в Google на должность младшего инженера-программиста, на которую обычно берут тех, кто только что закончил университет и не имеет большого опыта в программировании. На данной позиции требуется уметь писать рабочий код, запускать его локально, использовать инструменты проверки предоставленного кода. В рамках тестового собеседования внутри компании ChatGPT обошел по основным показателям языковую

модель LaMDA от Google, ответив на все технические вопросы. А вот с вопросами на мышление у чат-бота ChatGPT не все оказалось хорошо.

Платформа пока все же не сможет заменить программистов, так как именно они решают, как именно применять данный инструмент, основываясь на своем опыте. Таким образом на данный момент GPT отлично подойдет для решения типовых задач, поиска довольно очевидных ошибок в коде, генерирование идей для ученических научных, курсовых работ, что все же является большим прорывом для GPT.

Так ли плохо влияние GPT на обучение? Во время выполнения лабораторных работ у студентов могут возникнуть некоторые трудности. Многие студенты, столкнувшись с трудной задачей, прибегают к интернет-ресурсам или тому же ChatGPT, чтобы все же разобраться с проблемой.

Мы попросили ChatGPT выполнить одно из заданий нашей лабораторной работы. Задание выглядит следующим образом:

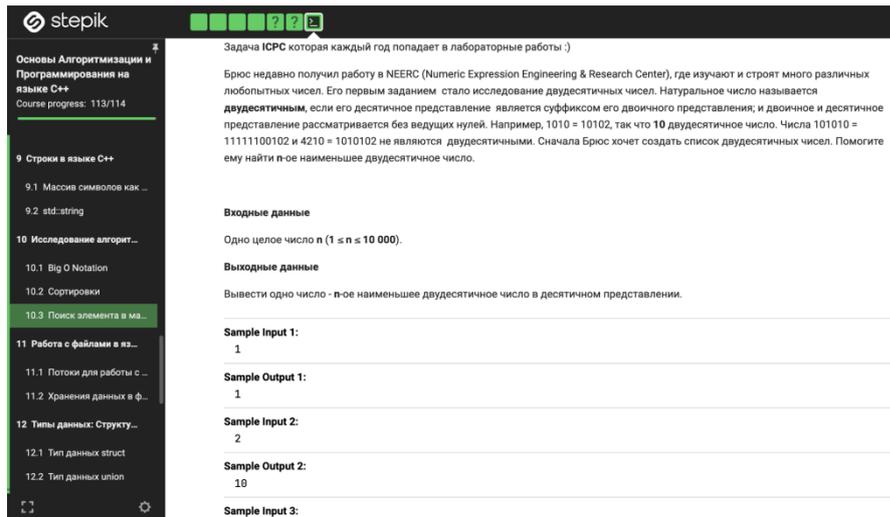


Рисунок 3 – Условие олимпиадной задачи

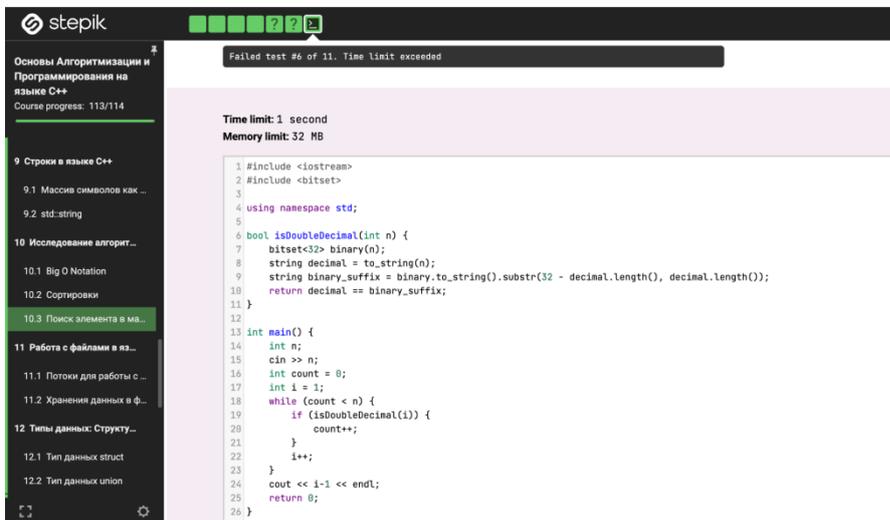


Рисунок 4 – Результат отправки решения, сгенерированного ChatGPT

Однако ответ, предоставленный чатом не удовлетворял требованиям задания ввиду того, что решение не прошло часть тестов, предоставленных сайтом. С другой стороны это дало основу для выполнения данного задания.

Исходя из этого можно сделать вывод о том, что решения предоставляемые ChatGPT не всегда корректны из-за неточности запросов, отправляемых платформе. Однако уже студентам предстоит решить, как именно использовать данный материал: разобрать ответ и, если нужно, оптимизировать и использовать в дальнейшем или бездумно скопировать решение.

С момента появления чат-бота прошло не так много времени, но некоторые университеты быстро отреагировали на изменения. Например, на сайтах MIT и других высших учебных заведений уже можно найти подробные гайды для преподавателей по формированию заданий, с которыми справятся только люди, и включению использования бота в программу.

Есть множество вариантов применения преподавателями ChatGPT в образовательном процессе. С помощью данной платформы можно быстро обработать большие объемы материала, чтобы создать на их основе курс или инструкцию, также она может создавать тестовые задания, проверочные и лабораторные работы и ответы к ним. Однако у такого подхода к использованию платформы в данной области есть и существенные недостатки по составлению материалов. Одним из самых существенных является то, что не вся информация в интернете содержит достоверные сведения, а GPT не способен создавать новую информацию и проверять уже существующую, поэтому важно проверять все материалы, предлагаемые ChatGPT. Не первостепенным минусом является невозможность в некоторых ситуациях корректно сформулировать запрос, поэтому требуется время, чтобы научиться правильно работать с платформой.

Список использованных источников:

1. "Sparks of Artificial General Intelligence: Early experiment with GPT-4" – научная статья [Электронный ресурс]. – Режим доступа: <https://arxiv.org/pdf/2303.12712.pdf> – Дата доступа 02.04.2023. – С. 21.
2. Habr – сообщество IT-специалистов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/news/t/715278/>. – Дата доступа: 02.04.2023.
3. Skillbox Media – сообщество профессионалов [Электронный ресурс]. – Режим доступа: <https://skillbox.ru/media/education/novaya-praktika-kak-chatgpt-ispolzuuyut-v-shkolakh-i-vuzakh-s-polzoy/>. – Дата доступа: 02.04.2023.
4. "How to cheat on your final paper: Assigning AI for student writing" – научная статья [Электронный ресурс]. – Режим доступа: <https://www.depts.ttu.edu/tlpc/AssigningAIForStudentWriting2021.pdf>. – Дата доступа: 02.04.2023.
5. Habr – сообщество IT-специалистов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/708646/>. – Дата доступа: 02.04.2023.

UDC

IMPACT OF GPT ON COMPETITIVE PROGRAMMING AND EDUCATION

Timoshevich K. S., Soroka D. F.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V.D. – Assistant of the Department of Informatics

Annotation. This scientific work will consider the GPT artificial intelligence model and its versions, the development of this platform, the positive and negative aspects of its use as an educational assistant and a tool for solving problems, the attitude of teachers to the use of this technology by students, as well as the impact of this platform on the field of programming in modern realities. The influence of the GPT on the **contest** movement, in particular competitive programming, the learning process in institutions of secondary education and technical universities, will be considered in detail.

Keywords. GPT, ChatGPT, competitive programming, olympic tasks, artificial intelligence, learning.

ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ НА ОСНОВЕ ФИЗИЧЕСКИ НЕКЛОНИРУЕМЫХ ФУНКЦИЙ

Тимошевич К. С., студент гр. 253503, Котова К. А., студент гр. 253503, Кваченюк Я. Д., студент гр. 253502, Николайчик А. С., студент гр. 253502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Стройникова Е.Д. – старший преподаватель каф. информатики

Аннотация. В данной работе рассмотрены различные методы генерации случайных чисел (ГСЧ) на основе физически неклонированных функций (ФНФ), такие как использование шума электромагнитных помех, оптических ФНФ, а также состояния памяти устройства. Для каждого метода были описаны основные принципы работы и процесс генерации случайных чисел (СЧ).

Ключевые слова. Генераторы случайных чисел, физически неклонированные функции.

Генераторы случайных чисел (ГСЧ) являются одним из ключевых инструментов в области компьютерной науки и информатики. Они используются для моделирования случайных явлений в различных приложениях, таких как криптография, статистические вычисления, моделирование случайных процессов и т.д.

Одним из основных требований к ГСЧ является высокая степень случайности. Это означает, что любой потенциальный злоумышленник не должен иметь возможности предсказать следующее число в последовательности. Поэтому генераторы СЧ обычно основаны на алгоритмах, которые используют определенные математические функции для создания случайных чисел.

Однако существует риск того, что алгоритмы генерации СЧ могут быть подвержены атакам и взлому. Для повышения уровня безопасности генераторов случайных чисел (СЧ) могут использоваться физически неклонированные функции (ФНФ), которые создают случайные значения на основе уникальных физических свойств устройства.

Генератор случайных чисел (ГСЧ) с использованием физически неклонированных функций (ФНФ) – это метод генерации последовательности случайных чисел, который основывается на использовании уникальных физических свойств компонентов системы, таких как микрочипы, датчики, транзисторы и т.д.

Основная идея заключается в том, что каждый компонент имеет свои уникальные физические свойства, которые могут быть использованы для создания уникальных и непредсказуемых последовательностей чисел. Такие функции могут включать в себя случайные шумы, флуктуации напряжения, температуры, сопротивления, затухания и многое другое.

Физически неклонированная функция (ФНФ) – это специальный вид функций, которые генерируют уникальный и непредсказуемый результат на основе физических характеристик конкретного компонента. Они должны удовлетворять следующим свойствам:

1. Структурная информация подобных систем может быть извлечена надежно и неоднократно путем проведения измерений для различных запросов и получения ответов.
2. Количество возможных запросов должно быть настолько велико, что значения всех соответствующих ответов не могут быть получены путем перебора всех возможных запросов за реальный временной промежуток.
3. Ввиду наличия в системе чрезмерно большого объема структурной информации должно быть невозможным смоделировать, рассчитать или каким-либо другим математическим способом предсказать пару запрос-ответ, зная другую пару или некоторое множество таких пар.
4. Для физической системы с чрезвычайно большим объемом структурной информации должно быть чрезвычайно сложным ее физическое воспроизведение или клонирование как аналогичной физической системы, описываемой идентичным множеством пар запрос-ответ.

Создание генератора случайных чисел с использованием оптических ФНФ

Среди окружающих нас материальных объектов сложно найти два абсолютно одинаковых предмета. Даже в серийном производстве каждый объект получается уникальным за счет погрешностей и случайностей. Эти особенности каждого отдельного объекта можно регистрировать и использовать как уникальный идентификатор, своеобразный «отпечаток пальца».

Наглядный пример — оптическая ФНФ. Возьмем расплавленное стекло, добавим в него пузырьки воздуха, остудим эту массу и разрежем на одинаковые бруски. Шанс получить два

абсолютно одинаковых бруска ничтожно мал, т.к. пузырьки воздуха внутри будут распределены неравномерно. Можно зафиксировать эти различия, отправляя на брусок пучок лазерного излучения (запрос) и получая на выходе уникальную интерференционную картину пучков излучения после преломления (ответ). В результате получится ФНФ, которая будет определять зависимость ответа от входного запроса.

Оптические ФНФ основаны на уникальных оптических свойствах материалов, которые обеспечивают высокую степень неравенства и непредсказуемости.

Оптические ФНФ могут быть реализованы на основе различных типов материалов, включая полимеры, металлы и полупроводники. Для создания оптической ФНФ необходимо использовать специальную технологию нанесения микроструктур на поверхность материала. Микроструктуры должны быть такими, чтобы они могли отражать или рассеивать световой луч в уникальный и случайный образ.

Для генерации псевдослучайных чисел на основе оптических ФНФ используется процесс чтения оптической информации с помощью лазера. Лазерное излучение позволяет получить уникальный образ, который может быть использован для генерации случайных чисел. Образы, полученные с помощью оптических ФНФ, могут быть использованы для создания последовательности псевдослучайных чисел, которые обладают высокой степенью неравенства и непредсказуемости.

Важным преимуществом использования оптических ФНФ для генерации случайных чисел является высокая стойкость к атакам, т.к. оптические ФНФ основаны на физических свойствах материалов, которые трудно скопировать или воссоздать. Кроме того, использование оптических ФНФ позволяет получать высококачественные СЧ, которые могут применяться в различных целях.

Создание генератора случайных чисел с использованием шума

Создание генератора случайных чисел с использованием шума как источника случайности является одним из наиболее распространенных методов генерации СЧ. Шум может быть получен из различных источников, включая электрический, оптический, термальный и другие типы шумов.

В таком генераторе СЧ получаются путем измерения случайных флуктуаций в шуме. Для этого используются различные типы датчиков, которые могут быть созданы на основе различных физических явлений, таких как термальный шум, шум Джонсона – Найквиста, шум с лазера и другие.

Для обработки сигнала шума используются различные методы, включая фильтрацию, усиление, цифровую обработку и другие. После обработки сигнала получают СЧ, которые могут быть использованы для различных целей, в том числе криптографических приложений, генерации СЧ в играх, симуляций и др.

Важным преимуществом генерации СЧ на основе шума является высокая стойкость к атакам, т.к. шум является физическим явлением, которое трудно воспроизвести и контролировать. Кроме того, использование шума как источника случайности позволяет получать высококачественные СЧ, которые могут быть использованы для различных целей.

Создание генератора случайных чисел с использованием электромагнитных помех

Создание ГСЧ с использованием электромагнитных помех как источника случайности является одним из наиболее эффективных методов генерации СЧ. Электромагнитные помехи – это случайные колебания электромагнитных полей, которые могут быть получены из различных источников, включая радиоволны, шумы силовых и сигнальных кабелей, соседних электронных устройств и т.д.

Для создания ГСЧ с использованием электромагнитных помех необходимо собрать антенну или датчик, который может измерять электромагнитные помехи в окружающей среде. Датчик может быть выполнен в различных формах и может иметь различные параметры, такие как усиление и чувствительность.

Для обработки сигнала электромагнитных помех используются различные методы, включая фильтрацию, усиление, анализ спектра и другие. После обработки сигнала получают СЧ, которые могут быть использованы для различных целей, в том числе криптографических приложений, генерации СЧ в играх, симуляций и др.

Генерация СЧ на основе электромагнитных помех имеет ряд преимуществ:

- Высокая энтропия: электромагнитные помехи могут быть использованы для создания СЧ с высокой степенью случайности и энтропии, что обеспечивает надежную защиту от взлома и подделки данных.
- Доступность: электромагнитные помехи являются широко доступным источником случайности, который может быть использован на различных устройствах, включая компьютеры, смартфоны и другие электронные устройства.
- Эффективность: ГСЧ на основе электромагнитных помех могут быть реализованы с использованием простых схем, что делает их эффективными и экономически выгодными.

Создание генератора случайных чисел с использованием силы давления пользователя на экран смартфона

Создание ГСЧ на основе ФНФ с использованием силы давления пользователя на экран смартфона может быть эффективным способом генерации СЧ.

Экраны современных смартфонов часто оснащены датчиками давления, которые могут быть использованы для измерения силы, с которой пользователь нажимает на экран. Для создания ГСЧ с использованием ФНФ на основе силы давления пользователя на экран смартфона необходимо провести ряд измерений силы давления, которые генерируются пользователем при нажатии на экран. Эти данные могут быть использованы для создания уникального шаблона, который может быть преобразован в последовательность СЧ.

Для увеличения степени случайности генерируемых чисел можно проводить несколько измерений силы давления при каждом нажатии на экран. Также можно использовать дополнительные источники случайности, например шум в электрической цепи датчика давления или случайные задержки между нажатиями на экран.

Одним из преимуществ использования ФНФ на основе силы давления пользователя на экран смартфона является возможность применения этого генератора в качестве альтернативы традиционным методам генерации СЧ, которые могут быть уязвимы к атакам. Кроме того, использование ФНФ на основе силы давления пользователя на экран смартфона может быть полезным в случаях, когда требуется генерация СЧ без дополнительных устройств или оборудования.

Создание генератора случайных чисел с использованием памяти устройства

Создание ГСЧ на основе ФНФ с использованием состояния памяти устройства может быть эффективным способом генерации СЧ.

Многие устройства, такие как компьютеры, телефоны и другие электронные устройства, содержат микросхемы памяти, которые могут быть использованы для генерации СЧ на основе состояния памяти устройства. Эта техника может быть основана на том, что начальное состояние памяти устройства может быть уникальным и случайным.

Для создания ГСЧ на основе состояния памяти устройства, необходимо сначала получить начальное состояние памяти, которое может быть использовано для генерации последовательности СЧ. Это может быть достигнуто, например, путем чтения некоторого количества байтов из памяти устройства в определенный момент времени.

Для увеличения степени случайности генерируемых чисел можно использовать дополнительные источники случайности, например шум в электрической цепи микросхемы памяти или случайные значения таймеров и счетчиков устройства.

Одним из преимуществ использования ФНФ на основе состояния памяти устройства является то, что этот метод может быть применен для генерации СЧ без дополнительных устройств или оборудования. Кроме того, начальное состояние памяти устройства может быть изменено только случайным образом, что делает его очень сложным для воспроизведения или атаки. Однако, для достижения высокой степени случайности необходимо использовать достаточно большой объем данных и хорошо продуманную методику для генерации СЧ.

Использование физически неклонировемых функций для генерации случайных чисел в программе на языке С++

Для использования ФНФ с целью генерации СЧ в программе на С++ можно подключить библиотеку OpenSSL. Данная библиотека предоставляет функции, которые используют криптографически безопасные генераторы случайных чисел, основанные на ФНФ.

Вот пример программы на С++, которая использует библиотеку OpenSSL для генерации случайного числа:

```

#include <openssl/rand.h>
#include <iostream>

int main() {
    unsigned char buffer[4];
    RAND_bytes(buffer, sizeof(buffer));

    unsigned int* random_number = reinterpret_cast<unsigned int*>(buffer);
    std::cout << "Random number: " << *random_number << std::endl;

    return 0;
}

```

Рисунок 1 — Код первой программы

Здесь генерируется массив «buffer» из 4-х случайных байтов с помощью функции «RAND_bytes». Затем эти байты интерпретируются как беззнаковое целое число с помощью приведения типа «reinterpret_cast<unsigned int*>(buffer)». Полученное число выводится на экран. Несколько примеров вывода программы:

```

Random number: 1285241114

Random number: 3152517205

```

Рисунок 2, 3 — Результаты работы первой программы

Вот пример еще одной программы на C++, использующей библиотеку OpenSSL для генерации случайного числа:

```

#include <openssl/rand.h>
#include <iostream>

int main() {
    unsigned char buffer[8];
    if(RAND_bytes(buffer, sizeof(buffer)) == 0) {
        std::cerr << "Error generating random numbers!" << std::endl;
        return 1;
    }

    uint64_t random_number = *reinterpret_cast<uint64_t*>(buffer);
    std::cout << "Random number: " << random_number << std::endl;

    return 0;
}

```

Рисунок 4 — Код второй программы

В этой программе «RAND_bytes» генерирует 8 случайных байтов, которые затем интерпретируются как 64-битное беззнаковое целое число. Если при вызове «RAND_bytes» произошла ошибка, программа выведет сообщение об ошибке. В противном случае программа выведет сгенерированное случайное число.

```

Random number: 11072600854160185063

Random number: 17206203176475216913

```

Рисунок 5, 6 — Результаты работы второй программы

Если увеличить размер буфера, то и сгенерированное случайное число будет больше. Например, если использовать unsigned char buffer[8], то сгенерированное число будет 64-битным (uint64_t), что позволит представлять числа в диапазоне от 0 до $2^{64} - 1$.

В заключение можно отметить, что это направление генерации СЧ на основе ФНФ является важным для многих областей науки и технологии, включая криптографию, статистику, моделирование и другие области.

Мы рассмотрели различные методы создания ГСЧ на основе ФНФ, включая использование шума, электромагнитных помех, оптических ФНФ, а также состояния памяти устройства. Каждый из этих методов имеет свои преимущества и недостатки, и выбор метода зависит от конкретной задачи.

В целом, использование ФНФ для генерации СЧ является эффективным и надежным подходом, который может быть применен во многих областях.

Однако следует учитывать, что использование ФНФ может быть затратным с точки зрения вычислительных ресурсов и времени, что может снижать производительность системы. Кроме того, необходимо обеспечивать достаточную защиту от физических атак на генератор, т.к. любые нарушения целостности источника энтропии могут привести к компрометации всей системы.

Список использованных источников:

1. «Информатика» – научный рецензируемый журнал [Электронный ресурс]. – Режим доступа: <https://inf.grid.by/jour/article/view/370/340>. – Дата доступа: 02.04.2023.
2. Habr – сообщество IT-специалистов [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/343386/>. – Дата доступа: 02.04.2023.
3. Интернет-портал для IT-специалистов [Электронный ресурс]. – Режим доступа: <https://itnan.ru/post.php?c=1&p=343386>. – Дата доступа: 02.04.2023.
4. Новости информационных технологий [Электронный ресурс]. – Режим доступа: <https://www.pvsm.ru/kontrafakt/269336>. – Дата доступа: 02.04.2023.

UDC

RANDOM NUMBER GENERATION BASED ON PHYSICALLY NON-CLONEABLE FUNCTIONS

Timoshevich K. S., Kotova K. A., Kvachenyuk Y. D., Nikolaychik A. S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Stroinikova E. D. – Senior Lecturer in Department of Informatics

Annotation. This work examines different methods of generating random numbers (RNGs) using physically unclonable functions (PUFs), including the utilization of electromagnetic noise, optical PUFs, and device memory states. Each method is discussed in terms of its underlying principles and the process of random number generation (RNG).

Keywords. Random number generators, physically unclonable functions.

УДК 004.838

ИСПОЛЬЗОВАНИЕ AI В ИНТЕЛЛЕКТУАЛЬНЫХ ИГРАХ

Клейн М.А., студент гр.253504, Косяков М.М., студент гр.253504,
Владыцев В. Д., ассистент каф. Информатики

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Владыцев В. Д. – ассистент каф. информатики

Аннотация. Научная статья посвящена применению машинного обучения в сфере интеллектуальных играх. Выяснилось, что машина вряд ли в скором времени заменит человека, ведь она не способна выстраивать большие логические цепочки. Также было построено несколько таблиц, отображающих результаты компьютера.

Ключевые слова. Машинное обучение, «Что? Где? Когда?», интеллектуальные игры.

В современном мире интеллект стал одним из главных поисковых ресурсов. Успехи можно заметить искусственный интеллект в играх, требующих большого багажа знаний. В 2011 году машина Watson победила опытных гроссмейстеров в «Jeopardy!», являющейся аналогом «Своей игры». Система представляла программный комплекс, работающий на кластере из десяти строк, каждая из которых содержала по девять стандартных серверов.

Система Watson была предназначена для поиска ответов на самые разные вопросы и разрабатывалась в течение двух лет. Находясь на финальной стадии проектирования, она использовала процессор естественного языка и другие анализирующие технологии, которые помогали выявлять скрытый за словами смысл. Это позволяло суперкомпьютеру находить информацию, соответствующую вопросу, правильно интерпретировать игру слов и синтезировать подходящие ответы.

Вопросы интеллектуальных игр можно условно разделить на несколько категорий: логические («Что? Где? Когда?»), требующие знаний («Своя игра») и медийные (Pub-quiz).

Искусственный интеллект может противостоять игрокам в медийном поле, ведь обладает улучшенными алгоритмами поиска информации. Начинающих игроков неоднократно упрекают в использовании приложения «Shazam», способного отыскать композицию по запросу.

Но на вопросы, предполагающими обширный багаж знаний или взаимодействие с другими игроками, машина вряд ли ответит.

Стоит упомянуть, что существует игры, в которых использование гаджетов приветствуется. Задания подразумевают большое количество реалий и форм.

Логические

В ходе исследования была построена таблица, отображающая способность искусственного интеллекта отвечать на вопросы «Что? Где? Когда?». Для этого мы взяли несколько вопросов, вбили их в строку нейронной сети, а затем сравнили ответы с результатами живых игроков.

Таблица 1 – Ответы нейронной сети GPT на вопросы спортивного «Что? Где? Когда?».

Вопросы	ChatGPT	Человек	Правильный ответ
Поэт Жамбыл Жабаев сравнил шумящих у озера малышей с НИМ. Дословный перевод ЕГО казахского названия — «поющая трава». Назовите ЕГО.	Кузнецова	Камыш	Камыш
В одном из ресторанов Порто во время трапезы дважды подается одно и то же французское угощение. Какое французское название носит повторная подача угощения?	Реплика	Дежавю	Дежавю

На социальном плакате, посвящённом спасению черепах, изображены медузы и ОНИ. Назовите ИХ двумя словами, начинающимися на одну и ту же букву.	Пластиковые пакеты	Пластиковые пакеты	Пластиковые пакеты
---	--------------------	--------------------	--------------------

Как мы видим из таблицы 1, искусственный интеллект сможет справиться с простыми вопросами, но не вряд ли проделает несколько ходов. Пускай машина способна оперативно обрабатывать поисковые запросы, она вряд ли когда-нибудь заменит человека, которому свойственно логическое мышление.

Требующие знаний

Также машине было задано несколько вопросов из «Своей игры»:

Таблица 2 – Ответы нейронной сети GPT на вопросы спортивного «Что? Где? Когда?».

Вопросы	ChatGPT	Человек	Правильный ответ
В каком городе приземлился Христос, согласно названию романа Владимира Короткевича?	Берендеева	Гродно	Гродно
Назовите чешско-австрийского биолога-генетика, являющегося основоположником учения о наследственности.	Грегор Мендель	Грегор Мендель	Грегор Мендель
Каким словом называют дворянский титул, средний между бароном и графом во Франции, Англии, а также лицо, носящее этот титул?	Виконт	Маркиз	Виконт

Можно увидеть, что машина способна отыскать информацию с опорой на ключевые слова, но не справится с вопросами, требующими погружения в специфику.

Также мы заставили нейронную сеть написать вопросы для «Что? Где? Когда?», используя готовые факты:

Таблица 3 – Генерация вопросов спортивного «Что? Где? Когда?» нейронной сетью GPT

Факты	Вопросы
В основу фильма «Выпускник» легла история брата сестёр Бронте, которого соблазнила сорокалетняя миссис Робинсон. В отличие от героя фильма, Бронте не заканчивал университет, а его роман свёл в могилу.	Какой реальной истории брата и сестер Бронте была вдохновлена знаменитая кинолента «Выпускник», и как она отличается от сюжета фильма?
Бумажником называют слово, собранное из частей других слов: например, «лошак» образовано от «лошади» и «ишака»	Как называется слово, образованное из частей других слов, и приведите примеры таких слов?

В англоязычной компьютерной игре Lemmings после того, как игрок нажимает на кнопку с изображением следов лап, процесс останавливается.	Какой кнопкой в компьютерной игре Lemmings можно остановить процесс и что происходит после ее нажатия?
Тайная переписка Уильяма и Бритни в начале 2000-х, так и не переросла в нечто серьезное. Тем не менее Ульям неоднократно называл Бритни принцессой.	Тайная переписка Уильяма и Бритни в начале 2000-х, так и не переросла в нечто серьезное. Тем не менее слово ОНА неоднократно звучало в адрес Бритни. Кто такая ОНА?
В телесериале «Альф» многие эпизоды называются как известные рок-композиции. В 64 серии сериала главный герой отправляется в специализированное учреждение, чтобы признаться в любви 64 летней актрисе чёрно-белых фильмов.	Многие эпизоды телесериала «Альф» называются как известные рок-композиции. В одной из серий Альф отправляется в специализированное учреждение, чтобы признаться в любви актрисе черно-белых фильмов. Сколько ей лет, согласно названию серии?

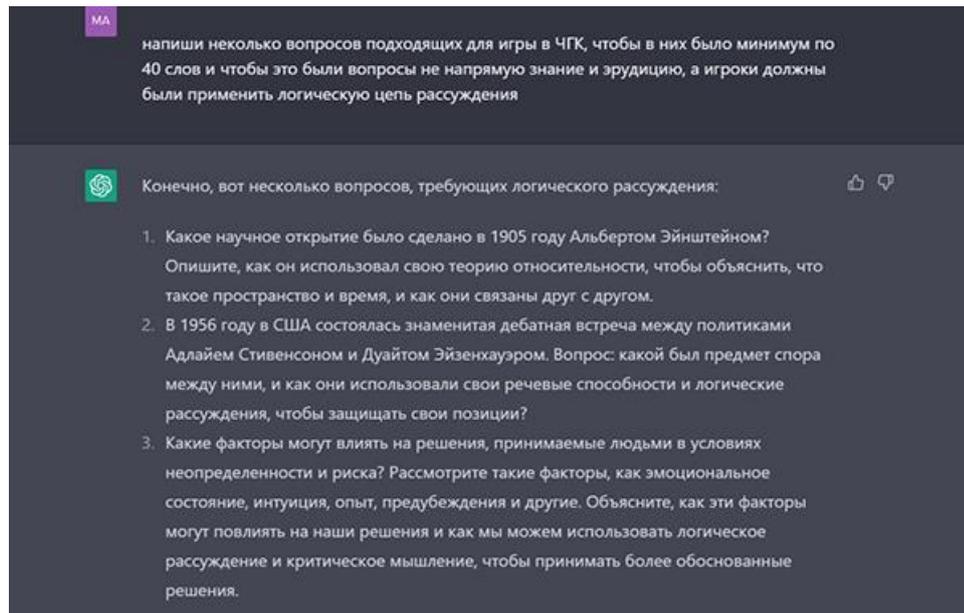


Рисунок 1 – Генерация вопросов

Чтобы разобраться в теме мы попросили комментарий у редактора «Мозгобойни» Алексея Блатуна:

Смогут ли нейросеть в ближайшем будущем самостоятельно написать вопрос?

Ответ: следует разграничивать понятие "вопрос". Текстовый – в ближайшее время хороший вопрос скорее нет. Спросить, в каком году открыли Америку – сможет, а вот заложить некоторые логические ходы – пока, видимо, нейросети это не умеют. Но с учётом скорости развития не удивлюсь, если уже через пару лет научатся.

А вот если говорить про картинку, музыку, ребусы и прочее, то нейросети уже стали для квизов массовым источником. Они пусть и не пишут сами, но выступают в роли "соавтора", давшего идею. Нарисовать героиню книги в средневековых реалиях, изобразить пословицу или поговорку, воспроизвести клип для песни – всё это нейросети умеют делать отлично, главное дать им начальный толчок. И материалы становятся хорошим дополнением к текстовым турам. В ходе исследования мы выяснили, что нейронная сеть не может заменить игроков в интеллектуальных играх, так как не обладает логическим мышлением. Несмотря на открытый доступ к «Википедии», технология не может отвечать на вопросы, требующие погружения в определённую специфику. Большинство же вопросов спортивного «Что? Где? Когда?» подразумевают наличие определённых обширных знаний, касающихся тематики вопроса. При этом нейронная сеть может помочь авторам с написанием пакета. Она не только послужит отличным поисковым ресурсом, что в том числе пригодится и на стадии апелляций, но и сгенерирует изображения по запросу пользователя.

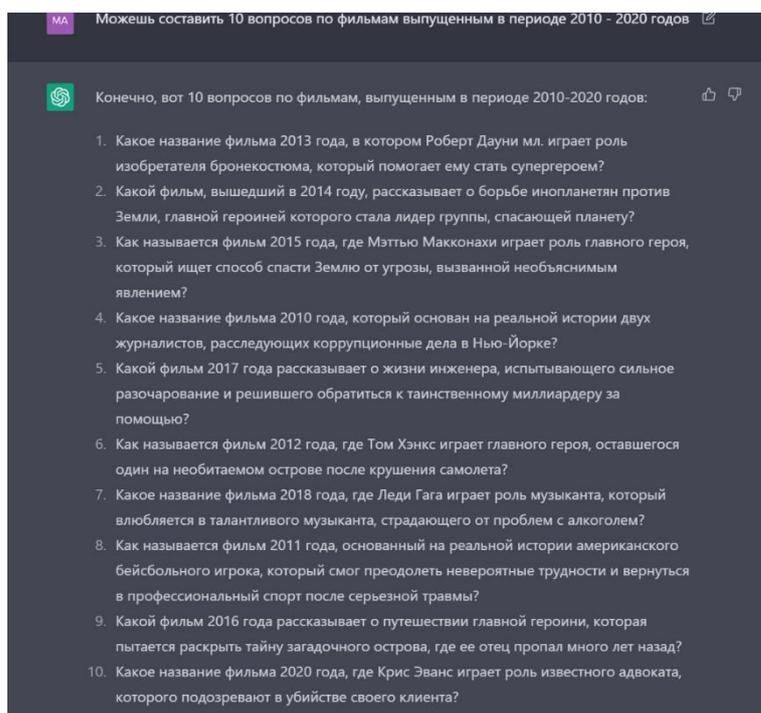


Рисунок 2 – Генерация вопросов для раунда о кино

Заключение. В ходе исследования мы выяснили, что нейронная сеть не может заменить игроков в интеллектуальных играх, так как не обладает логическим мышлением. Несмотря на открытый доступ к «Википедии», технология не может отвечать на вопросы, требующие погружения в определённую специфику. Большинство же вопросов спортивного «Что? Где? Когда?» подразумевают наличие определённых обширных знаний, касающихся тематики вопроса.

При этом нейронная сеть может помочь авторам с написанием пакета. Она не только послужит отличным поисковым ресурсом, что в том числе пригодится и на стадии апелляций, но и сгенерирует изображения по запросу пользователя. Полученные картины могут стать основой для полноценного тура, послужив дополнением к классическим раундам.

Список использованных источников:

1. Суперкомпьютер от IBM обыграл людей в викторине Jeopardy! [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/113278/>. – Дата доступа: 08.04.2023.
2. «Го играть, я создал»: как компьютеры побеждают людей. [Электронный ресурс]. – Режим доступа: <https://daily.afisha.ru/brain/580-go-igrat-ya-sozdal-kak-kompytery-pobezhdayt-lydey/>. – Дата доступа: 08.04.2023.
3. A Computer Called Watson. [Электронный ресурс]. – Режим доступа: <https://www.ibm.com/ibm/history/ibm100/us/en/icons/watson/>. – Дата доступа: 07.04.2023.

UDC 004.838

USING AI IN MIND GAMES

Klein M. A., Kosyakov M.M., Vladymtsev V. D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V. D. – Assistant of the Department of Informatics

Annotation. A scientific paper is devoted to the application of machine learning in the realm of intelligent games. It turns out that a machine is unlikely to replace a human in the near future, as it is not capable of building large logical chains. Several tables were also built displaying the results of the computer.

Keywords. Machine learning, "What? Where? When?", mind games.

УДК 159.99

СТЕГАНОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Вилкина К. А., студент гр.253504, Клебеко Е. Ю., студент гр.253504

Носкович П. Н., студент гр.253504

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Стройникова Е. Д. – ст. преп. кафедры информатики

Аннотация. В данной статье были рассмотрены методы защиты информации с использованием стеганографии, а также предложен вариант реализации метода стеганографии с наименьшим значащим битом.

Ключевые слова. Стеганография, стеганографические методы, сокрытие информации

Стеганография — это средство сокрытия секретной информации внутри обычного, несекретного документа или другого носителя, чтобы избежать обнаружения. Произошло от греческого *steganos* и *graphy*, что вместе означает “тайнопись” или “скрытое письмо”.

Стеганография и криптография преследуют одну и ту же цель — защитить сообщение или информацию от третьих лиц, но в отличие от криптографии, в стеганографии скрытым является сам факт передачи информации.

При использовании стеганографических методов защита информации происходит на трех уровнях:

1. Неизвестен сам факт передачи скрытой информации;
2. Неизвестен алгоритм помещения скрытой информации в контейнер (под контейнером подразумевается открытый текст, где скрыта зашифрованная информация);
3. Неизвестен способ кодирования информации.

Один из наиболее распространенных методов называется стеганографией с наименьшим значащим битом (LSB — *least significant bit*). Это включает в себя встраивание секретной информации в младшие значащие биты медиафайла. К примеру, изменение последнего бита значения пикселя не приводит к визуально заметному изменению изображения, а это означает, что никто не сможет отличить исходное изображение от стеганографически модифицированного.

Ниже представлен вариант реализации данного метода на базе языка Python с использованием библиотеки *steganocryptopy*. В результате активации данного кода, мы получаем ещё одну фотографию, которая внешне ничем не отличается от первоначального варианта, разница будет только в их размер, и то она минимальна. В дальнейшем, используя заранее сгенерированный ключ (строка 3), мы можем дешифровать сообщение.

```

1 from steganocryptopy.steganography import Steganography
2
3 Steganography.generate_key("")
4 secret = Steganography.encrypt("key.key", "C:/Users/VAG/Desktop/img/lisi4ka.png"
5     , "secrettext.txt")
6 secret.save("C:/Users/VAG/Desktop/img/lisi4ka_secret.png")
7 result = Steganography.decrypt("key.key", "C:/Users/VAG/Desktop/img
8     /lisi4ka_secret.png")
9 print(result)

```

Рисунок 1 - Реализация метода стенографии с наименьшим значащим битом

Этот же метод может быть применен к другим цифровым носителям, таким как аудио и видео, где данные скрыты в частях файла, что приводит к наименьшим изменениям в звуковом или визуальном выводе.

Различают 5 видов стеганографии: стеганография изображений, текстовая, видео, аудио и сетевая стеганографии.

Стеганография изображений.

Включает в себя сокрытие информации в файлах изображений. Интенсивность пикселей является ключом к сокрытию данных в стеганографии изображений. В цифровой стеганографии изображения часто используются для сокрытия информации, поскольку в цифровом представлении

изображения имеется большое количество элементов, и существуют различные способы сокрытия информации внутри изображения.

Текстовая стеганография.

Текстовая стеганография предполагает сокрытие информации внутри текстовых файлов. В этом методе скрытые данные кодируются буквой каждого слова. Метод также включает в себя изменение слов в тексте, использование контекстно-свободных грамматик для создания читаемых текстов или создание случайных последовательностей символов.

Видео стеганография.

Именно здесь данные скрываются в форматах цифрового видео. Видео стеганография позволяет скрывать большие объемы данных в движущемся потоке изображений и звуков. Дискретное косинусное преобразование (DCT) обычно используется для вставки значений, которые можно использовать для сокрытия данных в каждом изображении в видео, что невозможно обнаружить невооруженным глазом.

Аудио стеганография.

Аудио стеганография — это сокрытие данных в звуке. Она включает в себя секретные сообщения, встраиваемые в звуковой сигнал, который изменяет двоичную последовательность соответствующего звукового файла. Сокрытие секретных сообщений в цифровом звуке — более сложный процесс по сравнению с другими.

Сетевая (протокольная) стеганография.

Сетевая стеганография, иногда называемая стеганографией протокола, представляет собой метод встраивания информации в протоколы управления сетью, используемые при передаче данных, такие как TCP (Transmission Control Protocol), UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol) и т. д. Типичные методы сетевой стеганографии включают изменение свойств одного из сетевых протоколов.

К примерам стеганографии можно отнести:

- Написание невидимыми чернилами.
- Встраивание текста в изображение (водяные знаки).
- Обратную запись сообщения в аудиофайле.
- Сокрытие изображения в видео, которое можно увидеть, только если видео воспроизводится с определенной частотой кадров.
- Встраивание секретного сообщения в зеленый, синий или красный каналы изображения RGB.

Стеганографию можно использовать как в конструктивных, так и в деструктивных целях. Например, для создания авторами изображений невидимых водяных знаков. Последние не искажают изображение и при этом позволяют отслеживать, использовалось ли оно без разрешения.

Однако, с другой стороны, хакеры используют стеганографию для повреждения файлов, данных или сокрытия вредоносных программ в невинных документах. Например, злоумышленники могут использовать скрипты BASH и PowerShell для запуска автоматических атак, встраивая скрипты в документы Word или Excel. И когда пользователь открывает один из этих документов, он активирует скрытый сценарий, тем самым давая волю вредоносным программам. Этот процесс является предпочтительным методом доставки программ-вымогателей.

Практика обнаружения стеганографии называется стегоанализом. Существуют различные инструменты, способные обнаруживать наличие скрытых данных, например StegExpose и StegAlyze. Аналитики могут использовать другие инструменты общего анализа, такие как программы просмотра шестнадцатеричных файлов, для обнаружения аномалий в файлах.

Однако стоит отметить, что поиск файлов, которые были изменены с помощью стеганографии, является сложной задачей потому, что практически невозможно узнать, где начать искать скрытые данные среди миллионов изображений, загружаемых в социальные сети каждый день.

Список использованных источников:

1. kaspersky / What is steganography? Definition and explanation [Электронный ресурс] – kaspersky – 2023. – Режим доступа : <https://www.kaspersky.com/resource-center/definitions/what-is-steganography> / Дата доступа: 10.04.2023.
2. simplilearn / What is Steganography? Types, Techniques, Examples & Applications [Электронный ресурс] – simplilearn – 2023. – Режим доступа : <https://www.simplilearn.com/what-is-steganography-article> / Дата доступа: 10.04.2023.

UDC 159.99

STEGANOGRAPHIC METHODS OF INFORMATION PROTECTION

Vilkina. K. A., Klebeko E. Y., Noskovich P. N.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Stroinikova E. D. – Senior Lecturer in Department of Informatics

Annotation. In this article, methods for protecting information using steganography were considered, and a variant of implementing the steganography method with the least significant bit was proposed.

Keywords: steganography, steganographic methods, information hiding

ФУНКЦИИ АКТИВАЦИИ

Демещенко М.В. студент группы 253503, Марковец Р.С. студент группы 253501, Сугако Т.А. студентка группы 253503, Владимцев В. Д., ассистент каф. Информатики

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Владимцев В.Д. ассистент кафедры информатики

Аннотация. Данная статья посвящена функциям активации - важной составляющей нейронных сетей. В статье рассматриваются основные принципы работы функций активации, их роль в моделировании нелинейных зависимостей между входными и выходными данными, а также области их применения. Также в статье рассматриваются различные виды функций активации, такие как сигмоидная, гиперболический тангенс, ReLU и другие. Описываются их особенности и преимущества, а также рекомендации по выбору функции активации для конкретной задачи.

Ключевые слова. Нейронные сети, функция активации, сигмоидная функция активации, ReLU, ELU

Функция активации является фундаментальной составляющей нейронных сетей, и понимание принципов работы, а также областей применения, способствует созданию более эффективных сетей. Функция активации - это функция в нейроне, которая преобразует входные данные в выходные для их последующей передачи на следующий слой [1]. Входными данными является линейная сумма всех весов (числовая характеристика связи между двумя нейронами) и значений с предыдущего слоя и добавление смещения (bias).

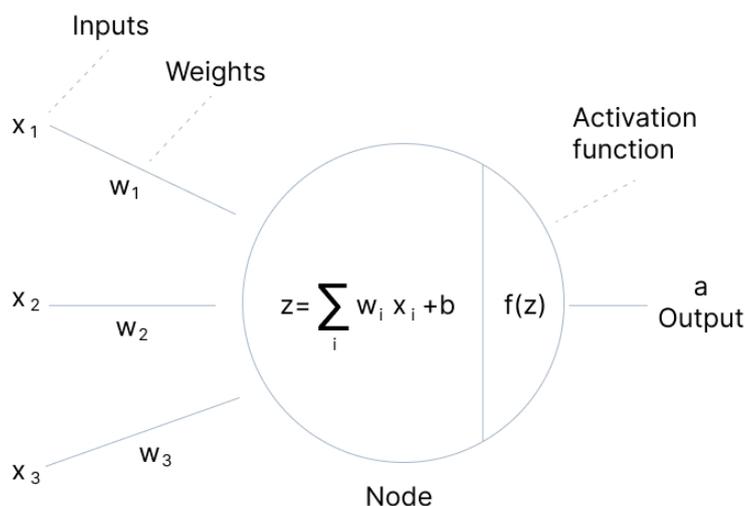


Рисунок 1 - Схема работы функции активации

Главная цель функции активации - внедрение нелинейности в модель нейронной сети. Наличие нелинейности позволяет нейронным сетям разрабатывать сложные представления и функции на основе входных данных, что было бы невозможно при использовании простых линейных функций, ведь композиция линейных функций есть линейная функция.

Одной из самых часто используемых функций активации является сигмоида.

Функция принимает значение от 0 до 1, задается формулой

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

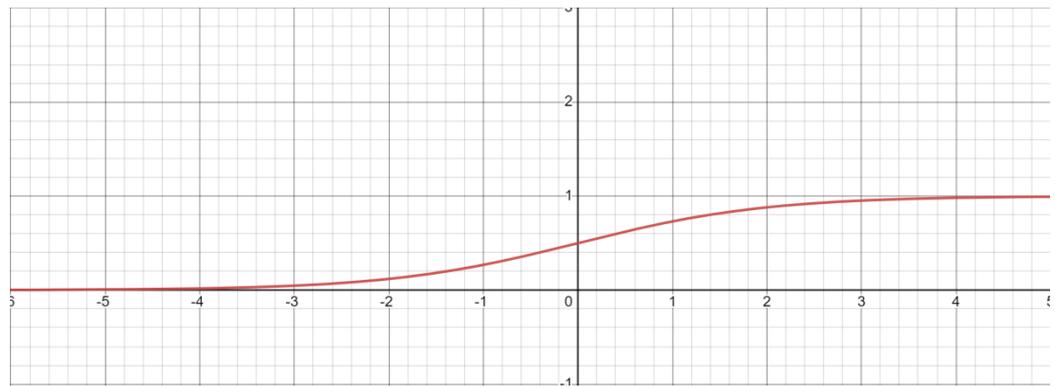


Рисунок 2 - График функции сигмоиды

Сигмоида полезна в задачах бинарной классификации, так как всегда отображает входные данные на значение между 0 и 1, что можно интерпретировать как вероятность принадлежности к одному из двух классов. Еще одним преимуществом сигмоидной функции является ее плавный градиент, что позволяет легко вычислять градиенты при обратном распространении и обновлять веса нейронной сети с помощью градиентного спуска.

Основной недостаток сигмоиды заключается в его производной.

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

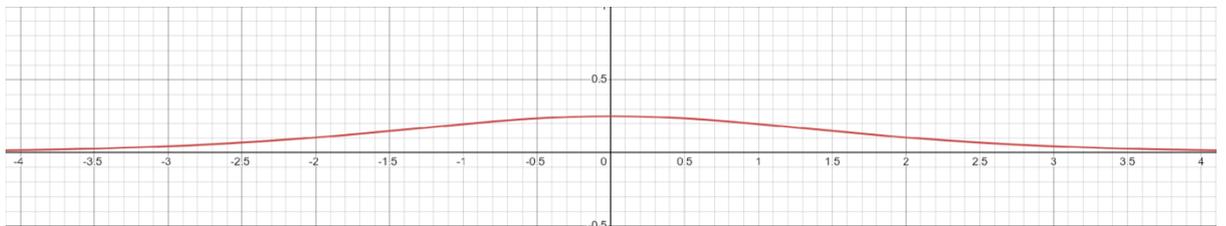


Рисунок 3 - График производной функции сигмоиды

Сигмоида имеет ограниченный диапазон значимых значений градиента, который находится между -3 и 3. За пределами этого диапазона функция становится более плоской, что приводит к очень маленьким градиентам. Данное свойство приводит к проблеме исчезающего градиента (The Vanishing Gradient Problem), когда сеть с трудом обучается, поскольку градиенты приближаются к нулю. Кроме того, значения функции не симметричны вокруг нуля, что означает, что выход всех нейронов будет иметь одинаковый знак. Эта особенность делает обучение нейронной сети более сложным и нестабильным.

Гиперболический тангенс (tanh) похож на сигмоидную функцию, но его диапазон значений находится между -1 и 1, что делает его более симметричным вокруг нуля.

Это позволяет использовать его в более широком диапазоне задач, таких как классификация с несколькими классами.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

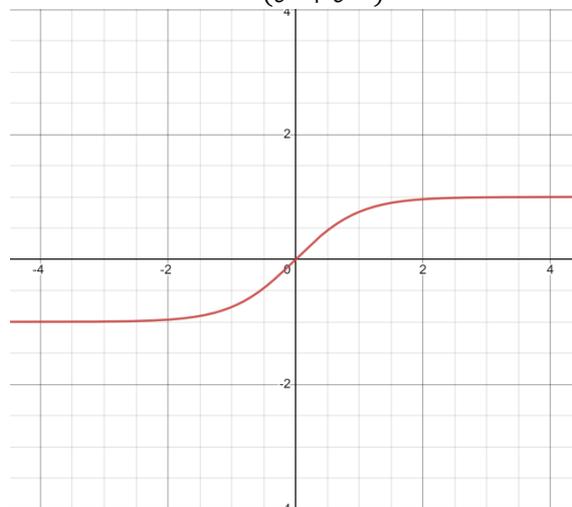


Рисунок 4 - График функции гиперболического тангенса

Как и сигмоида, градиент гиперболического тангенса также может столкнуться с проблемой исчезающего градиента за пределами диапазона значимых значений.

Производная тангенса имеет вид:

$$\tanh'(x) = 1 - \tanh^2(x)$$

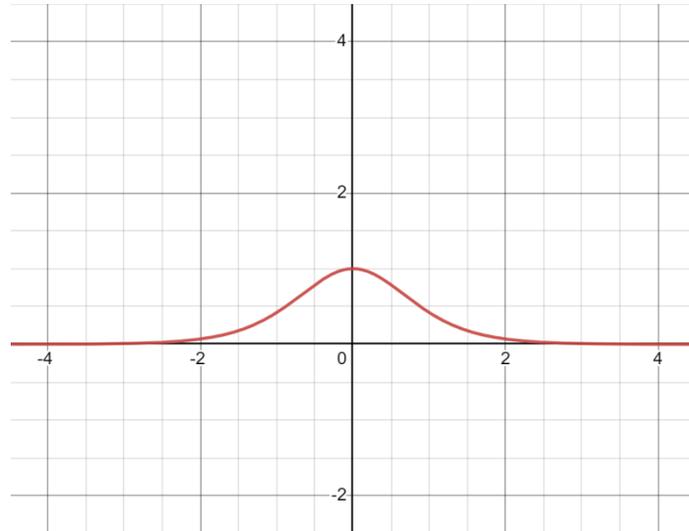


Рисунок 5 - График производной гиперболического тангенса

Решение проблемы исчезающего градиента является использование ReLU (Rectified Linear Unit).

$$\text{ReLU}(x) = \max(0, x)$$

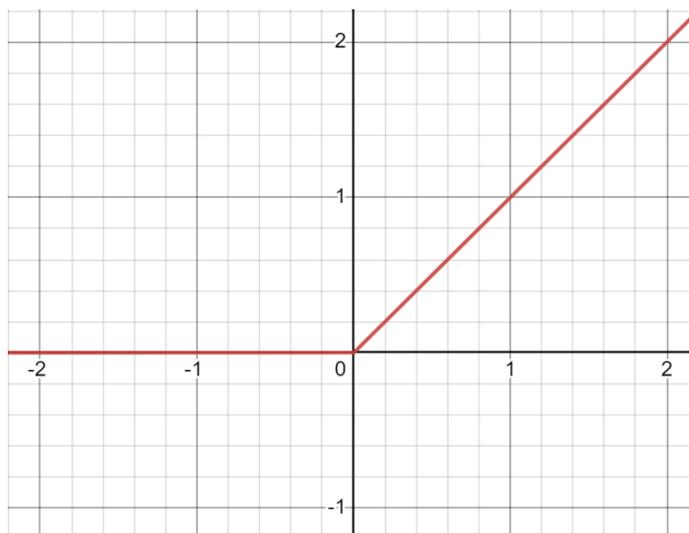


Рисунок 6 - График ReLU

Помимо решения проблемы с градиентом, ReLU также превосходит в простоте и скорости вычислений, что делает ее полезной для глубоких нейронных сетей с большим количеством параметров.

$$\text{ReLU}'(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

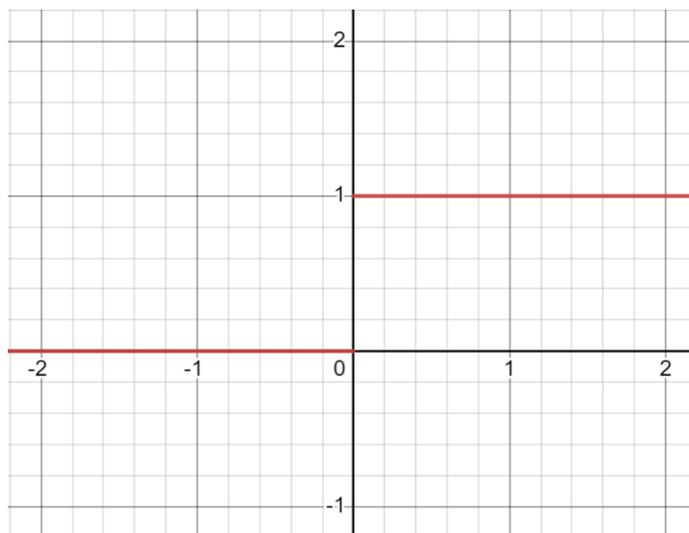


Рисунок 7 - График производной ReLU

Однако ReLU также имеет некоторые недостатки. Один из основных недостатков ReLU заключается в том, что он страдает от проблемы "умирающего ReLU" (dying ReLU [2]), когда большое количество нейронов в сети может стать неактивным и больше никогда не активироваться во время обучения. Это происходит, когда вход в нейрон отрицательный и градиент становится нулевым, в результате чего веса нейрона больше никогда не обновляются. Еще одним недостатком ReLU является то, что он не является плавной функцией, что делает его непригодным для некоторых алгоритмов оптимизации, которые полагаются на плавность, таких как метод сопряженного градиента. Наконец, ReLU не симметрична вокруг нуля, что может привести к тому, что выход всех нейронов будет иметь одинаковый знак, что приведет к замедлению сходимости и нестабильности обучения.

Проблема "умирающего ReLU" решается одной из вариаций данной функции – Leaky ReLU.

$$\text{LeakyReLU}(x) = \max(0.01x, x)$$

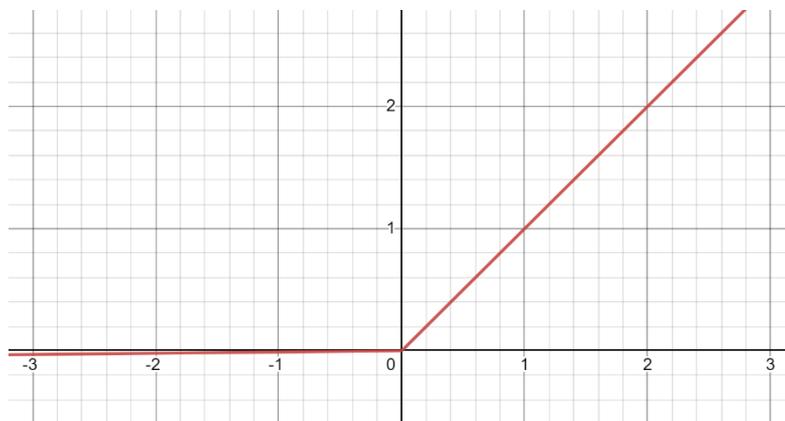


Рисунок 8 - График Leaky ReLU

Leaky ReLU добавляет небольшой наклон к отрицательной области функции, позволяя градиентам по-прежнему проходить через нее и обновлять веса нейронов. Еще одним преимуществом Leaky ReLU является то, что это гладкая функция, что делает ее подходящей для алгоритмов оптимизации, которые полагаются на гладкость, таких как метод сопряженного градиента. Более того, Leaky ReLU также не страдает от проблемы насыщения в положительной области, которая может привести к тому, что градиенты становятся очень маленькими и замедляют обучение.

$$\text{Leaky ReLU}'(x) = \begin{cases} 1, & x \geq 0 \\ 0.01, & x < 0 \end{cases}$$

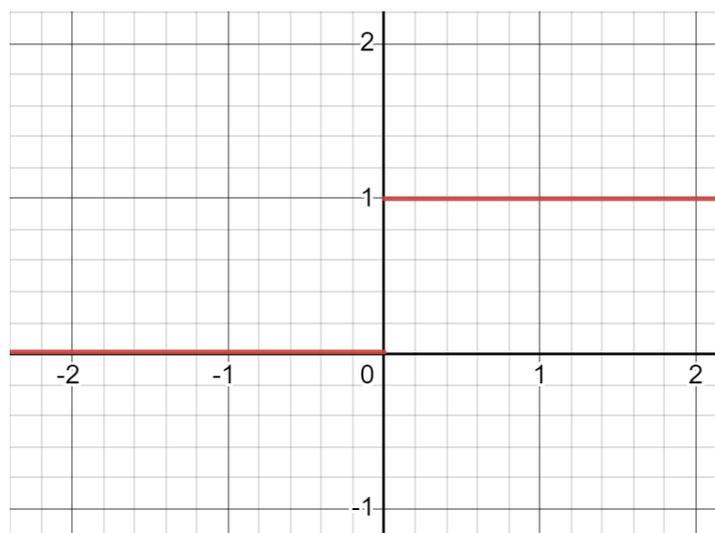


Рисунок 9 - Производная Leaky ReLU

К ограничениям Leaky ReLU относятся непоследовательные предсказания для отрицательных входных значений и небольшой градиент в отрицательной области, который может замедлить обучение параметров.

Еще одна вариация ReLU, направленная на решение проблемы мертвых нейронов - Параметрический ReLU. Эта функция предоставляет наклон отрицательной части функции в качестве аргумента a . Путем обратного распространения происходит обучение наиболее подходящему значению a .

$$\text{ParametricReLU}(x) = \max(ax, x)$$

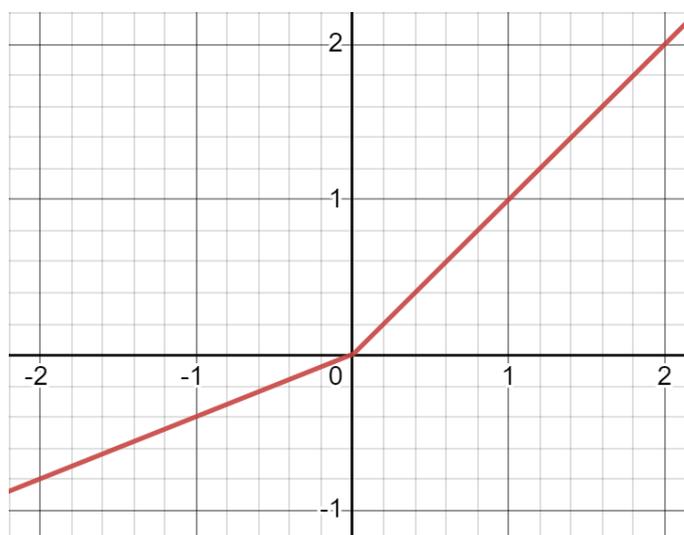


Рисунок 10 - График параметрической ReLU при $a = 0.4$

Где a - параметр наклона для отрицательных значений.

Параметрическая функция ReLU используется, когда функция Leaky ReLU не справляется с проблемой “мертвых” нейронов, и соответствующая информация не передается в следующий слой. Основное ограничение этой функции в том, что она может работать по-разному для разных задач в зависимости от значения параметра наклона a .

Также существует еще одна альтернатива ReLU - экспоненциальное ReLU, которое также решает проблему мертвых нейронов.

$$ELU(x) = \begin{cases} x, & x \geq 0 \\ a(e^x - 1), & x < 0 \end{cases}$$

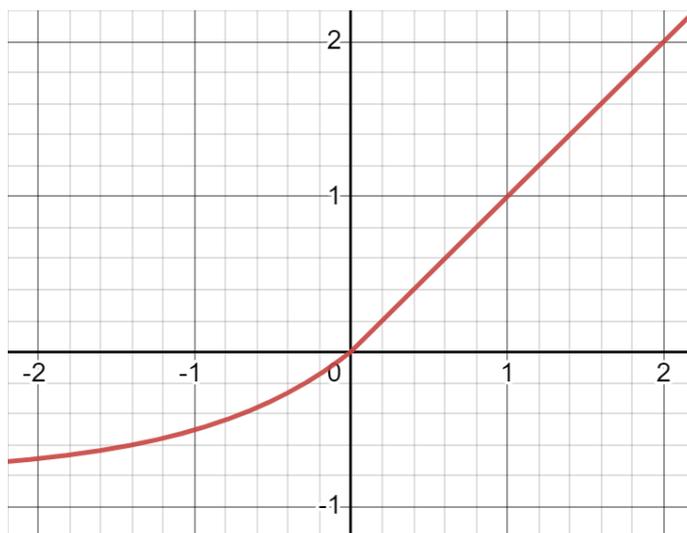


Рисунок 11 - График функции ELU при a = 0.8

Однако у функции активации ELU есть некоторые ограничения. Во-первых, она увеличивает время вычислений из-за экспоненциальной операции, включенной в функцию, что может быть неудобным при обучении больших моделей или работе с большими данными. Во-вторых, ELU имеет гиперпараметр α , который необходимо задавать вручную, так как невозможно обучить. Наконец, ELU не полностью решает проблему взрывающегося градиента (Exploding Gradient [3]), которая все еще может возникать в очень глубоких сетях, но в некоторой степени смягчает ее.

$$ELU'(x) = \begin{cases} 1, & x \geq 0 \\ ELU(x) + a, & x < 0 \end{cases}$$

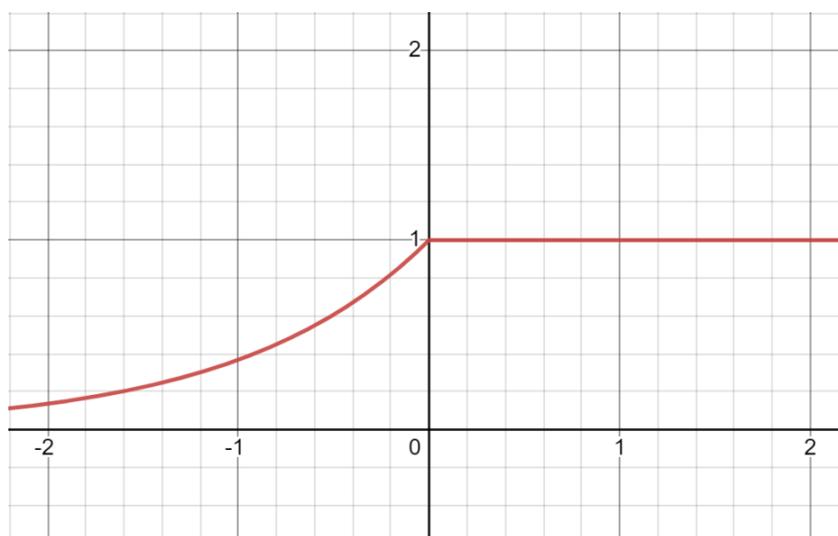


Рисунок 12 - График производной ELU при a = 1

После проведения анализа основных функций активации мы пришли к выводу о том, что некоторые функции могут быть предложены для использования в нейронных сетях. Мы выбрали данные функции с целью решить известные проблемы существующих функций активации, такие как сложные вычисления и исчезающий градиент у сигмоиды и тангенса, а также проблема "мертвых" нейронов у ReLU и его модификаций.

В сравнении с сигмоидной функцией, предложенные нами функции будут считаться намного более эффективными, так как для их вычисления необходимо выполнить лишь несколько простых арифметических операций, таких как сложение, умножение, деление и вычитание, в то время как для вычисления сигмоиды необходимо найти приближение экспоненты в некоторой степени, что требует использования ряда Маклорена и увеличивает время вычислений.

Кроме того, предложенные нами функции обладают непрерывными производными в каждой точке своей области определения, что делает возможным применение некоторых методов обучения, которые недоступны при использовании ReLU-подобных функций.

Строго говоря, предложенные нами функции не имеют параметров и гиперпараметров, что упрощает разработку нейронных сетей с использованием данных функций активации.

Первая предложенная функция активации задается формулой:

$$f(x) = \begin{cases} \frac{x}{1-x}, & x < 0 \\ -\frac{2x^2}{25} + x, & 0 \leq x \leq 5 \\ \frac{x}{5} + 2, & x > 5 \end{cases}$$

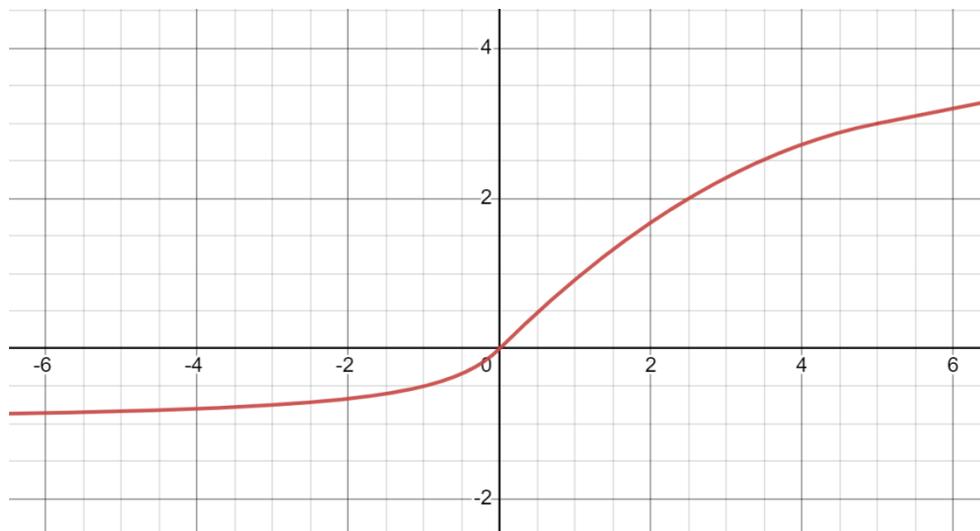


Рисунок 13 - График функции f(x)

$$f'(x) = \begin{cases} \frac{1}{(1-x)^2}, & x < 0 \\ -\frac{4x}{25} + 1, & 0 \leq x \leq 5 \\ \frac{1}{5}, & x > 5 \end{cases}$$

Вторая предложенная функция активации задается следующей формулой:

$$g(x) = \begin{cases} \frac{x}{1-x}, & x < 0 \\ \frac{x^2}{10} + x, & 0 \leq x \leq 5 \\ 2x - 2.5, & x > 5 \end{cases}$$

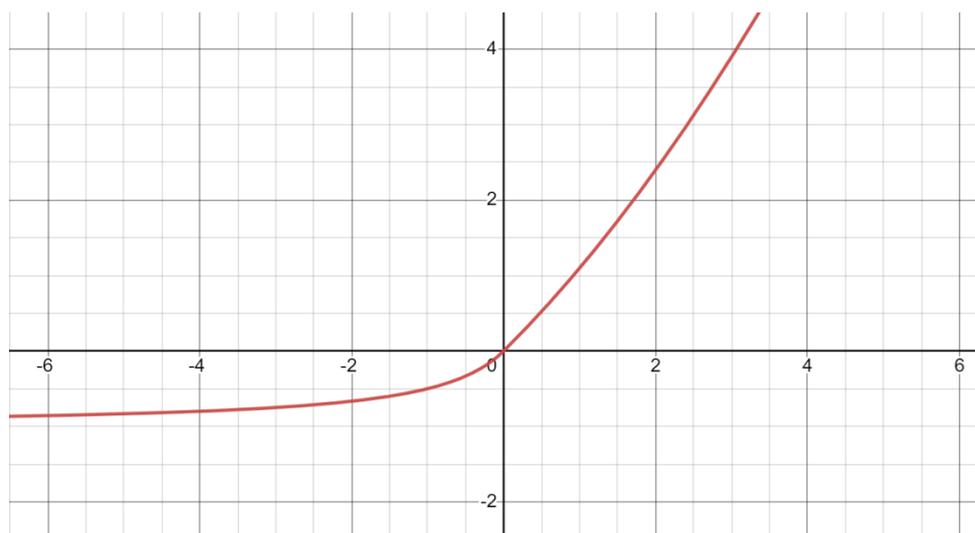


Рисунок 14 - График функции g(x)

$$g'(x) = \begin{cases} \frac{1}{(1-x)^2}, & x < 0 \\ \frac{x}{5} + 1, & 0 \leq x \leq 5 \\ 2, & x > 5 \end{cases}$$

Выбор функции активации является важным этапом при создании нейронной сети, так как она определяет, какой тип нелинейности будет использоваться для преобразования входных сигналов и генерации выходных сигналов. Это может существенно влиять на способность модели к обучению, скорость сходимости и качество результатов.

При выборе функции активации необходимо учитывать как ее математические свойства, так и специфику задачи, для которой создается нейронная сеть. Например, для задач классификации изображений может быть эффективно использовать ReLU, в то время как для задачи регрессии может быть полезно применить гиперболический тангенс.

Кроме того, необходимо учитывать возможные проблемы, связанные с выбранной функцией активации, такие как исчезающий градиент или проблема "мертвых" нейронов. В таких случаях можно рассмотреть альтернативные функции активации, которые не имеют этих проблем.

В целом, выбор функции активации должен основываться на балансе между ее вычислительной эффективностью, математическими свойствами и спецификой задачи. При необходимости можно экспериментировать с различными функциями активации и выбрать ту, которая дает лучшие результаты в конкретной задаче. Общая схема выбора может быть сведена к таблице на рис. 15

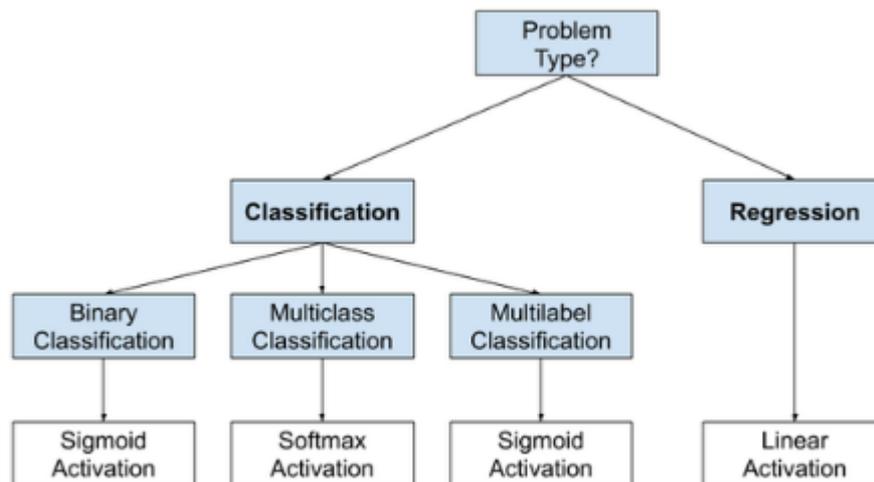


Рисунок 15 - Схема выбора функции активации

В ходе нашей работы по изучению функций активаций мы смогли выдвинуть несколько гипотез о том, какие ранее неиспользованные функции активации могут быть полезны для различных моделей нейронных сетей.

Список использованных источников:

1. Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. *Towards Data Sci*, 6(12), pp.310-316.
2. Lu, L., Shin, Y., Su, Y. and Karniadakis, G.E., 2019. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*.
3. Philipp, G., Song, D. and Carbonell, J.G., 2017. The exploding gradient problem demystified-definition, prevalence, impact, origin, tradeoffs, and solutions. *arXiv preprint arXiv:1712.05577*.
4. Agostinelli, F., Hoffman, M., Sadowski, P. and Baldi, P., 2014. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*.

UDC

ACTIVATION FUNCTIONS

Sugako T.A., Demeschenko M.V., Markovets R.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V.D. – Assistant of the Department of Informatics

Annotation. This paper focuses on activation functions, an important component of neural networks. The paper discusses the basic principles of activation functions, their role in modelling non-linear dependencies between input and output data, and their application areas. Various types of activation functions, such as sigmoid, hyperbolic tangent, ReLU and others are also considered. Their features and advantages are described, as well as recommendations for choosing an activation function for a particular task.

Keywords. Neural networks, activation function, sigmoid activation function, ReLU. ELU.

АРХИТЕКТУРА ИНТЕРАКТИВНОГО ПРИЛОЖЕНИЯ ДЛЯ ОБУЧЕНИЯ БЕЛОРУССКОМУ ЯЗЫКУ В УСЛОВИЯХ ИНФОРМАТИЗАЦИИ ОБЩЕСТВА

Макатров А.А. студент группы 053505, Ермолаев Н.В. студент группы 053505, Владимирцев В. Д., ассистент каф. Информатики

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Владимирцев В.Д. – ассистент кафедры информатики

Аннотация. Информатизация общества – глобальный процесс активного формирования и широкомасштабного использования информационных ресурсов. Одним из приоритетных направлений информатизации общества является процесс информатизации образования, который предполагает широкое использование информационных технологий обучения. Использование мультимедийных технологий в обучении белорусскому языку является эффективным инструментом, который может значительно повысить мотивацию учащихся и улучшить качество образования.

Ключевые слова. информатизация общества, информатизация образования, белорусский язык, информационные технологии, мультимедийные технологии, доступность образовательных ресурсов.

Введение

XXI век называют «эпохой информации», «веком глобального информационного общества». В таком обществе образование, знания, информация и коммуникация составляют основу развития и благополучия человеческой личности. Информационные технологии являются неотъемлемой частью образовательного процесса, позволяют значительно повысить эффективность обучения и улучшить качество знаний, особенно в сфере языкового образования.

Белорусский язык является одним из государственных языков Республики Беларусь.

Различные мультимедийные технологии, такие как интерактивные учебные материалы, видео уроки и онлайн-курсы, могут значительно улучшить процесс обучения и помочь учащимся лучше усваивать материал, понимать и использовать язык в повседневной жизни.

Информатизация, мультимедийные технологии

Глобальная информатизация общества является одной из доминирующих тенденций развития цивилизации в XXI в. Благодаря стремительному увеличению возможностей средств информатики, телекоммуникационных систем и новых информационных технологий формируется информационная среда обитания и жизнедеятельности людей, складывается информационное общество.

Информатизация – объективный политический, организационный, экономический, социально-культурный и научно-технический процесс создания оптимальных условий для реализации информационных отношений, формирования и применения информационных ресурсов с целью развития информационного общества [1].

Простыми словами информатизация – это внедрение информационных технологий во все сферы жизни человека. В то же время – это процесс, благодаря которому информация применяется более эффективно и способствует развитию отдельных отраслей и общества в целом.

В Беларуси построение информационного общества осуществляется на основе «Стратегии развития информатизации в Республике Беларусь на 2016-2022 годы», утвержденной Президиумом Совета Министров Республики Беларусь от 03.11.2015 № 26, а также в рамках государственных программ информатизации.

Одним из приоритетных направлений информатизации общества является процесс информатизации образования. Информатизация образования – это совершенствование обучения с помощью информационных технологий. Информатизация образования ставит своей целью изменить формы и методы преподавания так, чтобы обучающийся легче усваивал знания и не терял интереса к обучению [2].

Информатизация образования развивается по следующим направлениям:

совершенствование системы управления образованием в рамках единого образовательного информационного пространства;

развитие национальной системы образовательных информационных ресурсов;

развитие системы электронных услуг в сфере образования;

совершенствование программно-технической инфраструктуры системы образования.

При этом следует уделить должное внимание таким задачам как:

создание ключевых информационных ресурсов (единый регистр учреждений образования, единая база данных обучаемых и др.) и развитие на их основе информационного обеспечения управленческой деятельности и системы электронных услуг;

широкомасштабное использование электронных коммуникаций для информационного взаимодействия педагогов, обучаемых, родителей, внедрение проекта «Электронная школа», обеспечивающего комплексное решение управленческих задач и совершенствование образовательной деятельности в учреждениях общего среднего образования;

разработка совместимых открытых электронных образовательных ресурсов для всех уровней образования, по всем направлениям и специальностям подготовки, обеспечение их актуальности и доступности в национальной образовательной информационной среде;

обеспечение постоянного доступа педагогов к различным коллекциям электронных учебных объектов с целью самостоятельного конструирования занятий;

развитие перспективных направлений дистанционного обучения, внедрение элементов мобильного образования на базе «облачных» технологий [3].

В области информатизации образования особое внимание уделяется доступности обучающимся информационных ресурсов сети Интернет и совершенствованию национальной системы электронных образовательных ресурсов. Сегодня все без исключения общеобразовательные школы имеют возможность широкополосного доступа к сети Интернет.

Фундаментом национальной культуры, национального самосознания, объединяющим признаком народа является язык как средство коммуникации.

В современном мире белорусский язык сохраняет свою важность для белорусского народа и культуры. Он применяется в официальных документах, средствах массовой информации, образовании и науке, литературе и искусстве, а также в повседневной жизни.

Использование новейших информационных технологий в обучении белорусскому языку повышает мотивацию и познавательную активность обучающихся, позволяет применить личностно-ориентированную интерактивную образовательную технологию, даёт возможность избежать субъективной оценки и повышает эффективность обучения и качество образования.

В настоящее время к одному из наиболее динамично развивающихся и перспективных направлений информационных технологий относятся мультимедийные технологии. Внедрение мультимедийных технологий в образовательный процесс является одним из ключевых моментов информатизации образования [4].

Использование мультимедийных технологий в современном образовании способствует:

- совершенствованию системы управления обучением;
- повышению мотивации обучения;
- улучшению качества обучения;
- повышению уровня подготовки обучающихся в области современных информационных технологий;

– возможности использования компьютера, мобильного телефона и планшета для обучения.

Современные методы обучения с привлечением мультимедийных технологий помогают решить следующие дидактические задачи:

- усвоить базовые знания по предмету;
- систематизировать усвоенные знания;
- сформировать навыки самоконтроля;
- развивать творческие способности обучающихся;
- сформировать мотивацию к обучению;
- предоставить учебно-методическую помощь в самостоятельной работе над учебным материалом.

Мультимедиа – это совокупность программно-аппаратных средств, которые реализуют обработку информации в звуковом и зрительном видах. Графика, анимация, фото, видео, звук, текст в интерактивном режиме работы создают информационную среду, в которой пользователь обретает качественно новые возможности.

Мультимедиа технологии – это совокупность современных средств аудио-, теле-, визуальных и виртуальных коммуникаций, используемых в процессе организации, планирования и управления образовательной деятельностью [5].

Мультимедиа технологии обеспечивают такое представление информации, при котором человек воспринимает её сразу несколькими органами чувств параллельно, а не последовательно, как это делается при обычном обучении.

Обучающие мультимедийные программы используются для фронтального, группового и индивидуального обучения в аудитории, а также для самостоятельной работы дома. Они предлагают для пользователя очень много вариантов индивидуальной настройки: учащийся, осваивая учебный материал, сам устанавливает скорость изучения, объем материала и степень его трудности.

Позитивные факторы, которые говорят в пользу такого способа получения знаний, следующие:

- лучшее и более глубокое понимание изучаемого материала;
- мотивация обучаемого на контакт с новой областью знаний;
- экономия времени из-за значительного сокращения времени обучения.

Полученные знания остаются в памяти на более долгий срок и позднее легче восстанавливаются для применения на практике после краткого повторения.

В современном образовании можно выделить три основных направления использования мультимедийных технологий:

- использование уже готовых программных продуктов;
- применение программных продуктов, которые создаются непосредственно преподавателями (или преподавателями совместно с обучающимися);
- использование ресурсов сети Интернет.

Анализ существующих мультимедийных ресурсов

Quizlet – это онлайн-платформа для обучения, которая позволяет создавать, делиться и использовать наборы карточек, тесты и игры для изучения различных предметов и языков.

На данный момент есть возможность использовать Quizlet для изучения белорусского языка. Вы можете создать свои собственные наборы карточек с белорусскими словами и их переводами или же найти уже созданные наборы на белорусском языке. Также после изучения карточек, есть возможность пройти тест по пройденному материалу (рисунок 1).

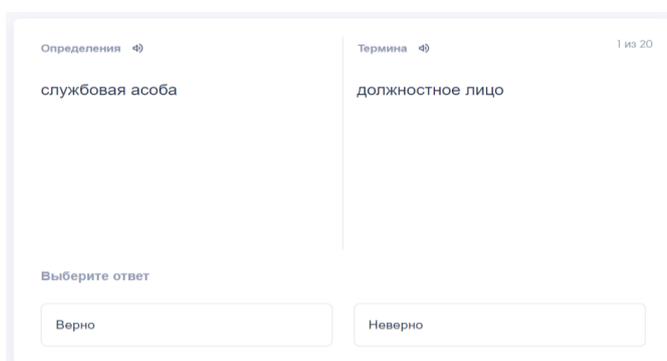


Рисунок 1 – Возможность прохождения тестов на онлайн-платформе Quizlet

Кроме того, Quizlet имеет функцию произношения слов, которая поможет вам правильно произносить белорусские слова (рисунок 2).

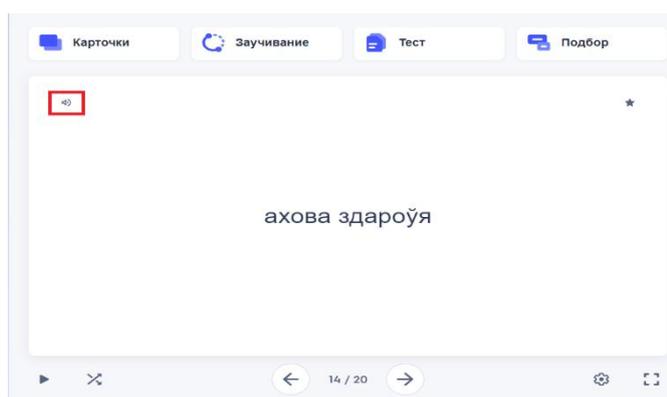


Рисунок 2 – Функция произношения слов на онлайн-платформе Quizlet

Выделим следующие плюсы платформы Quizlet для изучения белорусского языка:

1. Большой выбор наборов карточек на белорусском языке. Вы можете найти уже созданные наборы или создать свои собственные.
2. Функция произношения слов, которая помогает правильно произносить слова и фразы на белорусском языке.
3. Возможность использовать Quizlet на различных устройствах, таких как компьютеры, планшеты и смартфоны.
4. Простой и интуитивно понятный интерфейс, что делает использование платформы доступным для пользователей всех возрастов и уровней подготовки.

Также у платформы Quizlet можно выделить следующие минусы:

1. Некоторые пользователи могут ощутить нехватку возможностей для индивидуального обучения и получения обратной связи от преподавателей или наставников.
2. Некоторые функции могут быть доступны только для платных подписчиков;
3. Возможны ошибки в уже созданных наборах карточек, поэтому всегда следует проверять правильность перевода.

Kahoot! – это платформа для создания интерактивных образовательных игр, которая позволяет создавать, играть и делиться играми в режиме реального времени. Как правило, игры Kahoot! используются для обучения, проведения тестов и оценки знаний учеников.

Для создания данных игр на сайте присутствует удобный конструктор (рисунок 3).

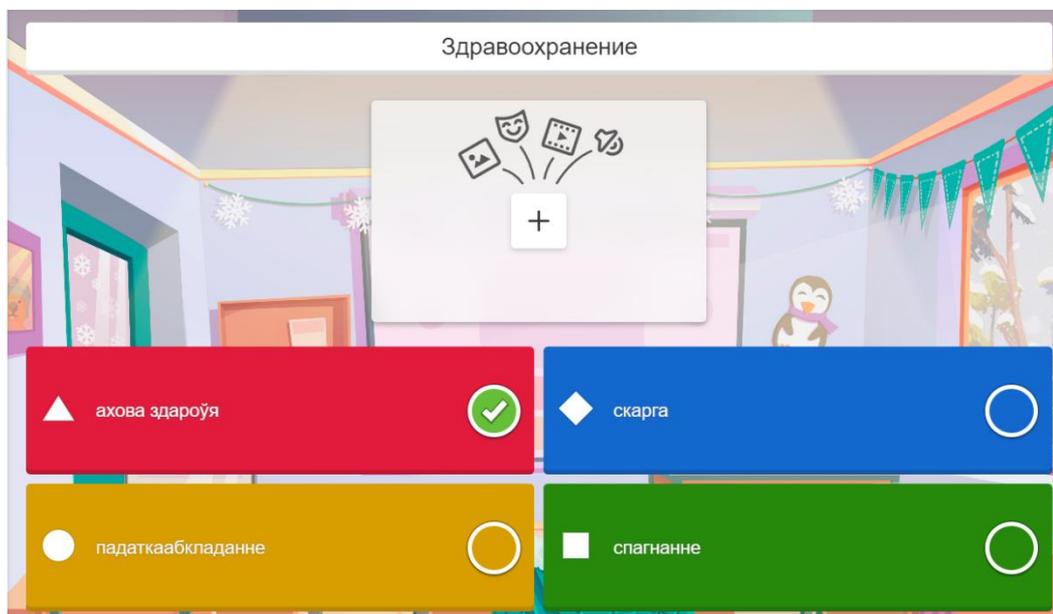


Рисунок 3 – Конструктор интерактивных игр Kahoot!

Мы можем использовать данный сервис для обучения различным аспектам белорусского языка, таким как правописание, грамматика, лексика и произношение. Кроме того, Kahoot! может использоваться для проведения тестов и проверки знаний в различных темах, связанных с белорусским языком.

Выделим следующие плюсы платформы Kahoot! для изучения белорусского языка:

1. Интерактивность: Kahoot позволяет создавать интерактивные игры на белорусском языке, которые могут помочь в усвоении грамматики, лексики, и произношения.
2. Развлекательный характер: Игры на Kahoot могут сделать процесс изучения белорусского языка более занимательным и увлекательным, особенно для детей.
3. Простота использования: Kahoot очень прост в использовании, что позволяет быстро создавать и настраивать игры для изучения белорусского языка.
4. Онлайн-доступность: Kahoot доступен в Интернете и может использоваться в любом месте и в любое время, если у вас есть подключение к Интернету.

Также у платформы Kahoot! можно выделить следующие минусы:

1. Ограниченность формата: Kahoot – это формат игры, который может быть неудобен для изучения некоторых тем и трудных понятий, например, для изучения сложной грамматики.
2. Не всегда подходит для всех возрастов: Kahoot может быть не подходящим для всех возрастов или уровней образования, так как для изучения белорусского языка могут потребоваться более серьезные и глубокие методы.
3. Зависимость от технологии: использование Kahoot требует подключения к Интернету и использования технологии, что может быть непрактично в некоторых условиях.
4. Неэффективное использование: если игры на Kahoot не созданы правильно или не соответствуют потребностям учеников, они могут не принести ожидаемых результатов.

WOOCCLAP – это интерактивная онлайн-платформа для создания и проведения опросов, викторин, голосований и других видов интерактивных мероприятий в режиме реального времени.

На WOOCCLAP вы можете создавать и настраивать опросы, викторины, голосования и другие интерактивные мероприятия, добавлять вопросы, ответы, картинки, видео, графики, таблицы и другие элементы для улучшения интерактивности и наглядности (рисунок 4). Вы можете настроить время на каждый вопрос или активность, а также выбрать из различных форматов ответов – множественный выбор, открытый ответ, рейтинги, голосование пальцами вверх/вниз и т.д.



Рисунок 4 – Пример теста на WOOCCLAP

Выделим следующие плюсы платформы WOOCCLAP для изучения белорусского языка:

1. WOOCCLAP предлагает различные типы интерактивных мероприятий, такие как тесты, опросы, голосования, открытые вопросы и другие, что может помочь студентам в изучении белорусского языка и получении обратной связи.
2. Платформа позволяет настраивать время на каждую активность и задавать вопросы в различных форматах, включая вставку картинок и видео, что может сделать уроки более интересными и понятными для студентов.
3. WOOCCLAP обеспечивает возможность интеграции с другими приложениями, такими как Google Slides, PowerPoint и другие, что может быть полезно для организаторов мероприятий, которые используют эти приложения в своей работе.
4. Платформа имеет мультязычный интерфейс, что может быть полезно для студентов, которые изучают белорусский язык как второй язык.

Также у платформы WOOCCLAP можно выделить следующие минусы:

1. WOOCCLAP является платной платформой, поэтому ее использование может быть недоступно для некоторых студентов и учителей из-за финансовых ограничений.
2. Некоторые функции могут быть сложными для новичков и могут потребовать некоторого времени и обучения для их освоения.

LearningApps.org – это бесплатная платформа для создания и обмена интерактивными учебными материалами. Она была создана в Германии в 2012 году и в настоящее время продолжает развиваться. На LearningApps.org педагоги и студенты могут создавать множество видов учебных материалов, таких как кроссворды, карточки, тесты, головоломки и многие другие. Эти материалы могут быть созданы с использованием различных типов контента, включая текст, изображения, аудио и видео.

На платформе есть множество учебных материалов на различных языках, включая белорусский. Вы можете использовать уже созданные учебные материалы или создавать свои собственные, чтобы подстроить их под свои нужды и интересы.

Т.к. данная платформа старше всех перечисленных выше, то и количество различных заданий здесь заметно больше если сравнивать его с остальными (рисунок 5).



Рисунок 5 – Готовые задания на LearningApps.org

Кроме возможности практики на уже готовых заданиях, есть возможность создавать новые (рисунок 6).

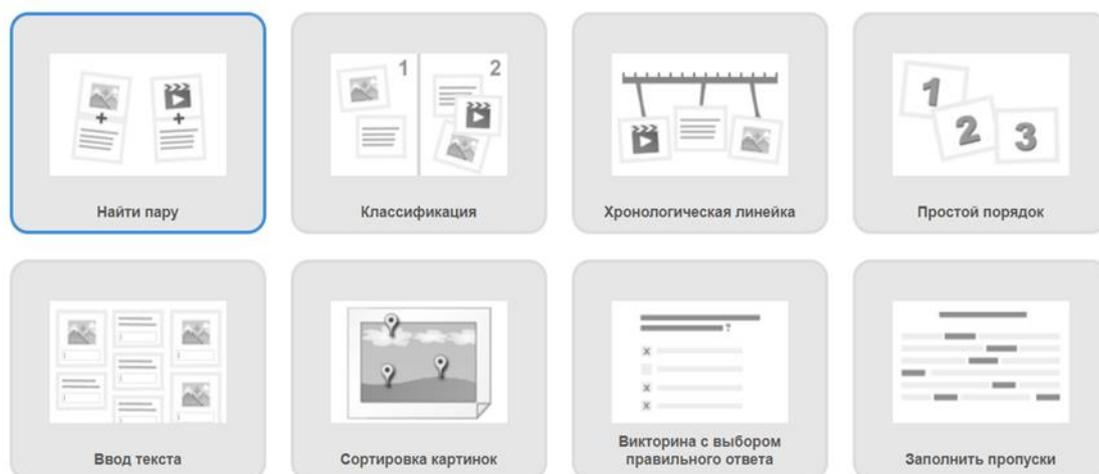


Рисунок 6 – Возможность создания новых заданий

Выделим следующие плюсы платформы LearningApps.org для изучения белорусского языка:

1. **Качество контента:** LearningApps.org имеет огромный выбор учебных материалов на различных языках, в том числе и на белорусском языке. Контент может быть создан не только преподавателями, но и студентами, что делает его более разнообразным и интересным.
2. **Разнообразие типов учебных материалов:** LearningApps.org предлагает множество типов учебных материалов, что позволяет создавать интерактивные задания, которые помогут студентам лучше усвоить материал на белорусском языке.
3. **Настраиваемость:** LearningApps.org позволяет настраивать учебные материалы и создавать их с нуля, что может быть полезно для преподавателей, которые хотят создавать свои собственные задания на белорусском языке.
4. **Бесплатность:** LearningApps.org является бесплатной платформой, что делает ее доступной для всех, кто хочет изучать белорусский язык.
5. **Языковые настройки:** LearningApps.org поддерживает множество языков, в том числе и белорусский, что делает ее удобной для использования студентами и преподавателями на территории Беларуси.

Однако можно отметить следующие минусы LearningApps.org для изучения белорусского языка:

1. **Ограниченность функционала:** LearningApps.org не является полноценной платформой для обучения языку и не может заменить учебник или преподавателя.
2. **Недостаточная теоретическая база:** LearningApps.org не предоставляет достаточного объема теоретических материалов на белорусском языке, необходимых для полноценного изучения языка.
3. **Необходимость наличия Интернет-соединения:** для работы с LearningApps.org необходимо наличие Интернет-соединения, что может быть неудобно для студентов, которые предпочитают работать без доступа в Интернет.
4. **Ограничения при создании заданий:** LearningApps.org имеет определенные ограничения при создании заданий, что может быть неудобно для преподавателей, которые хотят создать более сложные и многокомпонентные задания на белорусском языке.

Smoge – это онлайн-сервис для создания профессиональных электронных брошюр (рисунок 7), которые могут использоваться в различных образовательных целях, в том числе и для обучения белорусскому языку.

Преимущества Smoge для обучения белорусскому языку:

1. **Интерактивность:** на сайте можно создавать брошюры с различными мультимедийными элементами, такими как видео, изображения, аудио, гиперссылки и т.д., что делает процесс обучения более интересным и привлекательным для учеников.
2. **Гибкость:** Smoge позволяет создавать брошюры на различные темы и в различных форматах, что позволяет учителям адаптировать материалы под потребности конкретного класса или группы учеников.

3. Простота использования: сайт имеет простой и интуитивно понятный интерфейс, что делает его доступным для использования даже теми, кто не имеет опыта в создании электронных материалов.

Недостатки Smore для обучения белорусскому языку:

1. Ограничения на бесплатную версию: некоторые функции, такие как возможность загружать свои изображения и видео, доступны только в платной версии Smore.
2. Ограниченные возможности для создания интерактивных упражнений: хотя Smore позволяет добавлять мультимедийные элементы, но возможности для создания интерактивных упражнений и тестов ограничены.

Для использования Smore в обучении белорусскому языку можно создавать брошюры на различные темы, например, грамматика, лексика, чтение, письмо и т.д. В брошюрах можно включать видеоуроки, звуковые файлы для отработки произношения, интерактивные упражнения и тесты. Это может быть особенно полезно для дистанционного обучения, когда ученики не могут лично посещать уроки.

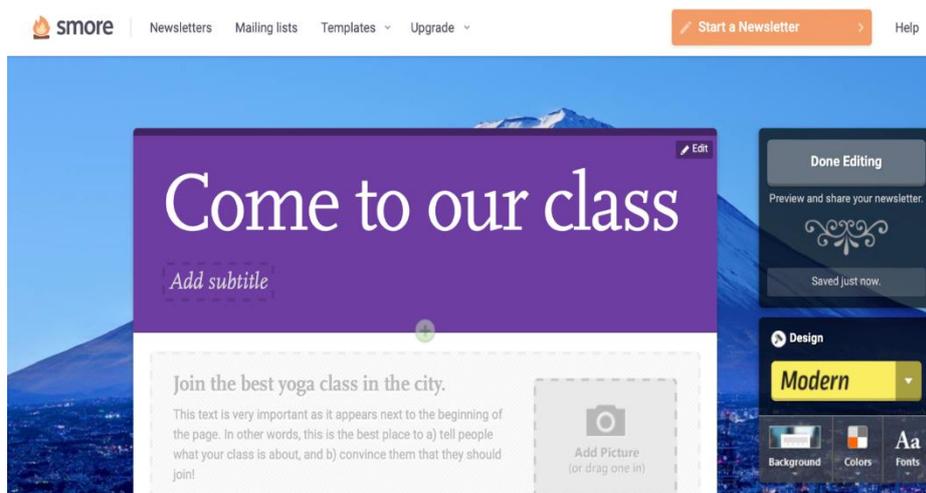


Рисунок 7 – Создание электронных брошюр в онлайн-сервисе Smore

В целом, Smore может быть полезным инструментом для обучения белорусскому языку, который позволяет учителям создавать интерактивные и привлекательные материалы для учеников. Однако, для эффективного использования этого инструмента, учителям необходимо учитывать его возможности и ограничения и комбинировать его с другими методами и инструментами обучения.

Jigsaw Planet – это сайт, предоставляющий возможность создавать пазлы онлайн. Он может использоваться для обучения белорусскому языку путем создания пазлов с изображениями, подписанными белорусскими словами или фразами.

Основные возможности сайта Jigsaw Planet:

1. Создание пазлов из собственных изображений: пользователи могут загружать свои изображения и создавать пазлы различной сложности.
2. Выбор количества и формы элементов пазла: пользователи могут выбирать количество и форму элементов пазла, что позволяет создавать пазлы разной сложности.
3. Возможность делиться пазлами с другими пользователями: пользователи могут делиться своими пазлами с другими пользователями или использовать пазлы, созданные другими.
4. Режим игры: пользователи могут играть в пазлы на сайте, что позволяет им развивать свои навыки решения задач и улучшать свои лексические знания в белорусском языке.

Преимущества использования Jigsaw Planet для обучения белорусскому языку:

1. Интерактивный подход: использование пазлов позволяет обучающимся взаимодействовать с языком в интерактивной форме.
2. Развитие лексических навыков: создание пазлов с изображениями, подписанными белорусскими словами или фразами, помогает обучающимся развивать свой словарный запас.
3. Приятный способ обучения: решение пазлов является приятным и увлекательным способом обучения, что может способствовать более эффективному запоминанию материала.

Недостатки использования Jigsaw Planet для обучения белорусскому языку:

1. Ограниченный функционал: сайт предназначен только для создания и решения пазлов, что может быть недостаточным для полноценного обучения языку.

2. Ограниченный контент: количество пазлов на сайте может быть ограничено, что может привести к ограниченному выбору материалов для обучения.
3. Отсутствие возможности проверки правильности выполнения заданий: сайт не предоставляет возможности проверить правильность выполнения заданий, что может быть проблемой для обучения языку.

Skarnik.by – электронный русско-белорусский словарь. За основу взят академический словарь, который был выпущен в 1953 году (под редакцией Я. Коласа, К. Крапивы и П. Глебки) и затем несколько раз переиздавался с исправлениями и дополнениями. Skarnik доработан с учётом современной практики.

Основные возможности и преимущества онлайн-словаря на сайте skarnik.by включают:

1. Большой объем информации: словарь содержит более 200 тысяч слов и словосочетаний, что обеспечивает покрытие большинства ситуаций в повседневной жизни.
2. Многоязычность: словарь на сайте skarnik.by позволяет искать слова и словосочетания как на белорусском, так и на русском языке, что обеспечивает удобство для всех, кто изучает белорусский язык.
3. Дополнительная информация: помимо перевода, словарь на сайте skarnik.by предоставляет дополнительную информацию о слове, такую как часть речи, синонимы, антонимы, примеры использования в предложениях и многое другое.
4. Удобство использования: онлайн-словарь на сайте skarnik.by быстро и легко ищет нужные слова и позволяет быстро получить информацию о них.

Однако, онлайн-словарь на сайте skarnik.by также имеет свои недостатки, среди которых:

1. Ограниченность: словарь может не содержать всех слов, которые нужны в конкретной ситуации, что может быть проблемой для людей, которые хотят узнать перевод более редких слов.
2. Ошибки перевода: как и любой другой словарь, онлайн-словарь на сайте skarnik.by может содержать ошибки перевода, что может привести к неправильному пониманию значения слова.
3. Отсутствие контекста: онлайн-словарь не может учитывать контекст использования слова, что может привести к неправильному переводу.

Архитектура разрабатываемого приложения

В предыдущем разделе были перечислены наиболее популярные ресурсы для самостоятельного обучения языкам, которые используются по всему миру. Однако, неудобство заключается в том, что все они представляют собой различные ресурсы и для того, чтобы качественно обучаться белорусскому языку, необходимо изучить механизмы работы множества сайтов.

В рамках данной научной работы было принято решение, что для комфортного обучения белорусскому языку необходимо разработать приложение, которое будет включать в себя все сильные стороны существующих на данный момент ресурсов.

В результате были выделены следующие сильные стороны каждого ресурса:

1 Quizlet. Однозначно самой сильной стороной этого ресурса являются его интерактивные карточки. Выигрышность в использовании данного вида обучения заключается в том, что создание таких карточек – нересурсоёмкий процесс, на который не уходит много времени, а учить белорусский язык, используя данные карточки, намного удобнее, чем прибегая каждый раз к словарю. Также процесс создания карточек можно автоматизировать. Выше был упомянут ресурс под названием Skarnik.by, который содержит огромную базу переводов слов (более 100000 слов) с русского на белорусский языки и наоборот.

На данном сайте нет раздела “Для разработчиков”, который используется некоторыми веб-приложениями для предоставления разработчикам API, отправляя запрос на который, они могут получить необходимую информацию с сайта, однако остаётся возможность использовать форму ввода слов (рисунок 8):

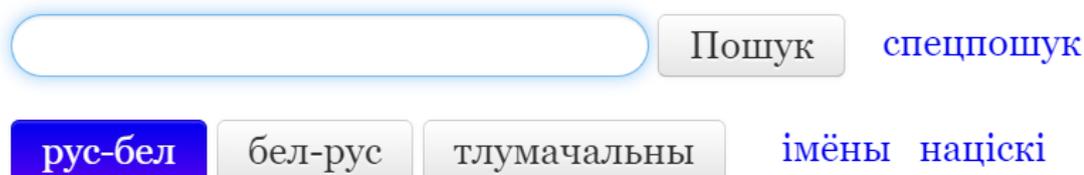


Рисунок 8 – Поле для поиска переводов на сайте Skarnik.by

В качестве URL в HTTP-Get запросе, отправляемом на сервер при поиске очередного слова используется следующий адрес: <https://www.skarnik.by/search?term=WORD&lang=rus>. Следовательно, отправляя запрос по данному адресу, всегда будет получен HTML-документ, который далее разбивается на токены и на выходе получается перевод слова.

Таким образом, можно настроить автоматическую генерацию карточек, путём выбора нескольких слов на русском языке из базы и поиска соответствующих им переводов.

2 Kahoot. Сильной стороной данного ресурса являются, безусловно, интерактивные игры. Особенностью данных игр является простота в создании. Kahoot предлагает удобный конструктор для этого (см. рисунок 3). Необходимо включить этот конструктор и функциональность по созданию игр в целом в итоговое приложение.

3 Wooclap. Сильной стороной данного ресурса являются опросы, викторины и голосования. В режиме реального времени можно построить процесс, наладив связь между обучающимися и преподавателем приближенную к той, которая формируется на очном уроке.

4 Jigsaw Planet. Сильной стороной данного ресурса является возможность создания пазлов на основе пользовательского изображения. Во время создания пазла можно установить количество и форму пазлов, таким образом настраивая его сложность. Решение позволяет развивать навыки решения задач и улучшать лексические знания в белорусском языке, что существенно влияет на эффективность обучения.

5 Skarnik.by. Данный ресурс предлагает функциональность обычного словаря, что для разработчиков итогового приложения является огромным плюсом, т.к. нет необходимости создавать и поддерживать огромную базу переводов. За подачу и обновление информации отвечает данный ресурс, а разработчики приложения пользуются этим.

Таким образом, необходимо включить в итоговое приложение отдельный сервис, который будет выполнять роль словаря, где пользователь в режиме реального времени сможет посмотреть перевод интересующего его слова, примеры его использования и некоторые морфологические особенности, касающиеся данного слова. Также внутри приложения данный ресурс будет использоваться как хранилище переводов.

На рисунке 9 представлена структура приложения на основе перечисленных выше преимуществ различных ресурсов.

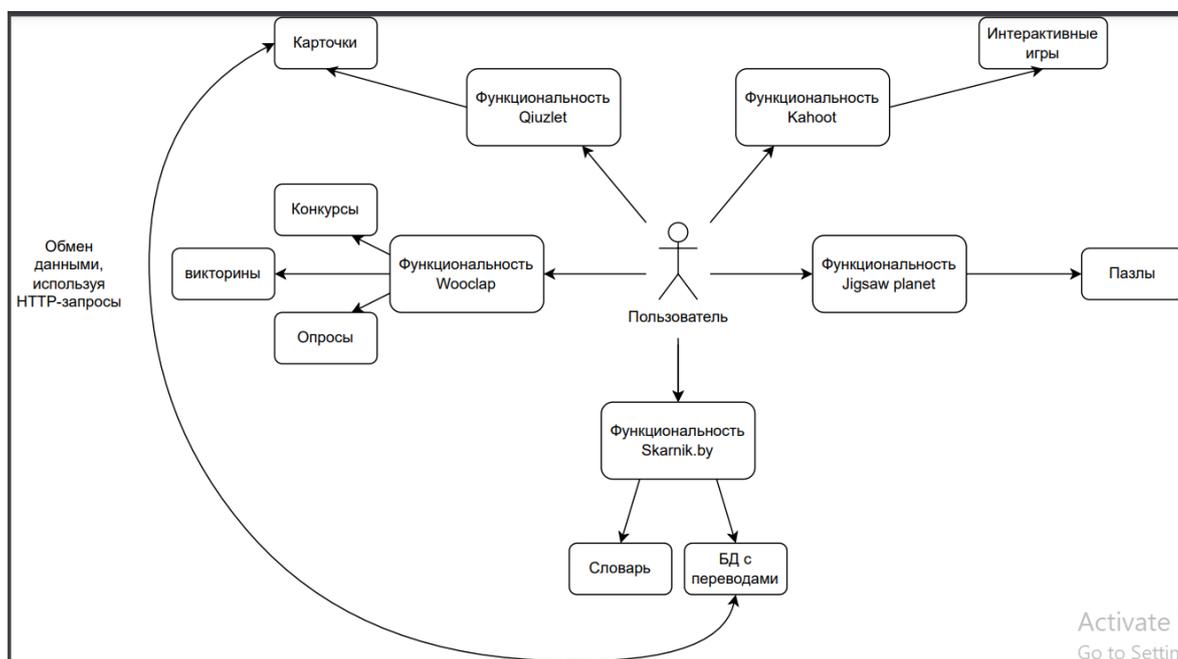


Рисунок 9 – Структура приложения

Предлагается разбить приложение на следующие микросервисы:

1. Микросервис чтения, основанный на лучших практиках Quizlet, который будет предоставлять пользователю возможность изучения новых слов путём заучивания карточек и прохождения тестам по ним.
2. Микросервис словаря, использующий технологии Scarnik.by который будет предоставлять пользователю возможность получения перевода интересующих его слов, а также служить базой с переводами для других микросервисов.

3. Микросервис чтения, основанный на WOOCAP, который будет предоставлять пользователю тексты на белорусском языке и задания на понимание прочитанного в виде викторин, опросов и голосований.
4. Микросервис интерактивного взаимодействия, использующий технологии Kahoot, который будет предоставлять пользователю возможность провести время за игрой. Однако, пользователь продолжит учить белорусский язык, за счёт того, как построены игры.
5. Микросервис сборки пазлов, основанный на Jigsaw Planet, который будет предоставлять пользователю задания на сборку пазлов с изображениями на белорусском языке.
6. Микросервис обратной связи, который будет предоставлять пользователю возможность обратиться за помощью и получить ответы на свои вопросы.

Приложение будет иметь модульную структуру, где каждый из микросервисов будет отвечать за конкретный аспект изучения языка и работать он будет на отдельном сервере. Взаимодействующее с другими микросервисами организуется посредством API-интерфейсов, что обеспечит гибкость и масштабируемость приложения.

Для удобства использования пользователем можно реализовать следующий функционал:

1. Регистрация и авторизация. Пользователь сможет зарегистрироваться в приложении и войти в свой аккаунт. Для регистрации можно использовать адрес электронной почты или социальные сети.
2. Личный кабинет. Пользователь сможет редактировать свой профиль, просматривать свою статистику, результаты выполненных заданий и сохраненные слова в словаре.
3. Календарь. Система будет автоматически создавать персональный календарь пользователя с рекомендуемыми заданиями на каждый день.
4. Оценка знаний. После выполнения задания система будет автоматически оценивать знания пользователя и рекомендовать задания на более высоком уровне сложности.
5. Уведомления. Система будет оповещать пользователя о новых заданиях, изменениях в расписании и ответах на свои вопросы.
6. Форум. Пользователь сможет общаться с другими пользователями, задавать вопросы и обсуждать темы, связанные с изучением белорусского языка.

Такое приложение для обучения белорусскому языку будет инновационным, удобным и эффективным инструментом для тех, кто хочет улучшить свои языковые навыки. Оно будет предоставлять широкий спектр возможностей для изучения языка на разных уровнях, начиная от базовых знаний и заканчивая продвинутыми уровнями.

Для обеспечения мотивации и интереса пользователей приложение может иметь систему наград и достижений за выполнение заданий и достижение определенных целей. Также можно предусмотреть возможность создания пользовательских заданий и курсов для изучения белорусского языка, что позволит пользователям делиться своими знаниями и опытом с другими пользователями.

В целом, приложение для обучения белорусскому языку с использованием лучших практик из WOOCAP, Kahoot!, Quizlet, Jigsaw Planet, Skarnik.by будет представлять собой уникальный продукт, который поможет пользователям эффективно и интересно изучать язык, улучшать свои языковые навыки и достигать своих целей.

Заключение

Использование мультимедийных технологий в обучении белорусскому языку является важным и перспективным направлением в условиях информатизации общества. Мультимедийные материалы позволяют сделать учебный процесс более интерактивным и эффективным, что может привести к улучшению результатов обучения и повышению мотивации учеников. Кроме того, использование мультимедийных технологий способствует более глубокому пониманию языковых явлений и развитию коммуникативных навыков учащихся. Однако, необходимо учитывать, что мультимедийные материалы не должны заменять традиционный учебный материал, а использоваться как дополнение к нему. Кроме того, необходимо учитывать индивидуальные особенности каждого ученика и предоставлять им возможность выбирать подходящие для них мультимедийные материалы. В целом, использование мультимедийных технологий в обучении белорусскому языку является важным инструментом, который позволяет сделать процесс обучения более интересным и эффективным.

Список использованных источников:

1. *Основы информатики и информационные технологии: учебно-метод. комплекс для студ. ист. фак. в 2 ч. Ч. 1 / Е. Э. Попова, Н. Н. Садова, Ю. Ю. Тагирова. – Минск: БГУ, 2008. – 160 с.*
2. *Захарова, И. Г. Информационные технологии в образовании: [учебное пособие для высших педагогических учебных заведений] / И. Г. Захарова. – М.: Академия, 2003. – 188 с.*
3. *Шпилевская, В. В. Информатизация в Республике Беларусь и становление электронного правительства: пособие / В.В. Шпилевская. – Минск: Минский государственный ПТК полиграфии, 2019. – 56 с.*
4. *Андерсен, Бент. Б. Мультимедиа в образовании: специализированный учеб. курс: [пер. с англ] / Бент. Б. Андерсен, Катя Ван Ден Бринк. – 2 - е изд.; испр. и доп. – М.: Дрофа, 2007. – 221 с.*

UDC

THE USE OF MULTIMEDIA TECHNOLOGIES IN TEACHING THE BELARUSIAN LANGUAGE IN THE CONTEXT OF SOCIETY'S INFORMATIZATION

Makatrau A. A., Yermalayeu N. V..

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V.D. – Assistant of the Department of Informatics

Annotation. The informatization of society is a global process of active formation and wide-scale use of information resources. One of the priority directions of the informatization of society is the process of informatization of education, which involves the widespread use of information technologies in teaching. The use of multimedia technologies in teaching the Belarusian language is an effective tool that can significantly increase students' motivation and improve the quality of education.

Keywords. society informatization, education informatization, Belarusian language, information technology, multimedia technology, accessibility of educational resources.

УДК

ПРИМЕНЕНИЕ КОМПЬЮТЕРНОГО ЗРЕНИЯ И MLOPS В СФЕРЕ МЕДИЦИНСКОГО ОБСЛУЖИВАНИЯ

Шкарубский М.А. студент группы 153503, Владимирцев В. Д., ассистент каф. Информатики

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Владимирцев В. Д. – ассистент кафедры информатики

Аннотация. В данной статье описывается применение современных технологий в сфере машинного обучения, больших данных и MLOps в медицинской сфере. Значительное внимание уделяется компьютерному зрению и возможным сценариям его применения. В работе также рассматривается использование облачных сервисов для разворачивания моделей и налаживания процессов, с целью создания надежной, расширяемой системы, способной не только давать точные прогнозы, но и надежно работать практически при любых обстоятельствах.

Ключевые слова. Машинное обучение (Machine Learning), MLOps, Компьютерное Зрение (Computer Vision), Большие Данные (Big Data), Google Cloud Platform, API, DICOM, медицина.

На сегодняшний день, современные технологии машинного обучения (ML) находят свое применение в самых разных сферах деятельности человека. От рекомендательных систем в социальных сетях и музыкальных сервисах, до анализа речи и эмоций человека – область применения алгоритмов ML действительно широка. Эта статья выделяет одно из главных направлений ML – компьютерное зрение, и рассказывает о том, как оно может улучшить медицинскую сферу уже сегодня.

Компьютерное зрение - это важная технология машинного обучения, которая находит широкое применение в медицине. Оно используется для анализа медицинских изображений, таких как РЭМ и МРТ. При этом компьютерное зрение может обнаруживать даже те признаки заболевания, которые не могут быть замечены человеческим глазом, что позволяет существенно улучшить точность диагностики и эффективность лечения. Более того, компьютерное зрение может быть использовано для анализа других типов медицинских изображений, что может помочь в различных областях, например, в планировании хирургических вмешательств, в оценке состояния костной ткани, или нахождении раковых опухолей в человеческом мозгу.

Однако, ни одна система, применяющаяся в промышленном масштабе, не способна существовать без четко налаженных процессов. Такая система не будет расширяемой, легко поддерживаемой и актуализируемой, а ведь от этого напрямую зависит ее эффективность. Часто бывает, что распределение входных данных, как и целевых переменных со временем может изменяться, а модель (или совокупность моделей), должна отвечать этим изменениям, и подстраиваясь под них, и в то же время не терять свою обобщающую способность. Другим также вполне реальным сценарием является переход на новые технологии: к примеру, команда ML-инженеров решила, что разумным будет попробовать использовать ResNet вместо Inception [1] в рамках текущей задачи. Для совершения такого перехода нужно иметь гибкую и легко тестируемую систему, чтобы проверить эффективность новой нейронной сети, провести сравнительный анализ и без больших задержек заменить старую модель новой. Здесь и приходит на помощь MLOps.

MLOps - это методология, которая объединяет несколько этапов в жизненном цикле моделей машинного обучения. Она включает в себя процессы разработки, тестирования, внедрения и управления моделями машинного обучения, а также управления данными. Важным аспектом MLOps является автоматизация процессов. Автоматизация позволяет сократить время, затрачиваемое на каждый этап, приводит к меньшей вероятности ошибок и упрощает масштабирование. MLOps также включает в себя мониторинг и управление моделями после их внедрения. Это важный аспект, который позволяет быстро обнаруживать проблемы и улучшать модели в реальном времени. Кроме вышеупомянутых плюсов, MLOps может быть использована для управления версиями моделей и данных, что облегчает совместную работу нескольких команд и ускоряет процесс разработки. Пример CI/CD и автоматизированного процесса разработки ML продукта показан на рисунке 1.

На данный момент существует несколько основных платформ, позволяющих относительно легко и быстро разворачивать целые приложения с использованием MLOps. Одним из примеров таких платформ является Google Cloud Platform (GCP) [2], которая имеет ряд необходимых платформ и API для взаимодействия с ними. Эти сервисы позволяют создавать, обучать и запускать

модели машинного обучения, а также обрабатывать большие объемы данных. Например, Vertex AI – это платформа, которая позволяет разрабатывать, обучать и запускать модели машинного обучения. Она обеспечивает автоматическое масштабирование и оптимизацию моделей, а также позволяет управлять их жизненным циклом. Еще одним важным сервисом является Cloud Storage [3] – сервис, который позволяет хранить и управлять данными. Он обеспечивает высокий уровень безопасности и доступности данных, а также позволяет легко интегрировать данные с другими сервисами Google Cloud Platform.

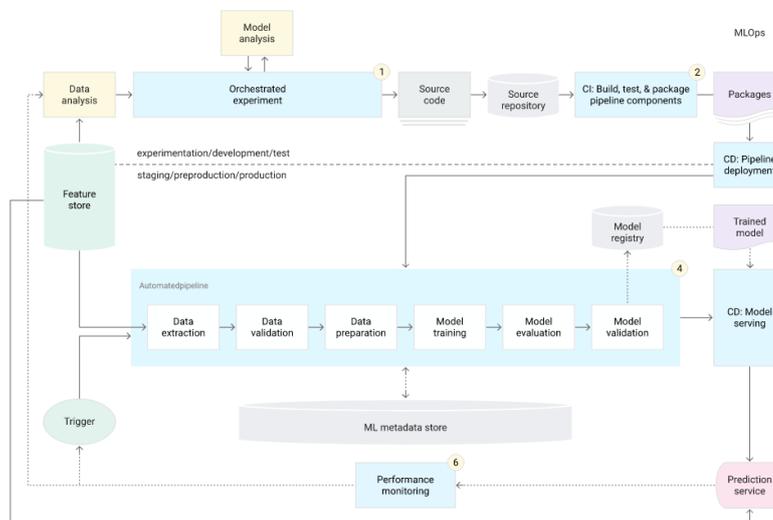


Рисунок 1 – автоматизация всего ML пайплайна с использованием continuous integration и continuous delivery.

Все вышеописанные субплатформы GCP, как и само машинное обучение (а именно нейронные сети, являющие собой основу компьютерного зрения), нашли свое применение в рамках курсовой работы по предмету «Объектно-ориентированное программирование». Суть работы состоит в разработке сервиса, помогающего врачам делать прогнозы о наличии у пациента раковой опухоли в мозгу на основе 3D скана его мозга. Обучение модели (как, впрочем, и весь проект) базируется на датасете, предоставленном некоммерческой организацией «Радиологическое общество Северной Америки» на ресурсе Kaggle в рамках одного из соревнований [4]. Датасет состоит из файлов в формате .dcm, каждый из которых соответствует срезу одного из четырех форматов: T1wCE, T1w, T2w, FLAIR, которые позволяют исследовать мозг на наличие в нем глиобластомы – самой частой и агрессивной формы опухоли мозга, которая составляет до 52 % первичных опухолей мозга и до 20 % всех внутримозговых опухолей [5]. Структура данных соревнования определяется тремя частями: Обучение, Валидация (Публичная) и Тестирование (Приватная). «Обучающая» и «Валидационная» части предоставляются участникам, в то время как часть для тестирования остается скрытой на протяжении всего соревнования и после его окончания. Эти 3 части организованы следующим образом: каждый независимый случай (в сущность – анонимизированный пациент) имеет отдельную папку, идентифицируемую пятизначным номером. В каждой из этих папок есть четыре подпапки, каждая из которых соответствует каждому из структурных МРТ-сканов (mpMRI).

Задача создания такого сервиса делится на несколько основных частей, каждая из которых также может включать в себя некоторые подзадачи: исследовательский анализ данных (EDA), выбор архитектуры и построение модели, развертывание модели в облачном сервисе.

Главной целью проведения EDA в данном проекте являлась выработка понимания формата DICOM, поиск особенностей сканов мозга различных типов, сравнение сканов больных людей со сканами здоровых и оценка распределения классов (целевой переменной). Для этого в проекте использовалась среда Jupyter Notebook, позволяющая точно манипулировать блоками кода, а не запускать всю программу целиком. Помимо этого, EDA помог обнаружить и исправить существенную проблему – неупорядоченность хранения данных в пределах одного 3D скана исходного датасете. Такая неупорядоченность могла навредить модели на этапе обучения, так как топология данных теряется, что приводит к потере эффективности свертки – признаки, которые она могла выделить на этапе обучения нарушены по умолчанию. Решением проблемы являлся переход от EDA к разработке программного продукта.

Для более надежной работы с данными было принято решение ввести две сущности – ScanSlice и Scan3D, представляющих собой двумерный скан (срез), и трехмерный скан, то есть совокупность срезов, соответственно. Каждый экземпляр класса Scan3D сортировал хранящиеся в

нем срезы, используя в качестве ключа сортировки метаданные среза, а именно поле Instance Number, которое отвечало за позицию (в контексте очередности) среза. Сравнение сортированных и несортированных срезов проводится на рисунке 2. Еще одной довольно серьезной проблемой являлось неодинаковое количество срезов (фактически глубина) каждого трехмерного скана. Довольно популярной практикой для ее решения было выделение центральной части изображения фиксированной глубины, что приводило к тому, что данные теряли свою репрезентативность. В рамках курсовой работы было принято альтернативное решение: было решено дополнять все сканы пустыми срезами до определенного, общего для всех сканов размера, после чего обучать модель уже на новых данных.

После формализации представления данных в коде и дальнейшего исследования было принято решение использовать модель InceptionV3 на первых этапах разработки продукта. Она отличается большим количеством слоев, но относительно небольшим (например, относительно моделей класса VGG) числом параметров, благодаря свёрточным слоям 1×1 (1×1×1 в случае трехмерной свертки), понижающим размерность входов последующих слоев. Хорошей практикой является использование вначале не самой сложной с точки зрения архитектуры и количества слоев модели и попытка подбора таких валидационных стратегий, которые действительно отражают поведение модели при обучении на данных (подтверждением тому является использование этой практики лидерами соревнования).

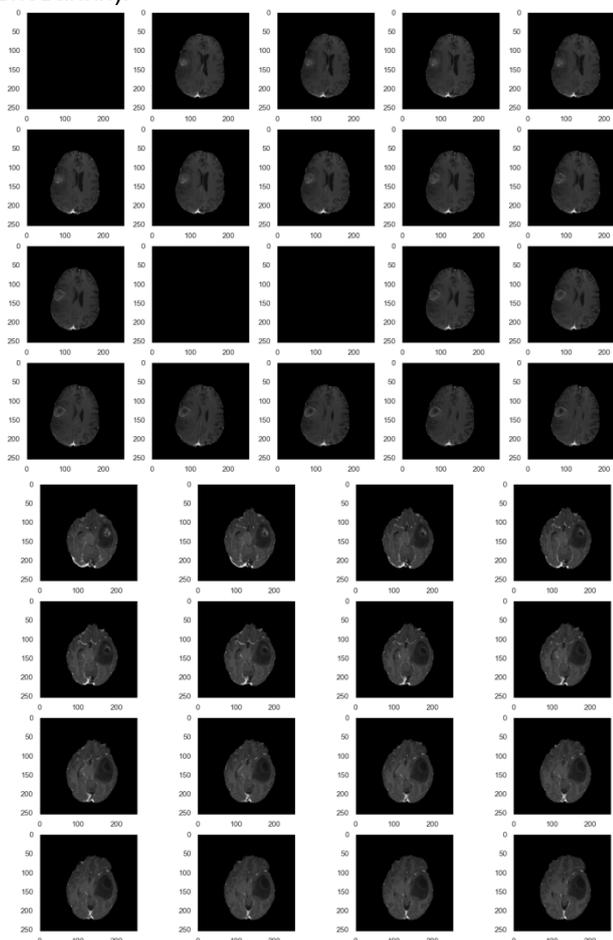


Рисунок 2 – слева находятся несортированные срезы в формате T1wCE пациента №43, справа – сортированные. Можно заметить что слева есть пустые изображения, что говорит о том, что срезы, которые на них изображены, пусты, и взяты либо сверху, либо снизу, но никак не с середины. Справа же имеем последовательную картину, где опухоль постепенно увеличивается.

Для имплементации модели, можно воспользоваться Vertex AI SDK, являющимся более высокоуровневой версией Vertex AI Client library. SDK предоставляет возможность создавать датасеты, манипулировать экспериментами (Experiments) и работами (Jobs). Для хранения тренировочных данных разумно использовать Google Cloud Storage, данные из которого можно напрямую направлять в датасеты Vertex AI, в том числе и мини-батчами.

Чтобы обучить модель на тренировочных данных, можно воспользоваться встроенным сервисом AutoML, который автоматически создаст работы, настроит гиперпараметры и оценит качество работы модели с помощью кросс-валидации. Для более гибкой работы имеет смысл создавать свои работы, используя либо Python Distributions, либо Docker контейнеры. В рамках

работы можно использовать собственные метрики качества, функции ошибки, тренировочные циклы, и даже способы хранения и подачи данных модель (это особенно важно в контексте работы с большими данными, т.к. такие данные труднее обрабатывать, что подразумевает использование более подходящих способов хранения и манипуляции, таких как TFRecord, например), что делает Vertex AI крайне мощной платформой, предоставляющих целый ряд решений для одной задачи. Для более продвинутой работы с Vertex AI следует обратиться к документации GCP [5].

Таким образом, данная работа описывает MLOps и ее важность для промышленной реализации алгоритмов машинного обучения. В документе представлен обзор платформ и API для MLOps, с акцентом на Google Cloud Platform, а также пример сервиса, разработанного для прогнозирования опухолей головного мозга с использованием 3D сканов головного мозга и различных сервисов Google Cloud Platform.

Список использованных источников:

1. *MLOps: Continuous delivery and automation pipelines in machine learning* – [Электронный ресурс] Электронные данные. – Режим доступа: Google Cloud Platform (<https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning>).
2. *Cloud Storage documentation* – [Электронный ресурс] Электронные данные. – Режим доступа: Google Cloud Platform (<https://cloud.google.com/storage/docs>).
4. U.Baid, et al., "The RSNA-ASNR-MICCAI BraTS 2021 Benchmark on Brain Tumor Segmentation and Radiogenomic Classification", arXiv:2107.02314, 2021.
5. Глиобластома – [Электронный ресурс] Электронные данные. – Режим доступа: Wikipedia.org (<https://www.wikipedia.com/ru/Глиобластома>).
6. *Vertex AI documentation* – [Электронный ресурс] Электронные данные. – Режим доступа: Google Cloud Platform (<https://cloud.google.com/vertex-ai/docs>).
7. *Going deeper with convolutions* / Christian Szegedy [et al.] // Cornell University, 2014.

UDC

APPLICATION OF COMPUTER VISION AND MLOPS IN THE FIELD OF MEDICAL CARE

Shkarubski M. A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V.D. – Assistant of the Department of Informatics

Annotation. This article describes the application of modern technologies in the field of machine learning, big data and MLOps in the medical field. Considerable attention is paid to computer vision and possible scenarios of its application. The paper also discusses the use of cloud services for deploying models and establishing processes, in order to create a reliable, extensible system that can not only give accurate forecasts, but also work reliably under almost any circumstances.

Keywords. Machine Learning, Apps, Computer Vision, Big Data, Google Cloud Platform, API, DICOM, Medicine.

УДК

АНАЛИЗ И ПРИМЕНЕНИЕ МЕТОДОВ ТЕПЛОВИЗИОННОЙ ОБРАБОТКИ ВИДЕОСИГНАЛОВ

*Вашкевич Н.В. студент группы 053505, Осадчий О.Э. студент группы 053505,
Владымцев В. Д., ассистент каф. Информатики*

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Владымцев В.Д. – ассистент кафедры информатики

Аннотация. Данная работа имеет целью провести анализ методов тепловизионной обработки видеосигналов и сама обработка данных сигналов на примере конкретных бинарных данных, которые необходимо будет обработать и перевести в формат кадров, а затем и видео.

Ключевые слова: обработка изображений, работа с бинарными данными, гамма-коррекция, тепловизионные камеры, нормализация гистограмм.

Введение.

Тепловизионные камеры являются незаменимым инструментом во многих областях, от промышленности до медицины. Они позволяют получать информацию о температуре объектов и процессов, которые невозможно заметить невооруженным глазом. Однако, данные, получаемые с тепловизионных камер, обычно представлены в бинарном формате, что затрудняет их визуализацию и анализ. В данной статье мы рассмотрим процесс преобразования данных тепловизионной камеры в изображения и создания видео на их основе. Конкретно, мы обсудим применение гамма-коррекции для улучшения контрастности и яркости изображений, а также процесс компиляции нескольких изображений в видео, позволяющий получить полную картину происходящего процесса.

Для увеличения качества тепловизионного изображения, перед преобразованием пикселей интенсивности в пиксели оттенков серого, необходимо произвести нормализацию гистограммы интенсивности[1] каждого кадра исходного видеоряда, чтобы добиться более качественного изображения. Для этого в теории обработки изображений существуют методы нормализации гистограммы интенсивности изображения. Перед перечислением наиболее распространённых методов нормализации необходимо дать определение этому процессу.

Нормализация гистограммы интенсивности изображения - это процесс приведения распределения значений интенсивности пикселей изображения к определенному стандартному виду. Целью нормализации гистограммы является улучшение качества изображения, повышение его контрастности и улучшение его восприятия.

Во время нормализации гистограммы интенсивности, обычно используются различные методы преобразования гистограммы. Некоторые из наиболее распространенных методов включают в себя масштабирование значений интенсивности пикселей таким образом, чтобы они попадали в определенный диапазон, либо же другие методы, которые позволяют повысить контрастность в областях изображения с низкой интенсивностью, и которые позволяют распределить значения интенсивности равномерно по всему диапазону.

Нормализация гистограммы интенсивности широко используется в обработке изображений для повышения качества изображений в медицинской диагностике, а также для улучшения качества фотографий и других изображений. Она может быть использована для коррекции оттенков кожи в фотографиях, улучшения качества документов при сканировании и для анализа изображений в машинном зрении.

Нормализация гистограммы интенсивности является важным этапом в обработке изображений и может быть необходима по нескольким причинам:

1. Улучшение визуализации: Нормализация гистограммы интенсивности может улучшить визуальное восприятие изображения. При ненормализованной гистограмме могут быть области, которые выглядят слишком темными или светлыми, что может приводить к потере деталей и информации на изображении. Нормализация гистограммы интенсивности может сделать изображение более ярким и контрастным, что улучшает восприятие и визуальный анализ.

2. Улучшение качества анализа: Нормализация гистограммы интенсивности может улучшить качество анализа изображения. Например, в задачах распознавания образов или классификации объектов, нормализация гистограммы интенсивности может улучшить точность и надежность алгоритмов.

3. Универсальность: Нормализация гистограммы интенсивности может быть применена к различным типам изображений и задачам, что делает ее универсальным методом обработки изображений.

В целом, нормализация гистограммы интенсивности является необходимой для обработки изображений, так как она может улучшить визуализацию, улучшить качество анализа и обеспечить универсальность методов обработки. Наиболее часто используемыми методами нормализации гистограммы интенсивности изображения являются:

1. Минимаксная нормализация: приводит значения пикселей к диапазону от 0 до 1, используя минимальное и максимальное значения в изображении.

2. Среднеквадратичная нормализация (Z-score нормализация): вычитает среднее значение из всех значений пикселей и делит на стандартное отклонение, чтобы получить новую гистограмму со средним значением 0 и стандартным отклонением 1.

3. Логарифмическая нормализация: берет логарифм от значений пикселей, чтобы сжать динамический диапазон изображения и распределить значения пикселей по гистограмме более равномерно.

4. Квантильная нормализация: приводит значения пикселей к процентному диапазону (например, от 0 до 99 перцентилей), игнорируя выбросы.

5. Эквализация гистограммы: перераспределяет значения пикселей, чтобы получить равномерное распределение гистограммы.

6. Адаптивная эквализация гистограммы: разбивает изображение на множество небольших блоков и выполняет эквализацию гистограммы для каждого блока отдельно.

7. Гистограммное выравнивание: изменяет яркость и контрастность изображения, чтобы максимально использовать весь динамический диапазон и сделать изображение более контрастным.

8. Цветовая коррекция: приводит яркость, контрастность и цвета изображения к определенным стандартным значениям, чтобы убрать цветовые и яркостные искажения.

9. Конвертация изображения в другое цветовое пространство: переводит изображение из одного цветового пространства в другое для улучшения качества изображения и улучшения гистограммы интенсивности.

10. Свертка изображения: используется для сглаживания и уменьшения шума в изображении, что может привести к более равномерному распределению гистограммы интенсивности.

11. Гамма-коррекция: используется для изменения яркости и контрастности изображения путем изменения гамма-кривой, которая определяет, как яркость пикселей в изображении должна быть преобразована в значения яркости на экране; может использоваться как самостоятельный метод нормализации, так и в сочетании с другими методами.

В данной научной работе продемонстрируем сочетание таких методов нормализации изображения как минимаксная нормализация и гамма-коррекция. Данные методы обработки изображения были выбраны в качестве основных из-за своей популярности и распространённости а также простоты реализации, что непосредственно сказывается на простоте и скорости работы алгоритма преобразования.

Гамма-коррекция

Гамма-коррекция[2] - это процесс изменения гамма-кривой, которая описывает соотношение между яркостью входных сигналов и яркостью соответствующих выходных значений в изображении. Гамма-коррекция выполняется с целью корректировки нелинейного отображения яркости входных сигналов на экране и приведения его к более линейному виду. Это позволяет получить более точные результаты обработки изображений.

Гамма-коррекция особенно важна при обработке изображений, полученных с использованием технологий, основанных на использовании полупроводниковых материалов, таких как датчики изображения в тепловизионных камерах. Такие материалы могут иметь нелинейную зависимость между сигналом и яркостью, что приводит к искажениям в изображениях. Гамма-коррекция позволяет корректировать такие искажения и приводить изображения к более точному и линейному виду.

Процесс гамма-коррекции

Процесс гамма-коррекции включает в себя следующие шаги:

- Измерение гамма-кривой для определенного устройства или датчика изображения.
- Вычисление коэффициента гамма-коррекции на основе измеренной гамма-кривой.
- Применение коэффициента гамма-коррекции к каждому пикселю изображения, чтобы скорректировать его яркость.

- Вывод откорректированного изображения на экран или сохранение его в цифровом формате.

Шаги 1 и 2 выполняются один раз для каждого устройства или датчика изображения. Шаги 3 и 4 выполняются для каждого изображения, полученного с использованием данного устройства или датчика.

В зависимости от приложения, может быть использована различная гамма-кривая для выполнения гамма-коррекции. Например, для приложений в области кино и телевидения, используется стандартная гамма-кривая ($\gamma=2.2$), а для некоторых приложений в области компьютерной графики и мультимедиа может использоваться другая гамма-кривая.

Алгоритм работы программного продукта

Сам алгоритм преобразования кадров видеоряда состоит из следующих шагов:

1. Открытие потока на чтение из бинарного файла видеоряда
2. Вычисление количества пикселей в кадре
3. Чтение кадра
4. Вычисление верхней и нижней границы интенсивности пикселей в кадре
5. Применение минмаксной нормализации
6. Применение гамма-коррекции
7. Запись 3-х канального изображения с 255 оттенками серого
8. Завершение работы программного продукта

Шаги 2-8 повторяются до того момента, пока не закончатся данные в бинарном файле.

Используемые программные средства

Для реализации данного программного продукта понадобится открытая библиотека для обработки изображения, написанная для языка программирования C++, библиотека OpenCV. Конкретно пригодятся модули для работы с видеопотоками `videoio`, модуль для работы с изображением `imgproc`. На вход подаётся один бинарный файл, который помимо служебной информации в заголовке имеет 1000 кадров одной размерности. В каждом кадре один пиксель имеет значение интенсивности пикселя, которое и подлежит дальнейшим преобразованиям.

Полученное видеоряд выглядит следующим образом, представленным на рисунке 1:



Рисунок 1. — результат работы программного продукта
А кадр изображения при дополнительном использовании минмаксной нормализации
представлен на рисунке 2:



Рисунок 2. - применение минмаксной нормализации вместе с гамма-коррекцией

Заключение

Процесс обработки бинарных данных тепловизионной камеры включает несколько этапов, начиная от получения данных в бинарном формате и заканчивая составлением видео из полученных изображений. Один из важных этапов - это гамма-коррекция, которая позволяет корректировать нелинейное отображение яркости входных сигналов и получать более точные результаты обработки изображений.

Данный метод может иметь множество приложений, включая обнаружение неисправностей в промышленности, диагностику заболеваний в медицине, исследования в науке и технологиях, а также в различных других областях. Гамма-коррекция в данном процессе является важным шагом при обработке данных, который позволяет получать более точные и качественные результаты. Кроме того, развитие технологий обработки изображений и алгоритмов позволяет получать все более точные и эффективные методы обработки данных тепловизионных и прочих камер, что открывает новые возможности для исследований и приложений в различных областях.

Список использованных источников:

1. Аникин, А.А. *Цифровая обработка изображений*, 2006. – 400с.
2. А. И. Пономарев, А. А. Берестнев, *Цифровая обработка изображений: Начальный курс*, 2018 – 272с.

UDC

ANALYSIS AND APPLICATION OF METHODS OF THERMAL IMAGING PROCESSING OF VIDEO SIGNALS

Vashkevich N. V., Asadchy A.E.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V.D. – Assistant of the Department of Informatics

Annotation. This work aims to analyze the methods of thermal imaging processing of video signals, as well as the processing of these signals itself on the example of specific binary data that will need to be processed and translated into frame format, and then video.

Keywords. image processing, binary data processing, gamma correction, thermal imaging cameras, histogram normalization.

УДК

РАЗВЕРТЫВАНИЕ ПРИЛОЖЕНИЙ С ПОМОЩЬЮ ТЕХНОЛОГИЙ БЕССЕРВЕРНОЙ АРХИТЕКТУРЫ С ПРИМЕНЕНИЕМ ОБЛАЧНОЙ РАЗРАБОТКИ

Лаппо К.С., студент группы 153503, Владимирцев В. Д., ассистент каф. Информатики

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Владимирцев В.Д. – ассистент кафедры информатики

Аннотация. Бессерверная архитектура, особенно в области облачной разработки, набрала значительную популярность в последние годы. В данной работе мы предоставим обзор бессерверной архитектуры и ее последствий для облачной разработки. Также мы обсудим преимущества и недостатки бессерверного вычисления и предоставим примеры его использования в облачной разработке.

Ключевые слова. Бессерверная архитектура, облачная разработка, облачное вычисление, функции, масштабируемость, гибкость, экономичность, AWS Lambda, Microsoft Azure Functions, Google Cloud Functions.

Традиционная модель облачной разработки включает развертывание приложений на виртуальных машинах или контейнерах. Однако, этот подход имеет несколько ограничений, включая необходимость ручного масштабирования и обслуживания, что может привести к более высоким затратам и меньшей гибкости. Бессерверная архитектура [1] предлагает альтернативный подход к облачной разработке, позволяя разработчикам сосредоточиться на написании кода, а не на управлении инфраструктурой.

Бессерверная архитектура относится к модели облачного вычисления, где провайдер облака управляет инфраструктурой и автоматически выделяет ресурсы по мере необходимости. В этой модели разработчики пишут код в виде функций, которые запускаются при событиях, таких как HTTP-запросы или обновления базы данных. Провайдер облака автоматически масштабирует инфраструктуру для обработки нагрузки, и разработчики платят только за ресурсы, используемые во время выполнения функции.

Бессерверное вычисление [2] предлагает несколько преимуществ по сравнению с традиционной облачной разработкой. Они включают в себя снижение затрат, увеличение масштабируемости и улучшение гибкости. Поскольку разработчики платят только за ресурсы, используемые во время выполнения функции, они могут значительно снизить затраты по сравнению с традиционной облачной разработкой. Автоматическое масштабирование ресурсов также позволяет приложениям обрабатывать большой объем трафика без ручного вмешательства [3]. Кроме того, бессерверное вычисление улучшает гибкость, сокращая время, необходимое для управления инфраструктурой и позволяя разработчикам сосредоточиться на написании кода и доставке ценности пользователям.

Несмотря на преимущества, бессерверное вычисление также имеет некоторые недостатки. Они включают ограниченный контроль над инфраструктурой, увеличенную сложность проектирования приложений и время инициализации. Поскольку провайдер облака управляет инфраструктурой, разработчики имеют ограниченный контроль над аппаратным и программным обеспечением. Это может привести к ограничениям в части настройки и конфигурации. Кроме того, сложность бессерверных приложений может сделать их более трудными в проектировании и тестировании.

Несколько провайдеров облачных услуг предлагают платформы бессерверного вычисления, включая Amazon Web Services (AWS) Lambda [4], Microsoft Azure Functions [5] и Google Cloud Functions [6]. Эти платформы позволяют разработчикам писать функции на различных языках программирования, включая Node.js, Python и Java. Эти функции могут использоваться для различных случаев использования облачной разработки, включая веб-приложения, обработку данных и приложения интернета вещей (IoT).

Разработанный проект RaXiom является веб-приложением для определения раковых опухолей на основе МРТ снимков пациентов, разработанным с использованием бессерверной архитектуры и облачных вычислений. Мы использовали Google Cloud Platform (GCP) и Google App Engine (GAE) [7] для развертывания нашего приложения, что позволяет без затрат на настройку инфраструктуры получить доступ к проекту в сети. В основе бессерверного вычисления находится Firebase с его сервисами Firebase Firestore, NoSQL база данных с полной интеграцией с GCP, и Firebase Functions для настройки обработчиков событий.

Структура Raxiom была настроена так, чтобы минимизировать затраты на инфраструктуру, так как использование облачных сервисов берет плату за отдельные сервисы и их использование, а также максимальную гибкость и безопасность. Так же стоит учесть, что провайдеры облачных услуг часто используют систему скидок и других привилегий, которые влияют на стоимость сервисов. Таким образом общие затраты Raxiom за последний месяц представлены на рисунке 1.

Service	Cost	Discounts	Promotions and others	Subtotal	
Compute Engine	\$0.01	\$0.00	-\$0.01	\$0.00	
Cloud Healthcare	\$0.00	\$0.00	\$0.00	\$0.00	
Cloud Build	\$0.00	\$0.00	\$0.00	\$0.00	
Cloud Storage	\$0.52	\$0.00	-\$0.52	\$0.00	
Vertex AI	\$0.00	\$0.00	\$0.00	\$0.00	
Networking	\$0.56	-\$0.56	\$0.00	\$0.00	
App Engine	\$11.72	\$0.00	-\$11.72	\$0.00	
Cloud Logging	\$0.00	\$0.00	\$0.00	\$0.00	
				Subtotal	\$0.00
				Tax	—
				Total	\$0.00

Рисунок 1 – Общие затраты на инфраструктуру и поддержку Raxiom на Апрель 2023 г.

Для организации обмена данными использовался Apollo GraphQL, который позволяет упростить взаимодействие между клиентом и сервером, подробная иллюстрация технологий для обмена данных указана ниже. Для Frontend части приложения был использован NextJS который предоставляет возможность серверного рендеринга. На рисунке 2 представлена статистика среднего времени выполнения запросов к облачным сервисам и их производительность.

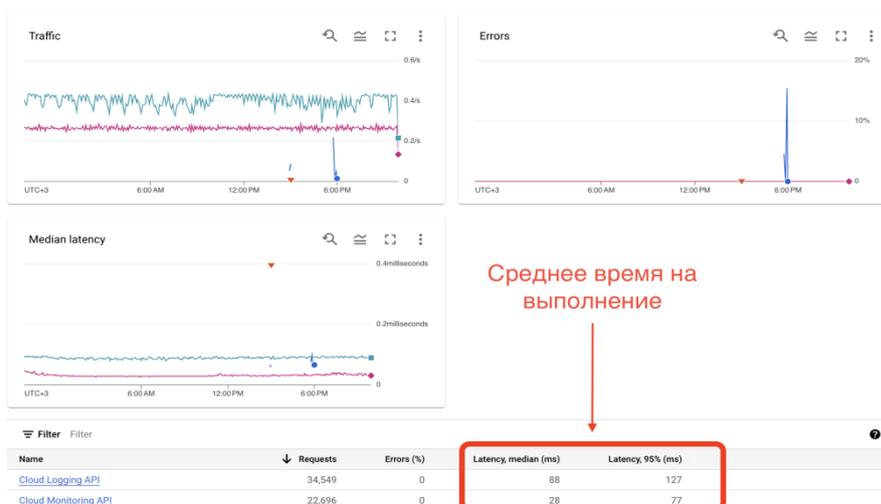


Рисунок 2 – Общие затраты на инфраструктуру и поддержку Raxiom на Апрель 2023 г.

В результате проект показал, что использование бессерверной архитектуры и облачных вычислений может значительно упростить создание и развертывание веб-приложений. Мы смогли создать безопасное и масштабируемое приложение, которое может обрабатывать большой объем трафика без необходимости ручного масштабирования.

Заключение. Как итог бессерверное вычисление перевернуло облачную разработку, предоставляя разработчикам более эффективный и экономичный подход к созданию приложений. Несмотря на некоторые ограничения бессерверного вычисления, его преимущества делают его привлекательным вариантом для многих случаев использования облачной разработки. С продолжением роста облачных вычислений, вероятно, что бессерверное вычисление станет еще более распространенным в следующие годы.

Список использованных источников:

1. Roberts, J. (2019). *Serverless Architecture: A Comprehensive Guide*. O'Reilly Media, Inc.
2. Kalyanaraman, V. (2020). *Cloud Computing: A Hands-On Approach*. CRC Press.
3. Appleby, S. (2018). *Enterprise Serverless: Designing Applications for the Cloud*. O'Reilly Media, Inc.
4. Amazon Web Services. (2021). *AWS Lambda documentation* – [Электронный ресурс] Электронные данные. – Режим доступа: <https://aws.amazon.com/lambda/>
5. Microsoft Azure. (2021). *Azure Functions. documentation* – [Электронный ресурс] Электронные данные. – Режим

доступна: <https://azure.microsoft.com/en-us/services/functions/>

6. Google Cloud. (2021). *Cloud Functions. documentation* – [Электронный ресурс] Электронные данные. – Режим

доступа: <https://cloud.google.com/functions>

7. Google App Engine. (2021). *documentation* – [Электронный ресурс] Электронные данные. – Режим доступа:

<https://cloud.google.com/appengine>

UDC

APPLICATION DEPLOYMENT USING SERVERLESS ARCHITECTURE TECHNOLOGIES USING CLOUD DEVELOPMENT

Lappo.K.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Vladymtsev V.D. – Assistant of the Department of Informatics

Annotation. Serverless architecture, especially in the field of cloud development, has gained significant popularity in recent years. In this paper, we will provide an overview of the serverless architecture and its implications for cloud development. We will also discuss the advantages and disadvantages of serverless computing and provide examples of its use in cloud development.

Keywords. Serverless, cloud development, cloud computing, functions, scalability, flexibility, cost-effectiveness, AWS Lambda, Microsoft Azure Functions, Google Cloud Functions.

РАЗВИТИЕ И ОЦЕНКА НЕЙРОСЕТЕВЫХ МЕТОДОВ ВЕРИФИКАЦИИ СОБСТВЕННОРУЧНОЙ ПОДПИСИ

*Мискевич П.Л., магистрант гр.256241, Петровец В.Н., магистрант гр.256241,
Раловец А.А., магистрант гр.256241*

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Волорова Н.А. – канд. техн. наук, доцент

Аннотация. Рукописная подпись является уникальной биометрической характеристикой, широко используемой для подтверждения личности и согласия на различные операции. В связи с ростом киберугроз и подделок подписей возникает потребность в надежных и эффективных системах верификации подписи. В данной статье обсуждается развитие нейросетевых методов верификации собственноручной подписи, анализируются существующие подходы, выявляются проблемы и рассматриваются возможные перспективы.

Ключевые слова. Верификация подписи, нейронные сети, биометрическая аутентификация, киберугрозы, выявление подделок.

Введение. С ростом числа электронных документов и транзакций увеличилась и вероятность мошенничества с использованием поддельных подписей. Подделка подписей может привести к серьезным финансовым и правовым последствиям для жертв и организаций. В связи с этим, защита от подделок стала одним из приоритетных направлений в области кибербезопасности. Киберугрозы также стали серьезным вызовом для обеспечения надежности систем верификации подписи, так как злоумышленники могут использовать различные методы для обхода защиты и получения доступа к конфиденциальной информации.

Целью данного исследования является обзор современных методов верификации собственноручной подписи, основанных на нейросетях, анализ их преимуществ и недостатков в сравнении с классическими методами машинного обучения. А также определение потенциальных проблем и перспектив в области верификации подписи с использованием нейросетевых подходов, чтобы предоставить рекомендации для дальнейших исследований и разработок в этой сфере.

Основная часть. Ранние методы верификации рукописной подписи в основном основывались на экспертной оценке. Судебные эксперты-криминалисты изучали отличительные черты подписи, такие как форма букв, сложность линий, скорость и давление письма, а также наклон и размеры символов. В этих методах были включены статистические подходы и методы сравнения черт рукописи. Однако эти методы имеют ряд ограничений, таких как высокая степень субъективности, низкая масштабируемость и невысокая точность.

С развитием технологий и распространением компьютеров, внимание исследователей переключилось на автоматические методы верификации подписи, основанные на машинном обучении. Эти подходы используют алгоритмы, основанные на большом количестве образцов, для анализа и сравнения черт подписей. В первых исследованиях были применены различные методы машинного обучения, такие как k-ближайших соседей (k-NN), опорные вектора (SVM) и деревья решений [1].

В последние годы нейронные сети зарекомендовали себя как мощный инструмент для распознавания образов. Это привело к развитию различных нейросетевых подходов к верификации собственноручной подписи, включая следующие:

- сверточные нейронные сети (CNN);
- рекуррентные нейронные сети (RNN);
- сети с долгой краткосрочной памятью (LSTM);
- сети глубокого обучения с подкреплением (DRL);
- трансформеры (Transformer Neural Networks).

Сверточные нейронные сети (CNN) являются типом глубоких нейронных сетей, специализирующихся на анализе визуальных образов. Они состоят из последовательности сверточных, пулинговых и полносвязных слоев (таблица 1), позволяющих автоматически извлекать иерархические признаки из изображений [2].

Таблица 1 – Виды слоев, используемых в сверточных нейронных сетях

Слой	Описание
Входной слой	Принимает изображение подписи
Сверточный слой	Применяет фильтры для извлечения локальных признаков
Пулинговый слой	Уменьшает размерность данных, сохраняя важные признаки

Полносвязный слой	Производит классификацию извлеченных признаков
Выходной слой	Возвращает вероятность того, что подпись является подлинной

CNN хорошо справляются с распознаванием различных стилей подписей и устойчивы к масштабированию, вращению и другим преобразованиям изображений.

На рисунке 1 отображен пример модели сверточной нейронной сети для распознавания и верификации собственноручной подписи.

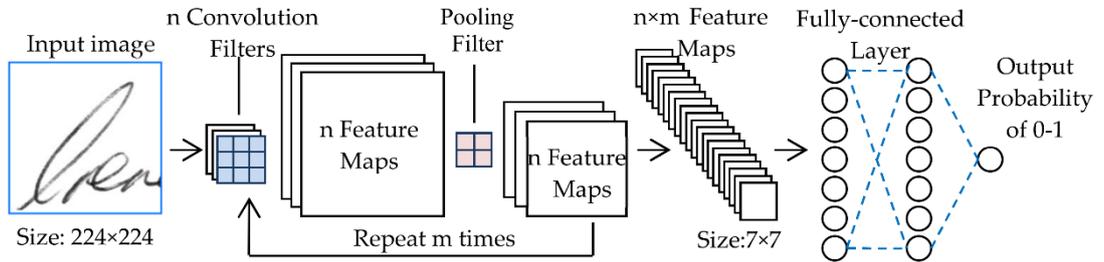


Рисунок 1 – Пример модели сверточной нейронной сети для верификации собственноручной подписи

На вход подается изображение подписи фиксированного размера. На следующем этапе входное изображение обрабатывается с помощью набора сверточных фильтров (ядер). Сверточные фильтры (convolutional filters) скользят по входным данным, применяя операцию свертки. Результаты свертки формируют карты признаков (feature maps). Далее пулинговый слой (pooling layer) уменьшает размерность карт признаков, сохраняя при этом важные признаки. Это делается путем применения агрегирующей операции, такой как максимальное или среднее значение, к непересекающимся областям карты признаков. Пулинг уменьшает вычислительную сложность модели и делает ее менее чувствительной к незначительным изменениям входных данных. После нескольких сверточных и пулинговых слоев, полученные карты признаков передаются в один или несколько полносвязных слоев, которые представляют собой традиционные многослойные перцептроны. Выходной слой данной модели содержит один нейрон, определяющий, является ли представленная на изображении подпись подлинной или поддельной.

Рекуррентные нейронные сети (RNN) разработаны для работы с последовательными данными, такими как временные ряды или текст. В контексте верификации подписи RNN обрабатывают данные о траектории движения пера, такие как координаты, скорость и давление [3]. Данный тип сети хорошо учитывает временные зависимости и может обнаруживать подделки, сделанные с использованием различных техник мошенничества. Виды слоев, используемых в рекуррентных нейронных сетях, представлены в таблице 2.

Таблица 2 – Виды слоев, используемых в рекуррентных нейронных сетях

Слой	Описание
Входной слой	Принимает изображение подписи
Рекуррентный слой	Обрабатывает последовательность, сохраняя информацию о предыдущих состояниях
Выходной слой	Возвращает вероятность того, что подпись является подлинной

Сети с долгой краткосрочной памятью (LSTM) являются разновидностью рекуррентных нейронных сетей, которые специально разработаны для обработки долгосрочных зависимостей в данных. Они состоят из специальных ячеек памяти, которые позволяют им сохранять и обновлять информацию на длительных промежутках времени. LSTM особенно подходят для анализа сложных и изменчивых характеристик рукописной подписи, таких как скорость и давление.

Сети глубокого обучения с подкреплением (DRL) используют стратегию обучения, основанную на опыте, для оптимизации процесса принятия решений. В контексте верификации подписи DRL может быть использован для адаптации и улучшения процесса извлечения признаков и классификации. Данный тип сети может адаптироваться к новым и изменяющимся обстоятельствам, что делает его потенциально полезным для обнаружения современных подделок и мошенничества.

Трансформеры (Transformer Neural Networks) представляют собой относительно новый класс нейросетевых моделей, основанных на механизме внимания. Они позволяют моделям уделять больше внимания определенным частям данных, что может улучшить качество верификации подписи. Трансформеры были успешно применены в задачах обработки естественного языка и начинают использоваться в области верификации подписей [4]. Варианты архитектур трансформеров, такие как BERT и GPT, могут быть адаптированы для работы с изображениями и последовательностями данных, связанными с рукописными подписями.

Схема разработанной системы верификации собственноручной подписи представлена на рисунке 2.

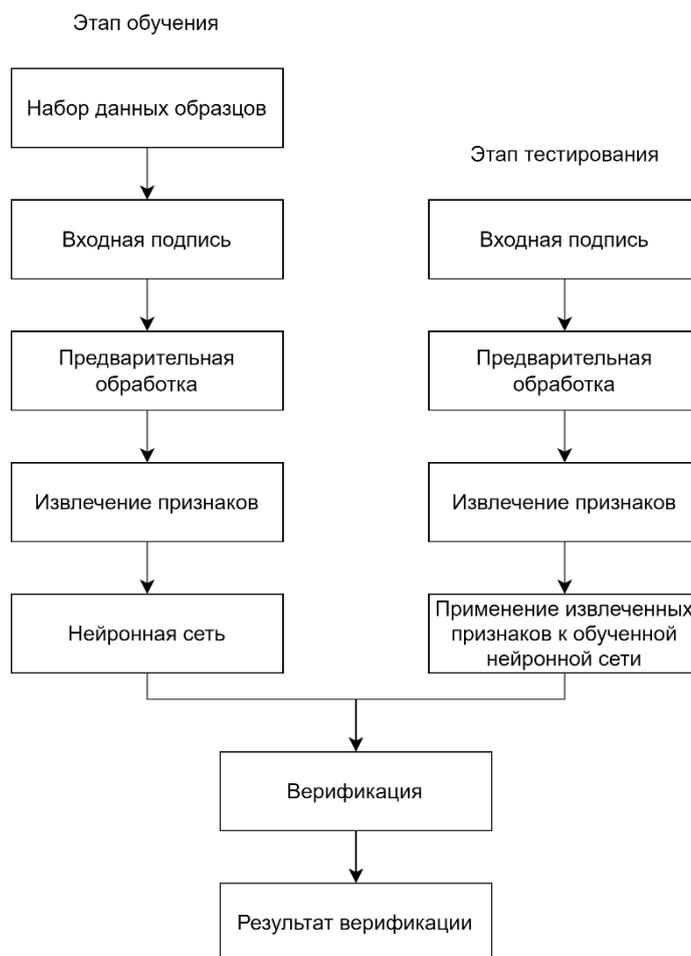


Рисунок 2 – Схема системы верификации собственноручной подписи с использованием нейронной сети

Для оценки эффективности нейросетевых и классических методов машинного обучения в задаче верификации рукописных подписей можно рассмотреть следующие критерии сравнения: точность и скорость.

Точность является одним из наиболее важных показателей эффективности алгоритма верификации подписей. Она определяет процент правильно классифицированных подписей (как подлинных, так и поддельных) от общего числа подписей в тестовом наборе данных. В целом, нейросетевые методы демонстрируют более высокую точность в задаче верификации рукописных подписей по сравнению с классическими методами машинного обучения. Это связано с тем, что нейросетевые методы могут автоматически извлекать иерархические признаки и учитывать сложные зависимости в данных, что делает их более мощными и адаптивными к различным стилям подписей и видам подделок.

Скорость относится к времени, необходимому для обработки и верификации подписи. В реальных условиях, особенно в системах, где требуется быстрое принятие решений, скорость является критическим фактором. Скорость может быть измерена как время обучения модели и время инференции (предсказания) для каждой подписи. В плане скорости обучения и инференции, классические методы машинного обучения, такие как SVM, деревья решений и k-NN, могут быть быстрее нейросетевых подходов, особенно на небольших наборах данных. Однако с увеличением объема данных и сложности задачи, нейросетевые методы могут обеспечивать более быстрое обучение и предсказание благодаря их распределенной и параллельной архитектуре.

Существующие подходы к верификации рукописных подписей, включая нейросетевые методы, имеют ряд недостатков, которые могут влиять на их эффективность и применимость в реальных условиях, а именно: требование больших объемов данных, вычислительная сложность и устойчивость к атакам и подделкам.

Перспективными направлениями для развития новых методов верификации собственноручной подписи можно рассмотреть федеративное, активное и однокадровое типы обучения.

Федеративное обучение (federated learning) может помочь решить проблемы сбора данных и защиты конфиденциальности пользователей. Оно позволяет обучать нейросетевые модели на данных, которые распределены между различными устройствами и организациями, не требуя централизованного хранения данных [5].

Активное обучение (active learning) может помочь решить проблемы с разметкой больших объемов данных. При активном обучении модель сама определяет, какие примеры наиболее полезны для обучения, и запрашивает разметку только для них, что сокращает время и стоимость разметки данных.

Однокадровое обучение (one-shot learning) может быть полезным для обучения моделей верификации подписи, когда доступно ограниченное количество образцов подписей для каждого пользователя [6]. Однокадровое обучение фокусируется на быстром адаптации модели к новым данным с минимальным количеством примеров, что может обеспечить эффективное обучение даже при небольших наборах данных.

Для улучшения общей надежности и безопасности системы аутентификации можно интегрировать верификацию собственноручной подписи с другими биометрическими системами, такими как распознавание отпечатков пальцев, распознавание лиц и распознавание голоса. Многофакторная аутентификация может быть использована для усиления защиты и повышения устойчивости к атакам и подделкам.

Заключение. В данном исследовании был проведен обзор существующих подходов к верификации собственноручной подписи, с акцентом на нейросетевых методах, таких как сверточные нейронные сети, рекуррентные нейронные сети, сети с долгой краткосрочной памятью, сети глубокого обучения с подкреплением и трансформеры. Эти методы были сравнены с классическими методами машинного обучения по критериям, таким как точность, скорость, устойчивость к атакам и подделкам. Также были рассмотрены их преимущества и недостатки.

Для дальнейших исследований и разработок в области верификации собственноручной подписи рекомендуется:

- исследовать и разрабатывать новые архитектуры нейросетей и методы обучения;
- рассмотреть применение федеративного, активного и однокадрового обучения для решения проблем сбора и разметки данных, а также для улучшения эффективности обучения моделей;
- исследовать возможности интеграции верификации собственноручной подписи с другими биометрическими системами, чтобы создать многофакторные системы аутентификации с повышенной безопасностью и надежностью.

В целом, исследование нейросетевых методов верификации собственноручной подписи является важным шагом на пути к созданию более безопасных, надежных и удобных систем аутентификации, которые могут быть широко использованы в различных отраслях и сферах деятельности, таких как банковское дело, юриспруденция и электронная коммерция.

Список использованных источников:

1. Impedovo, D. Automatic signature verification: The state of the art / D. Impedovo, & G. Pirlo // *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2008. – P. 609-635.
2. Graves, A. Learning features for offline handwritten signature verification using deep convolutional neural networks / L.G. Hafemann, R. Sabourin, L.S. Oliveira // *Pattern Recognition*, 2017. – P. 163-176.
3. Graves, A. Offline handwriting recognition with multidimensional recurrent neural networks / A. Graves, J. Schmidhuber // *Advances in Neural Information Processing Systems*, 2009. – P. 545-552.
4. Yousef, M. Handwritten Text Recognition using Transformers / M. Yousef, M.M. Abdelsamea, G.A. Gnecco // *Proceedings of the 29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2021. – P. 371-376.
5. McMahan, H.B. Communication-Efficient Learning of Deep Networks from Decentralized Data / H.B. McMahan [at el.] // *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017. – P. 371-376.
6. Vinyals, O. Matching networks for one shot learning / O. Vinyals [at el.] // *Advances in neural information processing systems*, 2016. – P. 3630-3638.

DEVELOPMENT AND EVALUATION OF NEURAL NETWORK METHODS FOR HANDWRITTEN SIGNATURE VERIFICATION

Miskevich P.L., Petravets U.N., Ralovets A.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Volorova N.A. – PhD in Engineering, Associate Professor

Annotation. Handwritten signature is a unique biometric feature, widely used for verifying identity and consent for various operations. With the rise of cyber threats and signature forgery, there is a need for reliable and efficient signature verification systems. In this article, the development of neural network methods for handwritten signature verification is discussed, existing approaches are analyzed, challenges are identified, and possible prospects are considered.

Keywords. Signature verification, neural networks, biometric authentication, cyber treats, forgery detection.

УДК 004.93

ПРИМЕНЕНИЕ МЕТОДОВ ЦИФРОВОЙ ОБРАБОТКИ СИГНАЛОВ И МАШИННОГО ОБУЧЕНИЯ ДЛЯ РАННЕЙ ДИАГНОСТИКИ ОТКЛОНЕНИЙ В РАБОТЕ ДВИГАТЕЛЯ

*Раловец А.А., магистрант гр. 256241, Мискевич П.Л., магистрант гр. 256241,
Деменковец Д.В., аспирант*

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Волорова Н.А. – канд. техн. наук, доцент

Аннотация. Статья посвящена исследованию возможности применения цифровой обработки звукового сигнала автомобиля для ранней диагностики неисправностей двигателя. Для автоматического выявления отклонений в работе двигателя было предложено использовать анализ частотных характеристик звука в сочетании с методами машинного обучения. Результаты эксперимента показали работоспособность предложенного метода.

Ключевые слова. Неисправности двигателя, цифровая обработка сигналов, частотный анализ, спектральный анализ, машинное обучение, модель гауссовых смесей.

Ранняя диагностика неисправностей автомобиля является важной задачей для обеспечения безопасности дорожного движения и экономической эффективности эксплуатации автомобилей. Неисправности, которые не были обнаружены и устранены вовремя, могут привести к авариям, повреждению оборудования и необходимости замены дорогостоящих деталей. С применением спектрального анализа можно выделить определенные амплитудно-частотные характеристики звука двигателя, которые могут указывать на наличие неисправностей. Для автоматизации процесса обнаружения отклонений в работе двигателя необходимо провести выбор оптимального метода машинного обучения.

С помощью микрофона, расположенного рядом с двигателем автомобиля, происходит преобразование аналогового звукового сигнала в цифровой вид с частотой дискретизации 44100 Гц и квантованием 16 бит [1]. Результатом является файл формата Waveform Audio File Format [2]. Преимуществом формата WAV является сохранение звукового файла без потерь качества. Визуализация записанного звукового сигнала представлена на рисунке 1.

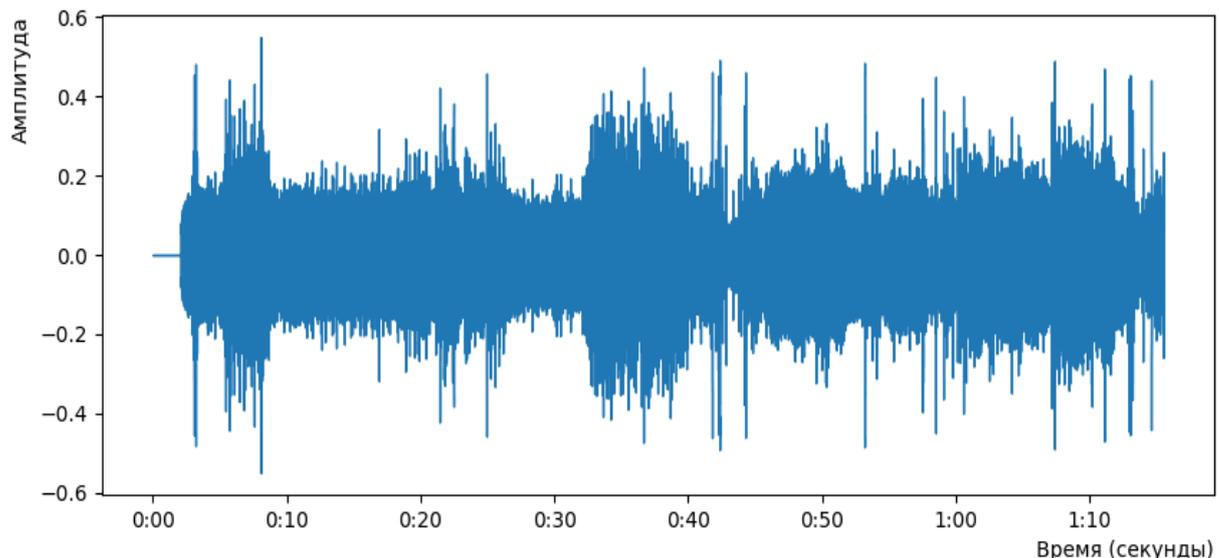


Рисунок 1 – Звуковой сигнал работы двигателя автомобиля

Для получения спектрограммы из представления звукового сигнала было выполнено преобразование Фурье (2048 значений) [3]. Входными данными для алгоритма быстрого преобразования Фурье являлся массив комплексных чисел, действительная часть которых – образцы оцифрованного сигнала, принадлежащие одному окну; мнимая часть – нулевая. После заполнения исходной комплексной матрицы, содержащей информацию об одном окне, выполняется БПФ-анализ для этого окна.

На основе полученного частотного спектра сигнала строится спектрограмма [4]. Каждые 93 миллисекунды исходного сигнала соответствуют вертикальной линии на изображении спектрограммы (рисунок 2).

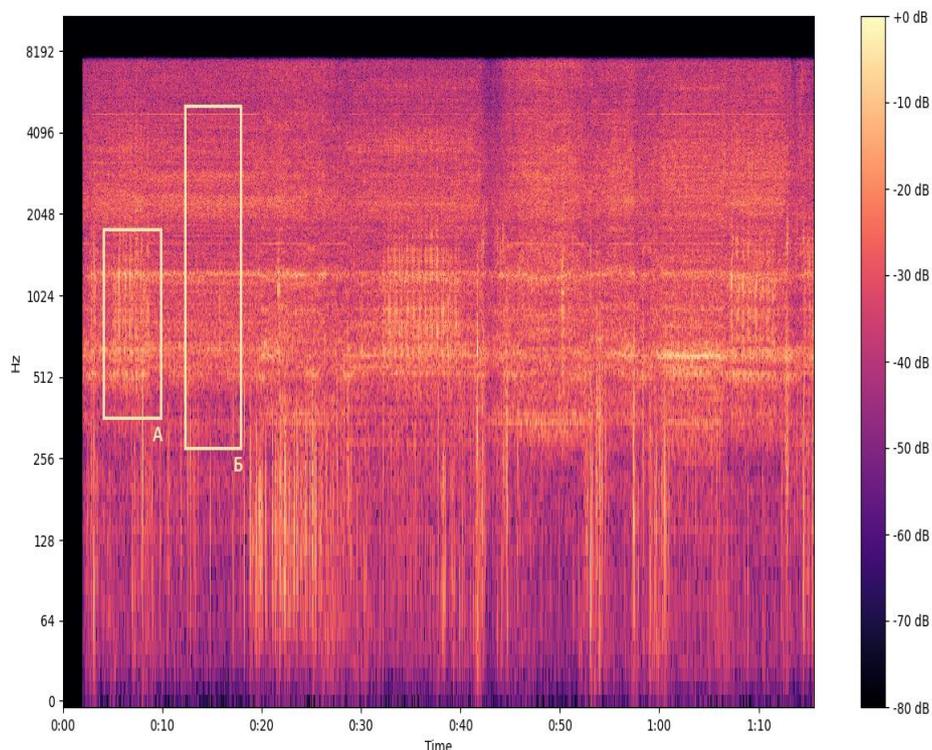


Рисунок 2 – Спектрограмма звукового сигнала работы двигателя

Запись звуковой дорожки была произведена в условиях повышенного уровня шума на открытом воздухе, поэтому на спектрограмме заметны искажения и это несколько усложняет задачу анализа. Однако в полученной спектрограмме явно выделяются частоты, преобладающие в исследуемом звуке. В исследуемой записи были выявлены основные частоты работы двигателя без дефектов холостого хода: 600, 1200, 2400 и 4800 Гц (область «Б» на рисунке 2).. Звуки неисправного двигателя (стуки) были выявлены в промежутке времени с 5 по 8 секунду записи на частотах 600-1200 Гц (область «А» на рисунке 2).

Появилась необходимость выбрать алгоритм, который будет учитывать наличие шумов в записи и будет способен обнаружить не только стук, но и другие отклонения. Возникает задача бинарной классификации, которая заключается в определении наличия дефекта в работе двигателя. Задачу можно решить с помощью модели гауссовых смесей [5]. Этот алгоритм хорошо подходит для решения задач кластеризации, особенно в случаях, когда кластеры имеют сложную форму. Его необходимо применить для распределения акустических событий по двум кластерам – "стационарная работа" и "неисправность".

Для обучения модели Гауссовых смесей было выбрано три компонента смеси [6]. Это число компонент было выбрано на основе экспериментов и оценки качества модели на тестовых данных. Идея модели заключается в описании данных сложным распределением, которое является линейной комбинацией нескольких многомерных нормальных распределений.

В случае работы двигателя звук его работы можно рассматривать как стационарный, что делает возможным описание этого звука с помощью такого распределения. Для подбора параметров модели, таких как тип ковариационной матрицы, был использован ВИС критерий [7]. Однако в данном случае оптимальные параметры, подобранные с помощью этого критерия, оказались менее эффективными, чем параметры, подобранные экспериментальным путём.

После обучения модели можно оценить степень схожести распределения некоторого неизвестного спектра на распределение спектра при нормальной работе двигателя. Для этого необходимо вычислить среднее правдоподобие столбцов спектрограммы исследуемого сигнала и подобрать порог, который будет отделять правдоподобие звуков хорошей работы от всех

остальных. Для каждой секунды можно определить коэффициент правдоподобия, сравнивая спектрограммы и вычисляя, в какой степени данные соответствуют распределению. Зависимость коэффициента правдоподобия сигнала, а также моменты начала и конца сигнала с неисправностью изображены на рисунке 3.

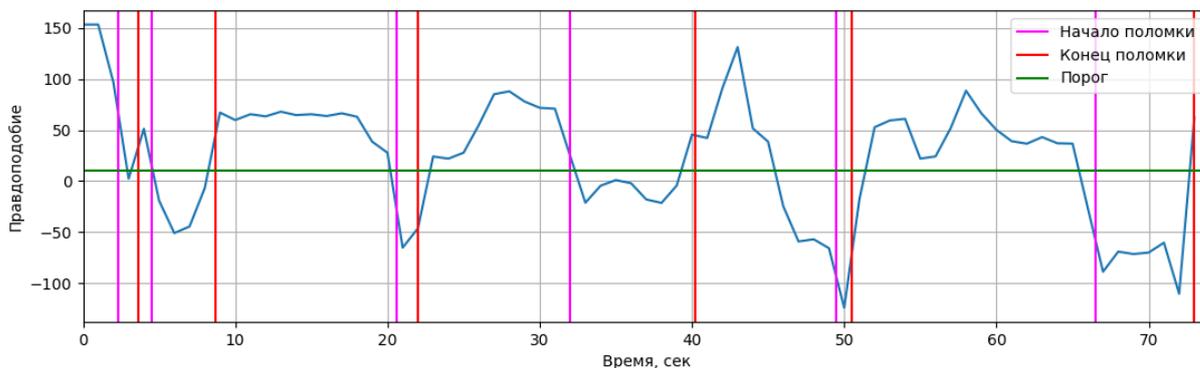


Рисунок 3 – Зависимость коэффициента правдоподобия сигнала, моменты начала и конца сигнала двигателя с неисправностью

Из графика зависимости коэффициента правдоподобия от времени на рисунке 3 видно, что в моменты звучания неисправности (стука) коэффициент правдоподобия записи становится ниже порога, что означает способность алгоритма различать записи со звуком неисправности и без неё.

Однако, стоит отметить, что местами значения правдоподобия находятся достаточно близко к пороговому значению и могут быть влиянием других звуков, таких как посторонние шумы. Это также может повлиять на величину коэффициента правдоподобия.

Таким образом, применение цифровой обработки сигналов и методов машинного обучения является достаточно перспективным направлением в области информационных технологий и диагностики автомобильной техники. Результаты выполненной работы показали, что метод модели гауссовых смесей может быть использован для выявления неисправностей в работе двигателя по звуковому сигналу. Однако, для более точной и надежной диагностики необходимо дополнительно расширять базу данных звуков и проводить дальнейшие исследования в этой области.

Список использованных источников:

1. Lyons, R. *Understanding Digital Signal Processing*. / Richard G. Lyons, 2010. – 954 p.
2. WAVE pcm soundfile format [Electronic resource]. – Mode of access: <http://soundfile.sapp.org/doc/WaveFormat/> – Date of access: 06.04.2023.
3. *Integral Calculations – EqWorld* [Electronic resource]. – Mode of access: <http://eqworld.ipmnet.ru/ru/auxiliary/aux-intrans.htm> – Date of access: 07.04.2023.
4. Oppenheim, A. V., Schaffer, R. W. *Discrete-time signal processing*. Pearson. / Alan V. Oppenheim, 2010.
5. Reynolds, D. A., *Gaussian mixture models*. Encyclopedia of biometrics. / David Reynolds, 2009.
6. Fraley, C., & Raftery, A. E., *Model-based clustering, discriminant analysis, and density estimation*. Journal of the American statistical Association / Chris Fraley, 2002. – 611-631 pp.
7. Schwarz, G.. *Estimating the dimension of a model*. The Annals of Statistics. / Gerald Walter Schwarz, 1978 г. – 461-464 pp.

UDC 004.93

APPLICATION OF DIGITAL SIGNAL PROCESSING AND MACHINE LEARNING METHODS FOR EARLY DIAGNOSIS OF DEVIATIONS IN ENGINE OPERATION

Ralovets A.A., Miskevich P.L., Demenkovets D.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Volorova N.A. – PhD in Engineering, Associate Professor

Annotation. The article is dedicated to studying the possibility of using digital processing of vehicle sound signals for early diagnosis of engine malfunctions. To automatically detect deviations in engine operation, the analysis of sound frequency characteristics in combination with machine learning methods was proposed. The experimental results demonstrated the effectiveness of the proposed method.

Keywords. Engine malfunctions, digital signal processing, frequency analysis, spectral analysis, machine learning, Gaussian mixture model.

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМА ШИФРОВАНИЯ IDEA В СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМАХ

Сытько М.В, студент гр.253504, Жак М.В, студент гр. 253504

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Стройникова Е. Д. – ст. преп. кафедры информатики

Аннотация. Данная работа посвящена исследованию эффективности алгоритма шифрования IDEA (International Data Encryption Algorithm) в современных информационных системах. В работе был произведен обзор существующих методов шифрования и проведено сравнение IDEA с альтернативными алгоритмами. Также была проведена аналитическая оценка производительности алгоритма IDEA на различных платформах и в различных условиях. В результате исследования было установлено, что IDEA является достаточно эффективным алгоритмом шифрования, обладающим высоким уровнем безопасности и подходящим для использования в современных информационных системах. Однако, были также выявлены некоторые ограничения и недостатки алгоритма IDEA, которые необходимо учитывать при его использовании. Данная работа представляет важный вклад в развитие теории и практики криптографии и информационной безопасности.

Ключевые слова. Защита данных, шифрование, IDEA, информационные системы, эффективность, методология, сравнение алгоритмов, информационная безопасность, централизованные и децентрализованные системы, клиент-серверная архитектура, производительность, вредоносные программы, ключи

Введение.

Защита данных сегодня играет критически важную роль, поскольку все больше информации хранится и передается онлайн. Во многих информационных системах методы шифрования являются необходимыми для обеспечения безопасности информации. Существует множество различных алгоритмов шифрования, и каждый из них имеет свои преимущества и недостатки.

Цель данной статьи - исследование эффективности алгоритма шифрования IDEA (International Data Encryption Algorithm) в современных информационных системах. Алгоритм IDEA является одним из наиболее распространенных алгоритмов шифрования и широко применяется в различных информационных системах.

В данной статье будет представлен обзор современных информационных систем, а также особенностей и уязвимостей, которые могут повлиять на выбор алгоритма шифрования. Будет описана методология исследования, проведенных экспериментов, и анализ результатов. Также будет проведено сравнение с другими алгоритмами шифрования, чтобы оценить эффективность IDEA в современных информационных системах.

Данная статья будет полезна для специалистов в области информационной безопасности, разработчиков программного обеспечения и всех, кто интересуется вопросами защиты данных в информационных системах.

Обзор современных информационных систем.

Современные информационные системы можно классифицировать как централизованные и децентрализованные. В централизованных системах все запросы отправляются на единый сервер, на котором хранятся данные. Децентрализованные системы распределяют данные между многочисленными серверами и позволяют любому серверу обрабатывать запросы.

Клиент-серверная архитектура является одной из самых популярных централизованных информационных систем, в которой клиенты отправляют запросы на центральный сервер. Примерами клиент-серверных систем являются веб-серверы, почтовые и файловые серверы.

Децентрализованные системы становятся все более популярными благодаря своей масштабируемости и устойчивости к отказам. Peer-to-peer (P2P) сеть - это один из видов децентрализованных систем, в которой каждый узел сети может функционировать как клиент и сервер одновременно. Сети P2P используются для обмена файлами, онлайн-игр и коммуникационных приложений. [1]

Безопасность данных, независимо от типа информационной системы, имеет решающее значение. Для предотвращения несанкционированного доступа или перехвата данных многие информационные системы используют методы шифрования. Выбор алгоритма шифрования зависит от различных факторов, таких как тип передаваемых данных, вычислительные ресурсы и требуемый уровень безопасности.

В последние годы использование алгоритмов шифрования стало еще более важным из-за растущей угрозы кибератак. Злоумышленники могут использовать различные методы, такие как атаки на основе связанных ключей, интерполяционные атаки и атаки «brute-force», для попытки взлома алгоритмов шифрования и получения доступа к конфиденциальным данным.

Исследование эффективности алгоритма IDEA.

В рамках исследования были проведены эксперименты, чтобы оценить эффективность алгоритма IDEA в современных информационных системах.

Для проведения экспериментов были использованы различные сценарии, включая передачу данных внутри одной локальной сети, передачу данных между различными сетями, а также передачу данных через Интернет. В каждом сценарии использовалась модель информационной системы, аналогичная той, которую используют в реальном мире, а также реализации алгоритма IDEA на нескольких языках программирования, таких как C++, Java и Python. Это было сделано для того, чтобы оценить, как различные реализации могут влиять на производительность алгоритма.

В ходе работы использовались различные виды данных, включая текстовые документы, изображения и видео. Размеры передаваемых файлов также были разными, от небольших файлов до файлов большого размера.

Проведение экспериментов позволило достичь необходимой статистической значимости результатов. Все тесты проводились на одинаковом аппаратном и программном обеспечении, гарантируя тем самым консистентность результатов. Таким образом, была создана методика оценки производительности и актуальности алгоритма IDEA в различных условиях, а также проведено сравнение его с другими алгоритмами шифрования.

Сравнение с другими алгоритмами шифрования.

Сравнение алгоритма IDEA с другими алгоритмами шифрования, такими как AES, Blowfish и RSA, позволяет понять преимущества и недостатки каждого из них. Для более глубокого понимания темы, рассмотрим подробнее каждый из сравниваемых алгоритмов

Advanced Encryption Standard (AES) является наиболее распространенным алгоритмом шифрования, используемым в современных информационных системах. Он был разработан как замена для DES (Data Encryption Standard), который был уязвим к атакам, и с тех пор стал стандартом для защиты информации во многих сферах, включая финансы, правительство и оборону. Основное преимущество AES заключается в том, что он является очень быстрым и эффективным, при этом обеспечивая высокий уровень безопасности. Однако AES может быть уязвим к атакам, основанным на вычислительной мощности, а также к атакам, основанным на криптоанализе.

Blowfish - это алгоритм симметричного шифрования, разработанный Брюсом Шнайером в 1993 году. Он используется для защиты данных в сети Интернет, электронной почте и файловых системах. В начале работы Blowfish инициализируется подготовкой ключа. Из ключа создается расширенный ключ, который состоит из нескольких субключей. Затем открытый текст разбивается на блоки длиной 64 бита. Каждый блок проходит через несколько раундов шифрования, каждый из которых состоит из подстановок и перестановок. После окончания раундов шифрования происходит перестановка бит в блоке, чтобы получить зашифрованный текст.

RSA (Rivest-Shamir-Adleman) - это криптографический алгоритм, используемый для шифрования и подписи данных. RSA основан на математической проблеме факторизации больших целых чисел, которая не имеет известных эффективных алгоритмов решения. Это позволяет использовать алгоритм для создания надежных криптографических ключей. Основная идея алгоритма RSA заключается в том, что каждый пользователь имеет два ключа - публичный и приватный. Публичный ключ имеет свободную лицензию, и его можно использовать для шифрования данных перед отправкой их владельцу приватного ключа. Приватный ключ используется только владельцем для расшифровки данных, которые были зашифрованы с помощью соответствующего публичного ключа. [2]

- IDEA и AES. IDEA и AES являются симметричными блочными шифрами, которые используются для шифрования данных. IDEA использует ключ длиной 128 бит, в то время как AES использует ключи длиной 128, 192 или 256 бит. IDEA шифрует данные с помощью 64-битных блоков, аналогично с этим AES шифрует данные с помощью блоков длиной 128 бит. IDEA имеет более сложную структуру, чем AES, что может привести к более высоким требованиям к ресурсам. AES, с другой стороны, обладает более простой структурой и более высокой производительностью.
- IDEA и Blowfish. Blowfish является другим симметричным блочным шифром, который использует ключи длиной до 448 бит, в то время как IDEA использует ключ длиной 128 бит. IDEA обычно считается более безопасным, чем Blowfish, так как Blowfish был разработан в 1993 году и может быть подвержен атакам более высокого уровня.
- IDEA и RSA. RSA является асимметричным шифром, который используется для защиты программного обеспечения и создания цифровых подписей. Он использует открытый и закрытый ключи, в то время как IDEA и другие симметричные шифры используют только один ключ для шифрования и расшифровки данных. RSA также может использоваться для обмена ключами, что обеспечивает дополнительную защиту. Однако

RSA обычно медленнее, чем IDEA, из-за сложной математической структуры асимметричного шифрования. [3, 4]

В целом, IDEA обычно считается одним из наиболее безопасных и эффективных алгоритмов шифрования. AES является более быстрым и производительным, но может быть менее безопасным при использовании коротких ключей. Blowfish и RSA также могут быть эффективными в определенных ситуациях, но могут быть менее безопасными или медленными, чем IDEA.

Оценка эффективности алгоритма IDEA в современных информационных системах.

Исследование эффективности алгоритма IDEA показало, что у этого алгоритма есть свои преимущества и недостатки по сравнению с другими алгоритмами шифрования, такими как AES, Blowfish и RSA.

Основным преимуществом алгоритма IDEA является его высокая степень защиты данных. IDEA использует 128-битный ключ и 64-битный блок данных, что делает его очень надежным и безопасным алгоритмом шифрования, который может быть использован для защиты конфиденциальных данных.

Кроме того, IDEA не подвержен атакам типа "brute-force" и "known-plaintext attack", что делает его еще более надежным.

Недостатками алгоритма IDEA являются его относительно низкая скорость и сложность реализации, что может потребовать больших вычислительных ресурсов. IDEA медленнее, чем некоторые другие алгоритмы шифрования, такие как AES и Blowfish, что может оказаться проблемой в системах с большим объемом данных, которые требуют быстрого шифрования и расшифровки.

Практические рекомендации по использованию алгоритма IDEA в информационных системах.

Исходя из результатов исследования, можно сделать следующие практические рекомендации по использованию алгоритма IDEA в информационных системах:

- Использование IDEA в сочетании с другими алгоритмами шифрования: хотя IDEA обеспечивает достаточный уровень защиты данных, его можно использовать в сочетании с другими алгоритмами шифрования, чтобы увеличить уровень безопасности. Например, можно использовать IDEA для шифрования данных, а затем AES для дополнительного шифрования.
- Использование ключей большой длины: IDEA использует 128-битные ключи, что обеспечивает надежную защиту данных. Однако, для обеспечения более высокого уровня безопасности рекомендуется использовать ключи большей длины, например, 256 бит.
- Использовать случайно сгенерированные ключи, а не ключи, которые могут быть вычислены из другой информации, например, пароля пользователя. Такой подход снижает вероятность успешной атаки на систему.
- Не использовать один и тот же ключ для шифрования разных сообщений. Вместо этого следует генерировать уникальный ключ для каждого сообщения или использовать метод гаммирования (метод шифрования, который использует гамму, то есть случайно сгенерированный ключевой поток битов, для изменения открытого текста и создания шифротекста), при котором каждое сообщение шифруется с использованием уникального ключа, полученного из генератора случайных чисел.
- Использование режимов работы с блочными алгоритмами шифрования: IDEA, как и многие другие алгоритмы шифрования, может использоваться в различных режимах работы, таких как ECB, CBC, CTR и другие. Режимы CBC и CTR обеспечивают дополнительный уровень безопасности и являются более рекомендуемыми для использования в информационных системах.
- Использование алгоритма IDEA в сочетании с аутентификацией сообщений: IDEA обеспечивает только конфиденциальность данных, но не гарантирует их целостность и подлинность. Для обеспечения целостности и подлинности данных рекомендуется использовать алгоритм IDEA в сочетании с аутентификацией сообщений, например, HMAC.
- Регулярное обновление ключей: для обеспечения надежной защиты данных рекомендуется регулярно обновлять ключи шифрования. В зависимости от уровня безопасности, можно использовать различные периоды обновления ключей.
- Использование проверенных и сертифицированных реализаций IDEA: для обеспечения безопасности данных, рекомендуется использовать проверенные и сертифицированные реализации алгоритма IDEA, такие как OpenSSL, Crypto++ и другие.
- Соблюдение принципов информационной безопасности: использовать алгоритм IDEA в сочетании с другими методами защиты информации, например, аутентификацией пользователей, контролем доступа и защитой от вредоносных программ.

Заключение.

В данной научной статье была произведена оценка эффективности алгоритма шифрования IDEA в современных информационных системах. В ходе исследования были рассмотрены основные преимущества и недостатки алгоритма IDEA, а также проведено сравнение с другими алгоритмами шифрования, такими как AES, Blowfish и RSA.

Было показано, что IDEA обладает высоким уровнем безопасности и производительности, что делает его привлекательным для использования в информационных системах, где требуется надежная защита данных.

В сравнении с другими алгоритмами шифрования, IDEA демонстрирует более высокую производительность на небольших объемах данных, но на больших объемах данных AES и Blowfish проявляют себя лучше. RSA же отличается от IDEA и других алгоритмов тем, что используется для шифрования данных, а не для шифрования ключей.

В целом, результаты исследования показывают, что алгоритм IDEA является эффективным и безопасным для использования в информационных системах. Однако, необходимо учитывать зависимость от размера ключа и потенциальную уязвимость при использовании коротких ключей.

Список использованных источников:

- Когаловский М. П. (2003). *Перспективные технологии информационных систем*
- W. Stallings, (2016). *Cryptography and Network Security: Principles and Practice*
- Kocur, D., & Pospisil, J. (2017). *Evaluation of Security and Performance of IDEA*
- Шнайер Б. (2002). *Прикладная криптография*

UDC 004.056.55

STUDY OF THE EFFECTIVENESS OF IDEA ENCRYPTION ALGORITHM IN MODERN INFORMATION SYSTEMS

*Sytsko M.V, student gr.253504, Zhak M.V, student gr. 253504
Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus*

Stroinikova E. D. – Senior Lecturer in Department of Informatics

Annotation. This work is devoted to the study of the effectiveness of the IDEA (International Data Encryption Algorithm) encryption algorithm in modern information systems. The paper reviewed the existing encryption methods and compared the IDEA with alternative algorithms. The analytical evaluation of the performance of the IDEA algorithm on different platforms and under different conditions was also carried out. As a result of the research, it was found that the IDEA is quite effective encryption algorithm, having a high level of security and suitable for use in modern information systems. However, some limitations and shortcomings of the IDEA algorithm have also been identified, which should be taken into account when using it. This work is an important contribution to the development of the theory and practice of cryptography and information security.

Keywords. Data protection, encryption, IDEA, information systems, efficiency, methodology, algorithm comparison, information security, centralized and decentralized systems, client-server architecture, performance, malware, keys

МЕТОДЫ ОТОБРАЖЕНИЯ ЧЕТЫРЕХМЕРНОГО ПРОСТРАНСТВА НА ПРИМЕРЕ ВИДЕОИГР

Новиков В.А., студент гр.253504, Давыдовский Д.В., студент гр.253504

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Стройникова Е. Д. – ст. преп. кафедры информатики

Аннотация. Данная научная работа посвящена методам отображения четырехмерных пространств на примере видеоигр. В работе проводится теоретический обзор, включающий понятие четырехмерного пространства, основные методы отображения четырехмерных пространств, используемых в игровой индустрии. В результате исследования были рассмотрены методы отображения четырехмерных пространств в нескольких видеоиграх. Работа может быть полезна для разработчиков видеоигр и специалистов в области компьютерной графики, интересующихся различными методами отображения четырехмерных пространств.

Ключевые слова. Методы отображения, четырехмерное пространство, видеоигры, компьютерная графика, игровая индустрия.

Введение

Четырехмерное пространство, хотя и не может быть визуализировано в реальном мире, является важной областью математики и физики. В последнее время его применение расширилось и нашло применение в различных областях, включая видеоигры. Однако, отображение четырехмерного пространства в трехмерном мире, таком как в видеоиграх, представляет собой сложную задачу.

Цель данной научной работы - изучить методы отображения четырехмерного пространства на примере видеоигр. Для достижения этой цели, будут рассмотрены основные методы отображения четырехмерных пространств, используемые в игровой индустрии, а также основные принципы и технологии, используемые в видеоиграх.

В ходе исследования будут выбраны определенные видеоигры и методы отображения, проведены тесты и оценены результаты. Данная работа позволит определить наиболее эффективные методы отображения четырехмерного пространства в контексте видеоигр и даст рекомендации для дальнейших исследований в данной области.

Работа может быть полезна для специалистов в области игровой индустрии, а также для ученых, интересующихся визуализацией четырехмерного пространства.

Геометрия евклидовых пространств

В трехмерном пространстве с заданной системой координат точки и векторы характеризуются тремя координатами, в четырехмерном пространстве аналогично – четырьмя, что приведено на рисунке 1.

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}.$$

Рисунок 1 – Четырехмерный вектор.

Четыре стандартных базисных вектора будут выглядеть так, как показано на рисунке 2.

$$\mathbf{e}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}; \mathbf{e}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}; \mathbf{e}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}; \mathbf{e}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

Рисунок 2 – Базисные вектора.

Общий вектор показан на рисунке 3.

$$\mathbf{a} = a_1 \mathbf{e}_1 + a_2 \mathbf{e}_2 + a_3 \mathbf{e}_3 + a_4 \mathbf{e}_4.$$

Рисунок 3 – Общий вектор.

Для выполнения операций сложения и вычитания векторов в любом количестве измерений, необходимо проводить их покомпонентно. Аналогично, для вычисления скалярного произведения 4-векторов используется формула, приведенная на рисунке 4.

$$\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + a_3 b_3 + a_4 b_4.$$

Рисунок 4 – Сложение векторов.

Расчет угла между двумя векторами производится так же, как и для трехмерного пространства, что приведено на рисунке 5.

$$\theta = \arccos \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}.$$

Рисунок 5 – Нахождение угла между векторами.

Норма вектора в любом количестве измерений может быть выражена через квадратный корень из скалярного квадрата этого вектора, так же, как это делается в трехмерном случае. В отличие от трехмерной геометрии, в четырех измерениях не существует прямого аналога векторного произведения. Вместо этого, можно использовать бивектор внешнего произведения для выполнения аналогичных операций.

Визуализация четырехмерных тел в трехмерном пространстве.

Когда речь идет о отображении четырехмерного пространства, существует несколько методов, которые могут быть использованы. Одни из них основаны на визуализации четырехмерных объектов в трехмерном пространстве, а другие на математических алгоритмах, которые позволяют сделать проекцию четырехмерного пространства на более низкую размерность.

Проекция - это способ представления n-мерных фигур на картинном подпространстве через геометрическую идеализацию оптических механизмов. Проекция может быть выполнена на параллельное или центральное подпространство, или на гиперплоскость n-1 при использовании стереографических проекций. Часто используются параллельные проекции, где проекционные лучи параллельны, а также центральные проекции, где лучи исходят из некоторой точки. При проекции четырехмерных фигур, проектирование происходит на трехмерное пространство, которое имеет размерность на единицу меньше, чем само пространство, в котором находится проектируемая фигура. Стереографическая проекция является центральной проекцией, которая отображает n-1-сферу n-мерного шара на гиперплоскость n-1. Пример приведен на рисунке 6.

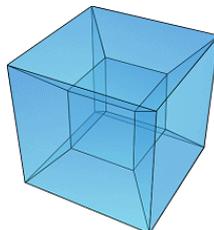


Рисунок 6 – Трехмерная проекция гиперкуба(тессеракта)

Сечение - это изображение фигуры, которая образуется, когда тело пересекается плоскостью, без изображения частей тела, находящихся за этой плоскостью. Аналогично созданию двумерных сечений трехмерных объектов, можно создавать трехмерные сечения четырехмерных тел. Однако, в отличие от двумерных сечений, трехмерные сечения могут иметь более разнообразные формы, поскольку они могут иметь разное количество граней и сторон у каждой грани сечения. Создание трехмерных сечений более сложный процесс, чем создание проекций, поскольку требуется логическое рассмотрение каждого конкретного случая. Пример приведен на рисунке 7.



Рисунок 7 – Сечение гиперкуба и гиперсферы

Развёртка гиперповерхности - это фигура, которая появляется в гиперплоскости (подпространстве), когда точки данной гиперповерхности совмещаются с этой плоскостью таким образом, чтобы длины линий оставались неизменными. Как и трёхмерные многогранники, которые могут быть созданы из бумажных развёрток, многомерные тела могут быть представлены в виде развёрток своих гиперповерхностей. Пример приведен на рисунке 8.

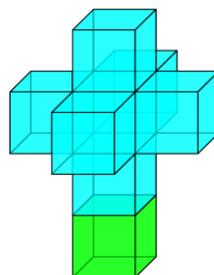


Рисунок 8 – Развертка гиперкуба(тессеракта)

Применение методов визуализации четырехмерных тел в видеоиграх.

1. 4D Toys

Метод отображения сечением применен в игре 4D Toys. Это игра-песочница, которая позволяет исследовать четвертое измерение, а также понять некоторые концепции математики и физики.

Игра была разработана и выпущена в 2016 году студией Osmo. Она представляет собой коллекцию трехмерных объектов, которые можно вращать и изменять в реальном времени. 4D Toys также позволяет перемещаться по четвертому измерению, что позволяет увидеть, как объекты изменяются и взаимодействуют в этом измерении.

Игроки могут экспериментировать с различными формами и структурами, а также изменять физические параметры, такие как скорость и гравитацию, чтобы увидеть, как это влияет на объекты в четвертом измерении. Например, можно создавать петли и дуги, которые выглядят невозможными в трех измерениях, но становятся возможными в четвертом.

На рисунке 9 мы можем заметить два кольца, которые, как может показаться, нельзя разъединить. Но поскольку это не трехмерные фигуры, а сечение четырехмерных фигур в трехмерном пространстве, то при изменении сечения фигуры уже оказываются не замкнутыми друг в друге. Результат на рисунке 10.

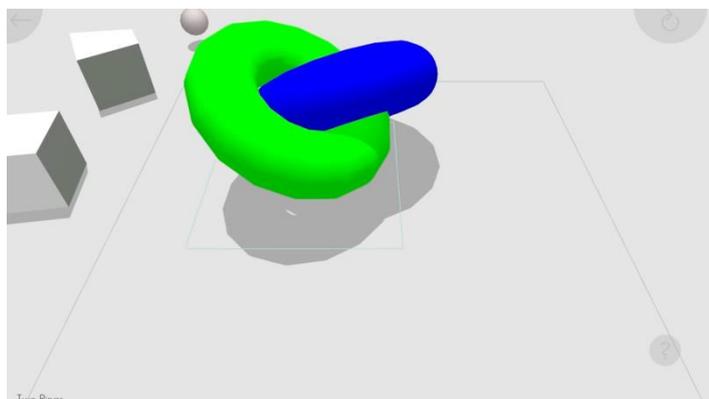


Рисунок 9 – Кольца до изменения сечения.

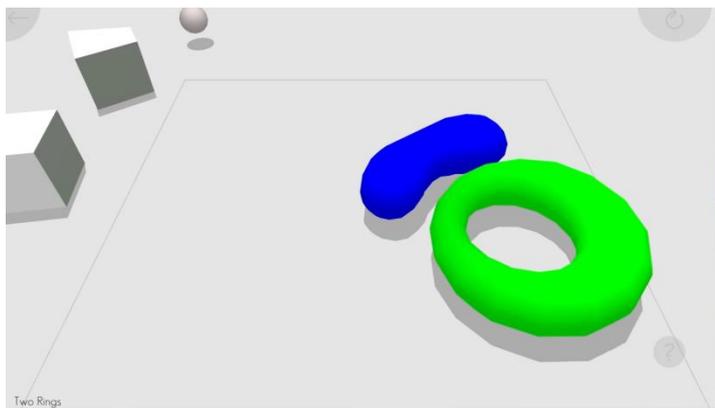


Рисунок 10 – Кольца после изменения сечения.

На рисунке 11 мы видим две фигуры, но на самом деле их больше. При изменении сечения появляются другие фигуры, которые существовали все это время, но заметить мы их смогли только при изменении сечения. Результат на рисунке 12.

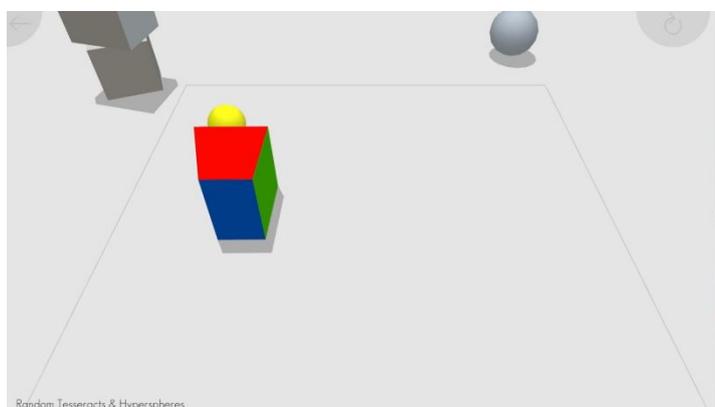


Рисунок 11 – Две фигуры(гиперкуб и гиперсфера).

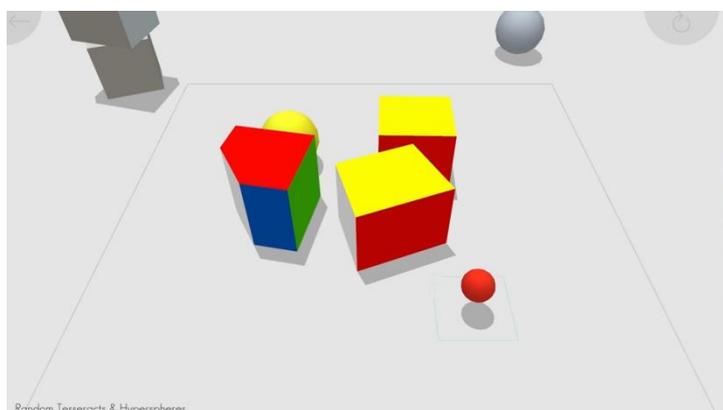


Рисунок 12 – Результат изменения сечения.

4D Toys предлагает интересное и простое в использовании введение в понятие четвертого измерения, и позволяет игрокам более глубоко понять принципы математики и физики, связанные с этой темой.

2. Miegakure

"Miegakure" - это инди-игра, разработанная Марком Брауном, которая позволяет игрокам исследовать мир, содержащий четыре измерения. Игра была выпущена в 2020 году после более чем десяти лет разработки. Механика игры так же как и в 4D Toys основана на методе отображения сечением.

В "Miegakure" игроку предстоит решать различные головоломки и исследовать четырехмерное пространство. Игрок управляет маленьким созданием, которое может перемещаться по трем измерениям, но также может изменять сечение четвертого измерения, чтобы получить доступ к новым областям игрового мира и решить сложные головоломки.

Примером взаимодействия с 4 измерением можно считать уровень, в котором перед нами расположена стена, обойти которую не представляется возможным. Однако не стоит забывать, что игра в четырехмерном пространстве. Если мы изменим сечение, то мы сможем обойти стену, что изображено на рисунках 13, 14, 15.



Рисунок 13 – Стена.



Рисунок 14 – Изменение сечения.



Рисунок 15 – Обход стены после изменения сечения.

3. 4D Maze Game

Игра 4D Maze Game разработана Джоном Макинтошем в 2002 году и предоставляет возможность исследовать 4-х мерный лабиринт. В этой игре используется метод проекции, а также имеется стереоскопический вид для более удобного восприятия лабиринта, что показано на рисунке 16.

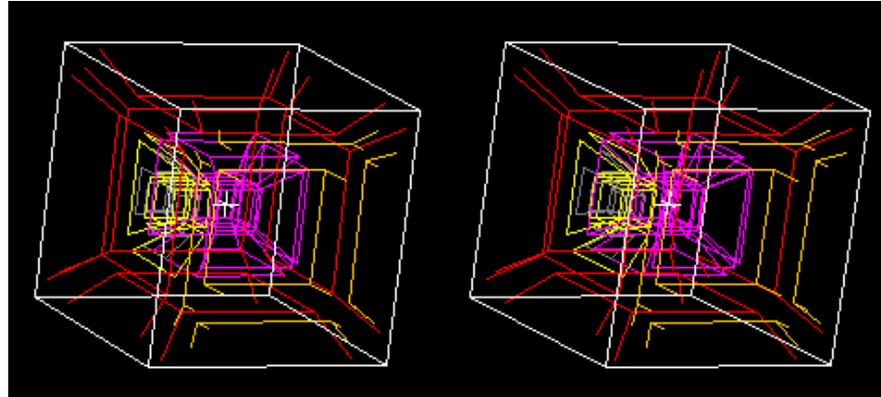


Рисунок 16 – 4D Maze Game.

В 4-х мерном лабиринте у нас есть возможность передвижения по всем четырем осям. Так как проекция четырехмерного лабиринта может быть довольно тяжелой для восприятия, имеется возможность пройти 3-х мерный лабиринт, что показано на рисунке 17.

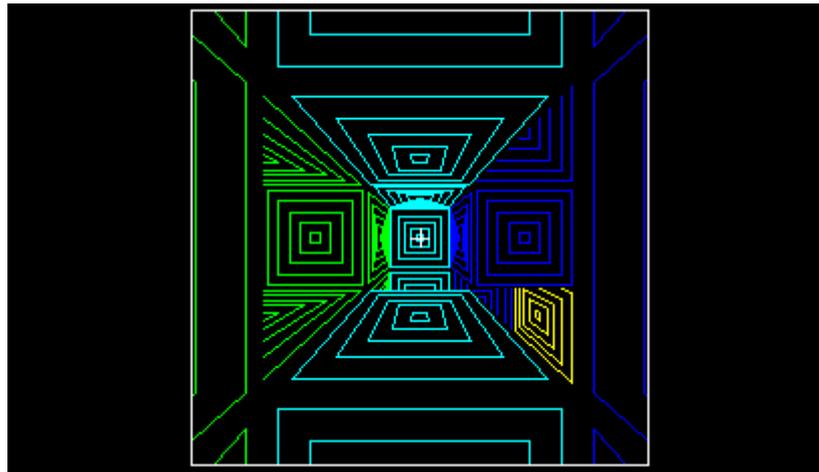


Рисунок 17 – 3-х мерный лабиринт.

Данная игра хорошо справляется с задачей визуализации проекции четырехмерного пространства на трехмерное, что позволяет человеку наглядно понять как работает четвертое измерение.

Заключение.

На данный момент использование 4-х мерной графики в видеоиграх мало распространено. Причин этому несколько. Одна из них это ограничения в мощностях нынешних компьютеров, что не позволяет создавать полноценные игры с открытыми мирами или игры от первого лица, которые бы использовали 4-е измерение. Еще одной причиной может послужить сложность восприятия человеком 4-го измерения, поэтому видеоигры с его использованием не обладают популярностью. Но не исключено, что в будущем мы будем все чаще и чаще сталкиваться с использованием 4-го измерения как в видеоиграх, так и в других сферах предоставления информации.

Однако, тема n-мерных пространств имеет свою популярность в искусстве (архитектура, кинематограф, научная фантастика и т.д.), поэтому весьма вероятно, что эта тема может найти свое применение в видеоиграх в большем масштабе, чем это есть сейчас.

Также стоит отметить, что разработка методов отображения четырехмерных пространств в видеоиграх может иметь большое значение для дальнейшего развития технологий. Такие методы могут быть использованы для создания новых форм виртуальной реальности, где игроки могут полностью погрузиться в мир игры и взаимодействовать с ним на более высоком уровне.

Кроме того, использование четырехмерных пространств может помочь улучшить и расширить интерактивные возможности игр. Например, игроки могут использовать временные перемещение в четвертый измерение, чтобы изменить игровой процесс или создать новые стратегии.

Таким образом, разработка методов отображения четырехмерных пространств в видеоиграх может иметь долгосрочное значение для игровой индустрии и технологий в целом.

Список использованных источников:

1. Владимиров Ю. С. *Пространство-время: явные и скрытые размерности*. — Изд. 2-е, перераб. и доп. — М.: Книжный дом «ЛИБРОКОМ», 2010. — 208 с. — (Науку — всем! Шедевры научно-популярной литературы). — ISBN 978-5-397-01072-6.

2. Ибаньес, Рауль. *Четвёртое измерение. Является ли наш мир тенью другой Вселенной?*. — М.: Де Агостини, 2014. — 160 с. — (Мир математики: в 45 томах, том 6). — ISBN 978-5-9774-0631-4.

3. Coxeter, H.S.M. (1973) [1948]. *Regular Polytopes* (3rd ed.). New York: Dover.

UDC 004.942

METHODS FOR DISPLAYING A FOUR-DIMENSIONAL SPACE ON THE EXAMPLE OF VIDEO GAMES

Novikov V. A., student gr.253504, Davydovskii D. V., student gr.253504

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Stroinikova E. D. – Art. Prep. of the Department of Informatics

Annotation. This scientific work is devoted to methods for displaying four-dimensional spaces on the example of video games. The paper provides a theoretical review, including the concept of four-dimensional space, the main methods for displaying four-dimensional spaces used in the gaming industry. As a result of the study, methods for displaying four-dimensional spaces in several video games were considered. The work may be useful for video game developers and computer graphics specialists interested in various methods for displaying four-dimensional spaces.

Keywords. Display methods, four-dimensional space, video games, computer graphics, game industry.

УДК 004.934

ИЗВЛЕЧЕНИЕ АКУСТИЧЕСКИХ ПРИЗНАКОВ ИЗ АУДИОСИГНАЛА В СИСТЕМАХ РАСПОЗНАВАНИЯ РЕЧИ

Крейс А.В., магистрант

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Боброва Н.Л. – канд. техн. наук

Аннотация. В настоящее время системы распознавания речи получили распространение в различных сферах жизни человека. Система распознавания речи – это система, позволяющая представить человеческую речь, содержащуюся в аудиосигнале, в форме, подлежащей интерпретации компьютером. В настоящее время существуют различные варианты реализации подобных систем, в которых можно выделить общие черты, например, необходимость выделения акустических признаков из исходного аудиосигнала, которые используются алгоритмом(-ами) для распознавания слов, произнесенных диктором. В данной работе выполняется обзор процесса получения данного набора признаков.

Ключевые слова. Акустический признак, система распознавания речи, медианный фильтр, фильтрация шумов, дискретное преобразование Фурье, окно Хэмминга, Мел-частотные кепстральные коэффициенты, Перцептуальное линейное предсказание, Кодирование с использованием линейного предсказания, фрейм, критические полосы слуха, кривые равной громкости, дискретное косинусное преобразование.

Прежде, чем приступить непосредственно к извлечению акустических признаков из аудиосигнала, его необходимо очистить от присутствующих в нем шумов. Существуют различные виды шумов, которые могут присутствовать в аудиосигнале. Они могут быть классифицированы по различным признакам:

- по виду взаимодействия с сигналом (аддитивный, мультипликативный);
- по наличию стационарности (стационарный, нестационарный);
- по постоянству (постоянный, непостоянный (в т. ч. импульсный));
- по частоте и т. д.

Для устранения шумов, присутствующих в сигнале, используются различные фильтры. Приведем некоторые примеры таких фильтров.

1. Медианный фильтр. Данный фильтр представляет собой пример фильтра с конечной импульсной характеристикой. Принцип работы данного фильтра заключается в применении скользящего окна из нескольких отсчетов входного сигнала, которые сортируются по значению с последующим получением отсчета из середины списка (или среднего значения двух отсчетов в середине списка в случае его четной длины). Полученное значение подается на выход фильтра. Медианный фильтр эффективен при устранении импульсного шума.

2. Низкочастотные / высокочастотные / полосно-пропускающие / полосно-заграждающие фильтры. Они применяются для отсеивания высоко-/низкочастотных составляющих сигнала / любых частотных составляющих вне указанного диапазона / определенного диапазона частот соответственно. Данные фильтры можно использовать для устранения аддитивного шума.

Для устранения искажений в спектре сигнала, вызванных применением преобразования Фурье к ограниченному участку сигнала, может быть использовано окно Хэмминга. Применение данного окна заключается в выполнении операции свертки входного сигнала со специальной оконной функцией. Помимо окна Хэмминга для этой цели применяются окна Ханна, Кайзера и др.

Для извлечения акустических признаков из аудиосигнала используются различные методы. Среди них можно выделить следующие: Мел-частотные кепстральные коэффициенты (Mel-frequency cepstral coefficients или MFCC), Перцептуальное линейное предсказание (Perceptual linear prediction или PLP), Дискретное вейвлет-преобразование (Discrete wavelet transform или DWT), Кодирование с использованием линейного предсказания (Linear predictive coding или LPC) и др. Некоторые из данных методов рассмотрены ниже.

Различные методы извлечения акустических признаков из исходного аудиосигнала подразумевают его дробление на небольшие отрезки. Данные отрезки называются фреймами (frames). В работе [1] проводились исследования влияния размера фрейма и количества кепстральных коэффициентов (о которых речь пойдет ниже) на производительность распознавания речи. Согласно результатам исследования размеры фрейма в диапазоне 16–32 мс не оказывают существенного влияния на показатель WER (Word Error Rate).

Популярным методом получения акустических признаков является MFCC. Человеческое ухо различает разные частотные составляющие сигнала неодинаково. Мел-представление позволяют учесть значимость определенных частот из спектра для человека, а также учесть тембр его голоса

[2]. В источнике [3] отмечается, что мел-частотный анализ представляет частоты речи с позиции психоакустического параметра слуха – высоты тона.

Мел представляет собой единицу высоты звука. Данная величина может быть получена из частоты с использованием следующей формулы:

$$m = 2595 \log_{10} \left(1 + \frac{f}{700} \right), \quad (1)$$

где f – частота звука (в Гц).

Для того, чтобы найти мел-частотные кепстральные коэффициенты, необходимо выполнить несколько шагов. Описание процесса получения данных коэффициентов можно найти в источниках [2, 4]. Первый шаг – применить дискретное преобразование Фурье к исходному звуковому сигналу в рамках фрейма. В источнике [5] указано, что размер таких фреймов обычно равен 10–40 мс, при этом кадры накладываются друг на друга. К полученному спектру применяется набор мел-фильтров. Далее определяется энергия для каждого фрейма. Последний этап – применение дискретного преобразования Фурье или дискретного косинусного преобразования.

Использование мел-частотных кепстральных коэффициентов позволяет сократить количество акустических признаков по сравнению с использованием отсчетов сигнала, или его спектра, или его периодограммы [2].

В настоящее время существуют различные модификации алгоритма MFCC, среди которых можно отметить линейно-частотные кепстральные коэффициенты (Linear-frequency cepstral coefficients или LFCC), метод кепстральных коэффициентов прямоугольного набора фильтров (Rectangular-frequency cepstral coefficients или RFCC), гамматон-частотные кепстральные коэффициенты (Gammaton-frequency cepstral coefficients или GFCC) [6].

Метод LPC подразумевает использование так называемого линейного предсказателя, который позволяет получать значение очередного отсчета сигнала $s(n)$, основываясь на предыдущих значениях отсчетов сигнала. Для этого используется следующая формула [3]:

$$s(n) = \sum_{k=1}^p a_k s(n-k), \quad (2)$$

где a_k – k -ый коэффициент линейного предсказателя;

p – порядок линейного предсказания.

Для данного предсказателя необходимо найти коэффициенты a_k , для чего используются три базовых алгоритма [3]:

- ковариационный;
- автокорреляционный;
- лестничный.

Полученные коэффициенты используются в качестве акустических признаков. Метод LPCC (Linear prediction cepstral coding), который является вариацией алгоритма LPC, подразумевает также применение к полученным коэффициентам дискретного преобразования Фурье или дискретного косинусного преобразования, в результате чего получают коэффициенты кепстра линейного предсказателя [7]. Одним из преимуществ LPCC является снижение влияния канала передачи на параметры речевого сигнала [7].

Метод PLP, разработанный Гинеком Германски (Hynek Hermansky), исключает несущественную информацию, содержащуюся в речи и тем самым улучшает качество ее распознавания [8]. Данный метод учитывает три психоакустических фактора: критические полосы слуха с маскированием, кривую равной громкости, степенное соотношение между громкостью и интенсивностью звука [3].

Критические полосы слуха являются частотными полосами, за пределами которых субъективные ощущения звука сильно изменяются [9]. В разных условиях при различных уровнях шума слышимость звука различается. Понижение уровня слышимости в условиях шума называется маскировкой звука.

Кривые равной громкости представляют собой графическое отображение нелинейности восприятия звука человеком [10]. Другое название – кривые Флетчера-Мэнсона [10]. Эти кривые показывают, какое звуковое давление необходимо создать для одинакового восприятия громкости различных частот [10].

Далее приведена последовательность шагов метода PLP [3]:

- применение к фрейму оконной функции и дискретного преобразования Фурье;
- определение спектра мощности для данного фрейма;
- перевод полученного спектра мощности в барк-шкалу;

- перемножение спектра со спектром мощности кривой маскирования критической полосы;
- сглаживание полученного спектра функцией кривой равной громкости;
- извлечение кубического корня из амплитуды полученного спектра.

После выполнения перечисленных выше шагов осуществляется расчет коэффициентов предсказания (см. метод LPC, описанный выше), на основе которых рассчитываются кепстральные коэффициенты [3].

Источник [8] отдает предпочтение использованию методов MFCC и PLP, а не LPC, так как первые два получены из концепции набора фильтров, находящихся в логарифмическом пространстве, и концепции человеческой слуховой сенсорной системы и, следовательно, демонстрируют более хороший результат [8]. Согласно исследованиям, описанным в источнике [11] метод LPC демонстрирует более хорошие результаты, чем MFCC, в условиях невысокого уровня шума в аудиосигнале.

Таким образом, в данной работе был выполнен сбор сведений, касающихся извлечения акустических признаков из аудиосигнала в системах распознавания речи. Была рассмотрена классификация шумов, которые могут присутствовать в аудиосигнале; фильтрация аудиосигнала с использованием различных фильтров; дробление аудиосигнала на фреймы и применение к ним оконных функций. Были описаны различные методы извлечения акустических признаков из исходного аудиосигнала: MFCC, LPC(C), PLP. Собранные в работе сведения могут быть использованы при разработке систем распознавания речи.

Список использованных источников:

1. Mporas, I. Examining the Influence of Speech Frame Size and Number of Cepstral Coefficients on the Speech Recognition Performance [Electronic resource] / I. Mporas, T. Ganchev, I. Kotinas, N. Fakotakis // ResearchGate. – Режим доступа: https://www.researchgate.net/profile/Todor-Ganchev/publication/239546404_Examining_the_Influence_of_Speech_Frame_Size_and_Number_of_Cepstral_Coefficients_on_the_Speech_Recognition_Performance/links/0c9605322c413b47db000000/Examining-the-Influence-of-Speech-Frame-Size-and-Number-of-Cepstral-Coefficients-on-the-Speech-Recognition-Performance.pdf?origin=publication_detail. – Date of access: 15.03.2023.
2. Алюнов, Д. Ю. Реализация алгоритма обработки и распознавания речи / Д. Ю. Алюнов, Е. С. Сергеев, П. В. Пигачев, А. Н. Мытников // Современные наукоемкие технологии. – 2016. – № 3. – С. 225 – 230.
3. Судьенкова А. В. Обзор методов извлечения акустических признаков речи в задаче распознавания диктора / А. В. Судьенкова // Сборник научных трудов НГТУ. – 2019. – № 3–4. – С. 139–164.
4. Мел-кепстральные коэффициенты (MFCC) и распознавание речи [Электронный ресурс] // Хабр. – Режим доступа: <https://habr.com/ru/post/140828/>. – Дата доступа: 12.02.23.
5. Воробьева, С. А. Выделение границ фонов речевого сигнала с помощью мел-частотных спектральных коэффициентов / А. С. Воробьева // Молодой ученый. – 2017. – № 13. – С. 2–5.
6. Гуртуева, И. А. Аналитический обзор и классификация методов выделения признаков акустического сигнала в речевых системах / И. А. Гуртуева, К. Ч. Бжихатлов // Известия Кабардино-Балкарского научного центра РАН. – 2022. – Вып. 1. – С. 41–58.
7. Зо Хеин Мин. Построение системы распознавания речевых сигналов / Хеин Мин Зо, В. М. Довгаль, В. А. Кудинов // Экономика. Информатика. – 2019. – Т. 46, №2. – С. 367–374.
8. Namrata Dave. Feature extraction methods LPC, PLP and MFCC in speech recognition [Electronic resource] / Dave Namrata // International Journal For Advance Research in Engineering And Technology. – Mode of access: https://www.researchgate.net/profile/Namrata-Dave-2/publication/261914482_Feature_extraction_methods_LPC_PLP_and_MFCC_in_speech_recognition/links/562dce4908ae04c2aeb4aa1b/Feature-extraction-methods-LPC-PLP-and-MFCC-in-speech-recognition.pdf?origin=publication_detail. – Date of access: 12.03.2023.
9. Борискевич, А. А. Анализ частотных и временных свойств слухового аппарата: Метод. указания к лабораторной работе по дисциплинам «Цифровая обработка речи и изображений» и «Защита речевых сообщений и объектов связи от несанкционированного перехвата» для студентов спец. «Сети телекоммуникаций» дневной, вечерней и заочной форм обучения / Сост. А. А. Борискевич, В. К. Конопелько. – Мн.: БГУИР, 2003. – 19 с.: ил.
10. Коваленко, А. Кривые равной громкости [Электронный ресурс] / А. Коваленко // Создание электронной музыки. – Режим доступа: <https://fierymusic.net/teoriya-zvuka/krivye-ravnoy-gromkosti>. – Дата доступа: 17.03.2023.
11. Метод шумочистки речевых сигналов на основе мел-частотных кепстральных коэффициентов с использованием фильтрации Калмана / С. М. Горошко, С. Н. Петров // Известия Гомельского государственного университета имени Ф. Скорины. – 2009. – № 6 (117). – С.103–107.

ACOUSTIC FEATURES EXTRACTION FROM AUDIOSIGNAL IN SPEECH RECOGNITION SYSTEMS

Kreis A.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Bobrova N.L. – PhD in Technical Sciences

Annotation. Nowadays speech recognition systems are widespread in different spheres of humans` life. Speech recognition system refers to a system that allows to represent humans` speech located in an audio signal in the form that can be interpreted by a computer. At the present time different variants of such systems implementation exist. They possess common traits, for instance, the necessity of the acoustic features extraction from the source audio signal that are used by the algorithm(s) for the recognition of words that were pronounced by a speaker. In this work an overview of such feature set extraction process is conducted.

Keywords. Acoustic feature, speech recognition system, median filter, noise filtration, discrete Fourier transform, Hamming window, Mel-frequency cepstral coefficients, Perceptual linear prediction, Linear predictive coding, frame, critical band, equal-loudness-level-contours, discrete cosine transform.

РАВНОВЕСИЕ НЭША

Сугако Т.А

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Примичева З.Н. – канд. физ.-мат. наук

Равновесие Нэша - состояние в теории игр, когда каждый игрок выбирает свою оптимальную стратегию, учитывая стратегии других игроков, и ни у кого нет стимула менять свою стратегию, зная стратегии других. Оно названо в честь Джона Нэша, который разработал теорию некооперативных игр. Равновесие Нэша широко используется в экономике, политологии и других общественных науках для анализа поведения отдельных людей или групп в конкурентных ситуациях. Для проверки существования равновесия Нэша можно использовать алгоритм Лемке-Хоусона.

Концепция решения в теории игр, которая представляет собой набор стратегий для каждого игрока, такой, что ни у одного из игроков нет стимула в одностороннем порядке менять свою стратегию, учитывая стратегии других игроков, называется Равновесием Нэша [1]. Другими словами, равновесие по Нэшу - стабильное состояние игры, в котором ни один игрок не может увеличить свой выигрыш, изменив свою стратегию, при условии, что все остальные игроки сохраняют свои стратегии фиксированными.

Термин назван в честь математика Джона Нэша, который разработал теорию некооперативных игр. В знак признания его значительного вклада в эту область Нэш был удостоен в 1994 году Нобелевской премии памяти в области экономических наук [2]. Равновесие Нэша широко используется в экономике, политологии и других общественных науках для анализа поведения отдельных людей или групп в конкурентных ситуациях, поскольку оно позволяет предсказать результаты стратегических взаимодействий между рациональными агентами (участниками игры). Оно используется для моделирования широкого спектра явлений, таких как рынки олигополии, переговорные игры, системы голосования и международные конфликты. В качестве примера можно привести исторический факт, заключающийся в том, что во время холодной войны США и СССР находились в состоянии равновесия - взаимного гарантированного уничтожения, - когда каждая сторона точно знала, где находятся позиции другой, но при этом не начинала войну.

Равновесие Нэша не всегда является уникальным, может существовать несколько равновесий или оно вообще может отсутствовать, в зависимости от структуры игры и предпочтений игроков. Более того, равновесие по Нэшу не обязательно приводит к социально оптимальному результату, поскольку оно может соответствовать ситуации, когда все игроки находятся в худшем положении, чем они могли бы быть при другом сценарии сотрудничества. Поэтому равновесие Нэша не следует рассматривать как концепцию окончательного решения для каждой игры, а скорее как полезный инструмент для анализа стратегических взаимодействий простым и интуитивно понятным способом.

Дилемма заключенного – классический пример теории игр, была первоначально сформулирована Мерриллом Флудом и Мелвином Дрешером в 1950 году, когда они работали в RAND. Альберт В. Такер позже формализовал игру, структурировав вознаграждения в терминах тюремных сроков, и назвал ее "дилеммой заключенного". В 1993 году Уильям Паундстоун описал игру в своей книге "Дилемма заключенного" [3].

В данной игре участвуют двое заключенных, находящихся в одиночных камерах, лишенных возможности общаться друг с другом. В отсутствие достаточных доказательств для осуждения обоих преступников было принято решение предложить каждому заключенному возможность либо предать другого, дав показания о том, что преступление совершил другой, либо сотрудничать, храня молчание.

При условии, что оба заключенных предадут друг друга, оба получают пять лет заключения. В случае, когда один заключенный предает другого, а другой молчит, предавший получает полное освобождение, а молчащий отбывает десять лет заключения. Если же оба заключенных решают не предавать друг друга, они получают по одному году заключения (таблица 1).

Таблица 1 - Представление дилеммы заключенного

		Заключенный 2	
		Признаться	Молчать
Заключенный 1	Действия		
	Признаться	5,5	0,10
	Молчать	10,0	1,1

Равновесие по Нэшу в данном случае заключается в том, что оба заключенных передают друг друга. Это происходит несмотря на то, что совместное сотрудничество приводит к лучшему исходу. Если один заключенный выбирает сотрудничество, а другой - предательство, результат для первого становится хуже.

Для вычисления равновесия Нэша необходимо смоделировать каждый из возможных сценариев и из этого числа выбрать оптимальную стратегию. В игре для двух человек это предполагает учет возможных стратегий, которые могут выбрать оба игрока. Если ни один из игроков не изменит свою стратегию, зная всю информацию, то равновесие по Нэшу достигнуто.

Один из способов вычисления равновесия Нэша это алгоритм Лемке-Хоусона - итерационный алгоритм для нахождения равновесия Нэша в игре двух игроков [4]. Он был разработан независимо Кеннетом Дж. Эрроу, Жераром Деброй и Лайонелом В. Маккензи в 1950-х годах, но его название происходит от имени математиков Карла Густава Гемпеля и Альберта Уильяма Такера, которые были вдохновлены реализацией алгоритма Джеком У. К. Лемке и Т. С. Хоусоном в 1960 году.

Рассмотрим игру двух игроков со следующими характеристиками.

Игрок 1 имеет n стратегий, обозначаемых $S_1 = \{s_1, s_2, \dots, s_n\}$ и игрок 2 имеет m стратегий, обозначаемых $S_2 = \{t_1, t_2, \dots, t_m\}$. Матрица выплат для игры обозначается A , где A_{ij} представляет собой выплату игроку 1, когда игрок 1 выбирает стратегию s_i , а игрок 2 выбирает стратегию t_j .

Алгоритм Лемке-Хоусона работает путем построения новой игры, называемой "дополнительной игрой", которая эквивалентна исходной игре, но обладает некоторыми полезными свойствами, упрощающими вычисление равновесия Нэша. Дополняющая игра получается путем отрицания матрицы выплат и перемены ролей игроков, то есть, в дополняющей игре есть: игрок 2 с n стратегиями, обозначаемыми $S_2 = \{s_1, s_2, \dots, s_n\}$ и игрок 1 с m стратегиями, обозначаемыми $S_1 = \{t_1, t_2, \dots, t_m\}$. Матрица выплат для комплементарной игры обозначается B , где $B_{ij} = -A_{ji}$ представляет собой выплату игроку 2, когда игрок 2 выбирает стратегию s_i , а игрок 1 выбирает стратегию t_j .

Алгоритм Лемке-Хоусона работает следующим образом.

1. Выбрать начальную смешанную стратегию для каждого игрока. Эти стратегии обозначаются x и y , где x_i - вероятность того, что игрок 1 выберет стратегию s_i , а y_j - вероятность того, что игрок 2 выберет стратегию t_j .
2. Постройте новую игру, называемую "слабой игрой", которая является комбинацией исходной игры и дополнительной игры. В игре есть $2n + 2m$ переменных и $2n + 2m$ ограничений, и она может быть представлена в виде линейной программы.
3. Решите слабую игру, используя симплекс-метод. Решение слабой игры дает пару дополнительных выполнимых решений, обозначаемых (x', y') и (u, v) , где x' и y' - выполнимые стратегии для исходной игры, а u и v - выполнимые стратегии для дополнительной игры.
4. Обновите смешанные стратегии для каждого игрока. Если ни один из игроков не поменял стратегию с предыдущей итерации, то текущие стратегии являются равновесием Нэша. В противном случае вернитесь к шагу 2 и повторите процесс.

Математические обозначения для алгоритма Лемке-Хоусона могут быть довольно сложными и включать линейное программирование, многогранники и теорию игр. Однако основная идея заключается в использовании дополнительной игры для упрощения вычисления равновесия Нэша, а также в итерации между исходной игрой и дополнительной игрой до тех пор, пока не будет найдено равновесие Нэша.

Алгоритм Лемке-Хоусона лежит в основе функционала библиотеки Python для вычисления равновесий в теории игр NashPy. Она была разработана исследовательской группой по теории игр в Оксфордском университете, включая доктора Юпитера Гонсалвеса, доктора Винсента Найта и доктора Оуэна Кэмпбелла. Решение дилеммы заключенного с использованием NashPy приведена на рисунке 1.

```
import numpy as np #подключение необходимых библиотек
import nashpy as nash

A = np.array([[ -5, -5], [0, -10]]) # Матрица стратегии заключенного 1
B = np.array([[ -10, 0], [-1, -1]]) # Матрица стратегии заключенного 2
game1 = nash.Game(A, B)

equilibria = game1.support_enumeration()#вызов функции для подсчета равновесия
for eq in equilibria:
    print(eq)
```

Рисунок 1 - Решение дилеммы заключенного на языке Python

Список использованных источников:

1. *Kreps, D.M., 1989. Nash equilibrium. Game theory, с.167-177.*
2. *Майерсон, Р., 2010. Равновесие по Нэшу и история экономической науки. Вопросы экономики, (6), с.26-43.*
3. *Axelrod, R., 1980. Effective choice in the prisoner's dilemma. Journal of conflict resolution, 24(1), pp.3-25.*
4. *Shapley, L.S., 2009. A note on the Lemke-Howson algorithm. In Pivoting and Extension: In honor of AW Tucker (pp. 175-189). Berlin, Heidelberg: Springer Berlin Heidelberg.*
5. *Knight, V. and Campbell, J., 2018. Nashpy. Journal of Open Source Software, 3(30), p.904.*

СРАВНЕНИЕ МЕТОДА ОПОРНЫХ ВЕКТОРОВ С МЕТОДОМ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ В СТЕГАНОАНАЛИЗЕ

Барановский Г.В., Бекарев С.С., Гулис А.А., Шишов Е.П.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Стройникова Е.Д. – старший препод. каф. информатики

Рассмотрены понятия стеганографии и стеганоанализа. Внимание акцентировано на двух конкретных методах стеганоанализа: методе опорных векторов и методе генеративно-сопоставительных сетей, их преимуществах и недостатках.

Ключевые слова. Стеганография, стеганоанализ, стегоконтейнер, SVM, GNA, LSB, задача классификации, Machine Learning.

Введение

Стеганография — способ передачи и/или хранения информации с учётом сохранения в тайне самого факта такой передачи. В отличие от криптографии, которая скрывает содержимое тайного сообщения, стеганография скрывает сам факт его существования.

Понятие стеганоанализа

Стеганоанализ — раздел стеганографии изучающий способы выявления тайно передаваемой информации в анализируемом информационном объекте. Под тайно передаваемой информацией обычно подразумевается информация, скрытая теми или иными стеганографическими методами. Стеганоанализ имеет широкое применение в современном мире, например, в мультимедийных системах для обеспечения безопасности авторских прав и защиты цифрового контента.

Для обнаружения информации, скрытой с помощью стеганографических методов, разработан ряд методов стеганоанализа. В данной работе будет рассмотрено два метода : метод опорных векторов (*Support Vector Machine*) и метод генеративно-сопоставительных сетей (*Generative Adversarial Network*).

Метод опорных векторов (SVM)

Метод опорных векторов (SVM) — это алгоритм машинного обучения, который может использоваться для классификации, регрессии и других задач машинного обучения. В основе SVM лежит поиск гиперплоскости в n -мерном пространстве (где n — количество признаков), которая наилучшим образом разделяет данные на разные классы.

Рассмотрим бинарную классификацию, когда мы пытаемся разделить данные на два класса, например зеленые точки и синие точки на плоскости. Метод опорных векторов ищет гиперплоскость таким образом, чтобы она была максимально удалена от ближайших точек каждого класса. Эти ближайшие точки называются опорными векторами.

В стеганоанализе SVM используют для построения модели, которая будет разделять стеганографически изменённые файлы от их оригинальных версий. Обычно для обучения модели используются пары "чистых" и "скрытых" файлов. Оригинальный файл считается "чистым", а изменённый файл (в котором скрыта дополнительная информация) считается "скрытым". Затем модель обучается на этих парах файлов с помощью SVM. После обучения модель может использоваться для определения, является ли новый файл стеганографически изменённым или нет. Для этого изображение (аудио или видео) преобразуется в набор признаков, которые используются для классификации моделью SVM. Если файл классифицируется как "скрытый", то это означает, что он содержит скрытую информацию.

Плюсы метода опорных векторов:

1. Метод опорных векторов обладает высокой точностью при обнаружении скрытой информации в изображениях.
2. Позволяет рассматривать различные виды нелинейности, изменяя ядра или их параметры.
3. Максимизирует разделяющую полосу, которая позволяет уменьшить количество ошибок классификации.

Минусы метода опорных векторов:

1. Требует большого количества вычислительных ресурсов и времени для обучения модели на больших объемах данных.
2. Может быть чувствителен к шуму и выбросам в данных, что может снизить точность обнаружения скрытой информации.
3. Не описаны общие методы построения ядер, наиболее подходящих для конкретной задачи в случае линейной неразделимости классов.

Метод генеративно-сопоставительных сетей (GAN)

В стеганографии GAN (Generative Adversarial Network) может использоваться для создания стеганографических изображений, которые могут скрыть информацию внутри изображения.

Генеративная модель GAN состоит из двух компонентов: генератора и дискриминатора. Генератор создаёт фальшивые изображения, а дискриминатор классифицирует, являются ли изображения реальными или фальшивыми. Оба компонента обучаются вместе, и цель состоит в том, чтобы генератор создавал фальшивые изображения, которые дискриминатор не может отличить от реальных.

В стеганографии генератор может использоваться для создания изображений, которые содержат скрытую информацию, например текст. Этот текст может быть добавлен в пиксели изображения, изменяя их значения так, чтобы скрытая информация не была видна невооруженным глазом. Дискриминатор может затем использоваться для проверки, является ли изображение, содержащее скрытую информацию, нормальным изображением или нет.

Плюсы метода GAN:

1. Высокая скрытность: GAN может создавать непредсказуемые паттерны и текстуры, которые могут быть использованы для скрытого хранения данных.
2. Устойчивость к атакам: при использовании GAN в качестве метода стеганографии обнаружение скрытой информации может быть очень трудным.
3. Большой объём данных: GAN может использоваться для создания большого количества скрытых данных и информации, что может быть полезно для хранения больших объёмов конфиденциальной информации.

Минусы метода GAN:

1. Сложность создания: создание хорошо работающей модели GAN для стеганографии может быть очень трудным и требовательным к ресурсам.
2. Сложность обнаружения: в случае обнаружения скрытой информации может быть очень сложно понять, как она была спрятана и как её можно извлечь.

Заключение

Сравнивая методы GAN и SVM, было установлено, что GAN эффективен в генерации новых данных с высокой степенью реалистичности, скрытности и устойчивости к атакам, но менее эффективен в обнаружении скрытых сообщений в существующих данных. SVM более эффективен в обнаружении скрытых сообщений в существующих данных, но требует большого количество вычислительных ресурсов для больших данных, чувствителен к шуму в данных.

Список использованных источников:

1. Christopher M. Bishop. *Pattern recognition and machine learning*, 2006. – с.402 [Электронный ресурс] URL: <https://inlnk.ru/oe57QN>. - Дата доступа(20 апреля 2023)
2. А. В. Бычков, *Алгоритмы синтеза изображений в больших разрешениях на основе генеративно-сопоставительных нейронных сетей*, 2020, 53 с. [Электронный ресурс] URL: <https://inlnk.ru/3ZMLKz>. - Дата доступа(20 апреля 2023)
3. К.В.Воронцев, *Лекции по методу опорных векторов*, с.2-5, 12-13, 2019 [Электронный ресурс] URL: <http://www.ccas.ru/voron/download/SVM.pdf>. - Дата доступа(20 апреля 2023)
4. *The A-Z guide to Support Vector Machine*, 2021 [Электронный ресурс] URL : <https://inlnk.ru/w4y6Y1>. - Дата доступа(20 апреля 2023)
5. *Стегоанализ как поиск скрытых сообщений* [Электронный ресурс] URL : <https://inlnk.ru/meL3xk> . - Дата доступа(20 апреля 2023)

СХЕМА ШНОРРА В КРИПТОГРАФИИ

Колесников П. В., Антихович М. В., Малец В. С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Стройникова Е. Д. – ст. преп. кафедры информатики

Данная работа посвящена одному из наиболее эффективных способов проверки подлинности, а именно способу Шнорра. Схема Шнорра дает возможность проверить подпись сообщения и убедиться в подлинности отправителя. Она также обладает свойствами неделимости, невозможности отказаться от подписи и защиты от повторной передачи сообщения. Проведено раскрытие сути протокола Шнорра, сравнение способа Шнорра с другими методами аутентификации, изучение применения данной схемы в сфере информационной безопасности.

Протокол цифровой подписи Шнорра был разработан криптографом Клаусом Шнорром в 1989 году и является одним из наиболее распространенных протоколов данного типа. Эта схема, являющаяся улучшенной версией схемы Эль-Гамала, используется для обеспечения аутентификации сообщений и цифровых подписей. Цифровые подписи представляют собой математические схемы, которые используются для проверки подлинности и целостности цифровых сообщений. Они позволяют доказать, что сообщение было отправлено конкретным отправителем (подлинность) и не было изменено в процессе передачи (целостность).

В отличие от физической подписи, где отправитель аутентифицирует себя с помощью своего уникального почерка с определенным намерением, цифровая подпись использует математические алгоритмы. Схема Шнорра является одной из самых простых, эффективных и создает короткие подписи. Она также используется для реализации "Proof Of Knowledge", интерактивного доказательства, в котором проверяющий "убеждает" проверяемого в том, что он знает нечто "X". В случае с доказательством подписи, это означает, что верификатор должен быть убежден в том, что он общается с пользователем, который знает закрытый ключ, соответствующий открытому ключу.

Пара закрытый-открытый ключ (или асимметричный ключ) - это метод криптографии, где для шифрования и расшифровки информации используются два разных ключа: открытый и закрытый.

Открытый ключ (public key) – это криптографический ключ, который может быть раскрыт всем. Этот ключ используется для шифрования сообщений и файлов, которые будут отправлены конкретному получателю. Получатель использует свой закрытый ключ, чтобы расшифровать сообщение, таким образом, только получатель может прочитать сообщение, даже если кто-то перехватывает сообщение в процессе передачи.

Закрытый ключ (private key) – это секретный ключ, который используется для расшифровки сообщений, зашифрованных открытым ключом. Закрытый ключ не должен быть раскрыт никому, за исключением владельца. Владелец использует закрытый ключ для расшифровки сообщений, которые были зашифрованы его открытым ключом.

Таким образом, открытый ключ может быть свободно распространен и использован для шифрования данных, но не может быть использован для расшифровки. Закрытый ключ, с другой стороны, известен только владельцу ключа и используется для расшифровки данных, зашифрованных с помощью открытого ключа.

Пары закрытый-открытый ключ являются основой большинства криптографической безопасности, от безопасного просмотра веб-страниц до банковских операций и криптовалют. Пары закрытый-открытый ключ являются асимметричными, то есть по одному ключу можно вычислить другой, но не наоборот. Это позволяет кому-то делиться открытым ключом публично, но быть уверенным, что никто не сможет вычислить закрытый ключ (который хранится в тайне и безопасности).

Пары асимметричных ключей используются в двух основных приложениях:

- Аутентификация, где вы доказываете, что знаете закрытый ключ.
- Шифрования, где сообщения могут быть закодированы, и только человек, обладающий закрытым ключом, может расшифровать и прочитать сообщение.

Безопасность схемы Шнорра основана на неразрешимости некоторых проблем с дискретным логарифмом. Она использует криптографические хеш-функции для генерации уникальных идентификаторов сообщений, что позволяет обеспечить целостность данных и защитить их от подделки. Хеш-функция принимает на вход сообщение произвольной длины и возвращает фиксированный хеш-код конкретной длины. В качестве примера рассмотрим процесс создания цифровой подписи. Для создания цифровой подписи отправитель должен выполнить следующие шаги:

1. Выбрать простое число p и генератор g , такие, что g является первообразным корнем по модулю p .

2. Выбрать случайное число k из интервала $[1, p-1]$.
3. Вычислить $R = g^k \pmod{p}$.
4. Вычислить хеш-значение h от сообщения m .
5. Вычислить $e = h(R||m)$, где $||$ означает конкатенацию R и m .
6. Вычислить $s = k - x \cdot e \pmod{p-1}$, где x – это секретный ключ отправителя.
7. Цифровая подпись будет представлена парой (R, s) .

Для проверки подписи получатель должен выполнить следующие шаги:

1. Вычислить хеш-значение h от сообщения m .

Получатель должен использовать ту же хеш-функцию, которую использовал отправитель при создании подписи. Хеш-значение h является уникальным идентификатором сообщения и используется для проверки подписи.

2. Вычислить $v = g^s \cdot y^e \pmod{p}$, где y – это открытый ключ отправителя.

Получатель должен использовать открытый ключ отправителя, чтобы вычислить значение v . Значение v – это результат комбинации двух частей цифровой подписи: R и s . Часть R была выбрана отправителем на этапе создания подписи, а часть s была вычислена отправителем при помощи его секретного ключа. Получатель вычисляет v с помощью открытого ключа отправителя, которым он располагает, и значений R и s , которые были включены в подпись.

3. Если $R = v$, то подпись считается действительной.

Получатель сравнивает значение R из подписи с вычисленным значением v . Если они равны, то подпись считается действительной, и получатель может быть уверен, что сообщение не было изменено и что оно было отправлено именно тем отправителем, чья цифровая подпись была получена.

Чтобы наглядно продемонстрировать схему Шнора, авторы разработали консольное приложение на языке программирования C++ (в Интегрированной среде разработки (IDE) Microsoft Visual Studio), которая иллюстрирует алгоритм и результат работы схемы Шнора, показанные на рисунке 1.

```

Консоль отладки Microsoft Visual Studio
Создание подписи
Шаг 1: выбор простого числа p и генератора g
Простое число p = 2147483647
Генератор g = 2
Шаг 2: выбор случайного числа k
Случайное число k из интервала [1, p-1]: 42
Шаг 3: вычисление R = g^k mod p
Значение R = 2048
Шаг 4: вычисление хеш-значения h от сообщения m: Hello, World!
Хеш-значение h: 7993990320990026836
Шаг 5: вычисление e = h(R || m), где || обозначает конкатенацию
e: 2445625235840700430
Шаг 6: вычисление s = k - x * e mod (p-1), где x - секретный ключ отправителя
Секретный ключ отправителя: 12345
s: 741966690
Шаг 7: представление цифровой подписи в виде пары (R, s)
std::make_pair(R, s)

Проверка подписи
Шаг 1: вычисление хеш - значения h от сообщения m
7993990320990026836
Шаг 2: вычисление v = g^s * y^e mod p, где y - открытый ключ отправителя
Открытый ключ отправителя: 123456
v: -2
Шаг 3: Сравнение R и v
R != v
Подпись недействительна
    
```

Рисунок 1 -- Результат работы схемы Шнора на языке программирования C++

Алгоритм разработанного консольного приложения доступен по ссылке (может быть использовано студентами в учебных целях):

- <https://github.com/pupajupa/Shnorr-Diagram/tree/main/ShnorrDiagram>.

Таким образом, для проверки цифровой подписи схемы Шнора, получатель должен выполнить три шага: вычислить хеш-значение сообщения, вычислить значение v и сравнить его с

R. Если значения совпадают, то подпись считается действительной. Это обеспечивает целостность и подлинность сообщения и защищает от подделки или изменения данных во время передачи.

Схема Шнорра обладает рядом преимуществ по сравнению с другими протоколами цифровой подписи. Она обеспечивает высокую стойкость к атакам и предотвращает возможность подмены сообщения. Схема Шнорра использует более простой алгоритм, чем ECDSA (Elliptic Curve Digital Signature Algorithm). В ECDSA требуется генерация точек на эллиптической кривой, что требует больше вычислительных ресурсов. Схема Шнорра использует короткие цифровые подписи, что делает ее эффективной и удобной для передачи в цифровых подписях. В схеме Эль-Гамала и RSA размер подписи значительно больше. Схема Шнорра более безопасна, чем схема Эль-Гамала и RSA, так как использует случайные числа для генерации подписи, что усложняет возможность атаки методом подбора. Также, при использовании одинаковых длин ключей Схема Шнорра более безопасна, чем ECDSA. Это связано с тем, что алгоритм Схемы Шнорра основан на проблеме дискретного логарифма, которая считается более сложной, чем проблема дискретного логарифма на эллиптических кривых, используемая в ECDSA. Также, в схеме Шнорра используется хэширование сообщения, что защищает от атак с изменением сообщения. В схеме Шнорра используются операции с целыми числами, которые легче вычислять, чем операции с большими простыми числами, которые используются в схемах Эль-Гамала и RSA. Схема Шнорра обладает более простой структурой и удобнее для использования, поскольку требует меньшего количества вычислений и меньшего объема ключевых материалов. Таким образом, схема Шнорра является более эффективной и безопасной для использования в цифровых подписях, чем схемы Эль-Гамала и RSA.

Однако у схемы Шнорра есть и недостатки. При использовании одного и того же случайного числа k для нескольких подписей может возникнуть утечка секретного ключа.

Нет защиты от атак на основе квантовых вычислений. В настоящее время схема Шнорра не защищена от атак на основе квантовых вычислений. Если квантовые компьютеры станут доступными в будущем, то это может привести к компрометации подписи, созданной с использованием схемы Шнорра.

Схема Шнорра является одним из наиболее распространенных протоколов цифровой подписи в криптографии. Она обладает высокой стойкостью к атакам и обеспечивает надежную аутентификацию сообщений. Несмотря на некоторые недостатки, она остается одним из самых эффективных способов защиты информации.

Схема Шнорра может использоваться не только для создания цифровых подписей, но и для протоколов аутентификации. Например, она может использоваться для аутентификации пользователей в системах безопасности или для проверки целостности данных в распределенных сетях. Кроме того, схема Шнорра может быть расширена для поддержки мультиподписей, когда несколько участников могут создавать подпись от имени группы. Это может быть полезно, например, для создания мультиподписей в блокчейн-системах.

Одним из интересных свойств схемы Шнорра является ее невосприимчивость к атакам типа "человек посередине" (man-in-the-middle attack). Это связано с тем, что подпись создается на основе хеш-значения сообщения, которое не может быть изменено без изменения самого сообщения.

Наконец, схема Шнорра может быть использована в сочетании с другими криптографическими протоколами, например, для создания безопасных каналов связи или для обеспечения конфиденциальности данных.

Таким образом, схема Шнорра является важным инструментом в области криптографии и находит широкое применение в различных системах безопасности и защиты информации.

Схема Шнорра является одной из наиболее эффективных и теоретически обоснованных схем ЭЦП. На ее основе построен стандарт Республики Беларусь СТБ 1176.2-99, южнокорейские стандарты KCDSA и EC-KCDSA.

Что можно предположить о развитии схемы шнорра в будущем.

С точки зрения криптографии, схема Шнорра уже считается весьма надежным протоколом цифровой подписи. Однако, как и любая другая технология, она может стать устаревшей и уязвимой к новым методам атак.

Возможно, будущее развитие схемы Шнорра будет связано с улучшением ее скорости и эффективности, а также с адаптацией к квантовым вычислениям. Схема Шнорра основана на вычислительной сложности задачи дискретного логарифмирования, которая является одной из задач, которые квантовые компьютеры могут решать гораздо быстрее, чем классические компьютеры. Это означает, что в квантовом мире схема Шнорра может быть легко взломана, если не будут приняты соответствующие меры.

Одним из подходов для адаптации схемы Шнорра к квантовым вычислениям является использование квантовых криптографических протоколов в качестве замены классической схеме.

Например, можно использовать квантовые подписи, такие как подписи на основе определенных квантовых протоколов, например, квантовых ключей.

Другим подходом является использование техник, которые позволяют защитить схему Шнора от атак квантовых компьютеров. Один из таких подходов - это использование хеш-функций, которые являются устойчивыми к атакам квантовых компьютеров. В этом случае можно использовать хеш-функции на основе квантовых состояний, такие как квантовые хеши, которые используют квантовые свойства для создания устойчивых к атакам квантовых компьютеров хеш-функций.

Также можно использовать кодирование сообщений в квантовом состоянии, которое обеспечивает защиту от перехвата и изменения сообщения. Это может быть достигнуто, например, с помощью квантовых каналов связи.

Таким образом, схема Шнора может быть адаптирована к квантовым вычислениям, но для этого необходимо принять соответствующие меры для защиты от атак квантовых компьютеров.

Кроме того, существуют и другие протоколы цифровой подписи, которые могут стать более популярными в будущем, например протоколы на основе эллиптических кривых или линейных кодов.

Также возможно, что дальнейший рост схемы Шнора будет связан с ее использованием в блокчейн-технологиях. Схема Шнора уже была выбрана в качестве стандарта цифровой подписи для биткоина, и ее использование может распространиться и на другие криптовалюты и блокчейн-проекты.

Однако, как и в любой области технологий, будущее развитие схемы Шнора будет зависеть от различных факторов, таких как развитие вычислительных возможностей, научных открытий, а также регулятивных и юридических аспектов.

Список использованных источников:

1. *Geeks for Geeks, Цифровая подпись Шнора [Электронный ресурс]. – Режим доступа: <https://www.geeksforgeeks.org/schnorr-digital-signature/>.*
2. *Википедия, Схема Шнора [Электронный ресурс]. – Режим доступа: https://en.wikipedia.org/wiki/Schnorr_signature.*
3. *Хабр, Схема Шнора и её роль в Биткоине [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/articles/534306/>.*
4. *Tari Labs University, Основы подписи Шнора [Электронный ресурс]. – Режим доступа: <https://tlu.tarilabs.com/cryptography/introduction-schnorr-signatures#basics-of-schnorr-signatures>.*

АНАЛИТИЧЕСКИЕ РЕШЕНИЯ ПО УГЛУБЛЕННОМУ АНАЛИЗУ ГРАФОВЫХ БД

Зорко П. А., Кулевич А. О.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Пилецкий И. И. – канд. физ.-мат. наук, доцент

Приводится описание аналитических решений по углубленному анализу графической БД, с целью глубокого анализа данных Web-сайтов в некоторой научной области. Описываются принятые решения по применению ML, демонстрируются, результаты работы компонента получение данных с web-сайта.

Графовые технологии позволяют преобразовать представление Web-сайта в графовую БД со свойствами. В то же время, графовая БД со свойствами может быть использована для анализа свойств тематического сайта с помощью **графа знаний**.

На сайте Medium [1] имеется множество статей и публикаций на различную тематику. Данные этого сайта используются для демонстрации аналитических решений по углубленному анализу графовых БД. С помощью средств графовой БД Neo4j [2] встроенных плагинов APOC и Neosemantics, сериализуются данные статей в RDF и получается тематическая графовая БД. Общее представление полученной БД показано на рисунке 1, левая часть.

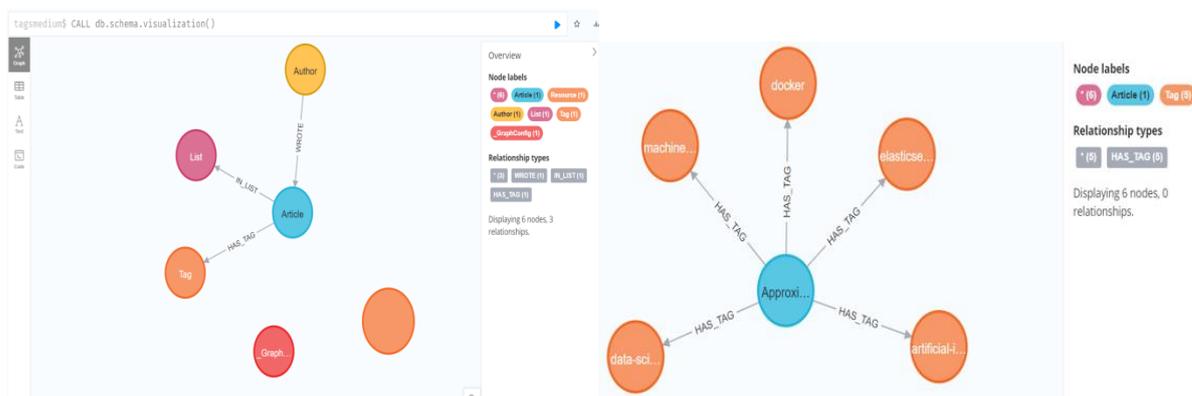


Рисунок 1 – Графовая базы данных и граф знаний

Из полученной базы данных можно получать различные **графы знаний**. К примеру, с помощью следующего запроса **MATCH** (ar:Article)-[:HAS_TAG]->(t:Tag)

WHERE ar.title = "Approximate Nearest Neighbors on Elastic Search with Docker" **RETURN** ar, t можно получить граф знаний, состоящий из статьи и ключевых слов, которые относятся к этой статье. Результат запроса представлен на рисунке 1, правая часть.

Проведем анализ полученного графа знаний. Анализировать данные будем с помощью комбинации возможностей языка программирования Python и Cypher [3].

Посчитаем **embedding узлов в графе**. Сделать это нам позволяют две функции Node2Vec [4] и GraphSAGE [5]. Будем использовать алгоритм GraphSAGE, т.к. вместо обучения отдельных включений для каждого узла алгоритм изучает функцию, которая генерирует включения путем выборки и агрегирования признаков из локальной окрестности узла, что позволяет нам не пересчитывать включения для всего графа при добавлении нового узла, а лишь применить функцию к новому узлу.

Чтобы упростить процесс обучения модели используем алгоритм моночастичной проекции. Идея проекции состоит в том, чтобы взять граф с двумя типами узлов и вывести из него граф с одним типом узлов. Библиотека Neo4j предоставляет алгоритм построения моночастичной проекции с помощью алгоритма Node Similarity [6] из библиотеки GDS.

Сходство узлов вычисляет парные сходства на основе либо показателя Жаккарда, также известного как оценка сходства Жаккарда, либо коэффициента перекрытия, также известного как коэффициент Шимкевича–Симпсона.

После применения алгоритма Node Similarity мы получим граф имеющий следующее представление (рисунок 2):

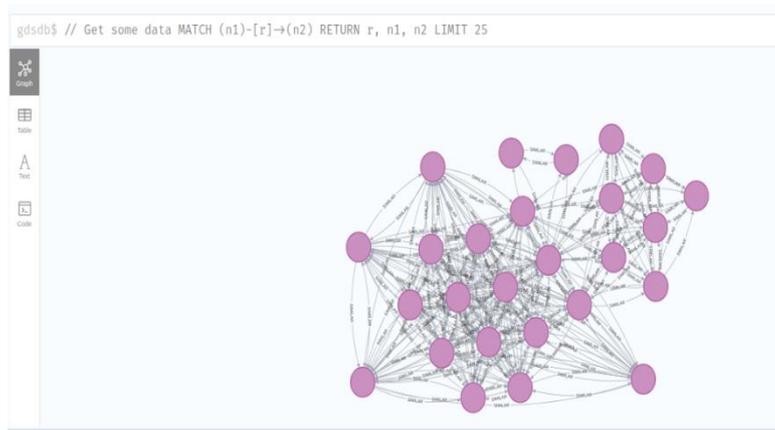


Рисунок 2 – Граф знаний после применения алгоритма Node Similarity

Обучим нашу модель GraphSAGE на выбранном подграфе и перейдем к **ML алгоритму классификации**.

С помощью разработанной нами функции, которая принимает на вход данные и столбцы и применяет к ним алгоритм классификации с несколькими метками, а затем возвращает наиболее эффективно построенную модель, определим теги для статей, число тегов у которых равняется нулю.

После запуска алгоритма получим следующий результат: Introduction to Data Mesh adoption in Adidas – motivation and takeaways → **[data]**; 3 Things to Do When You Feel Ruled by Time → **[productivity]**; A Data Science project start to finish -> **[coding, programming, python, python3, software-development]**; Time series anomaly detection – in the era of deep learning -> **[data-science, machine-learning]**; How to Optimize Your Apache Spark Application with Partitions -> **[spark]**; Rule Execution with SHACL -> **[knowledge-graph]**.

Проанализировав полученные данные, можно заметить, что теги действительно соответствуют тематике статей. Разработанный алгоритм можно использовать в дальнейшем для предсказания тематик, выделения ключевых слов и формирования списков с похожими статьями.

Для подсчета популярности статей будем использовать **ML алгоритм PageRank**:

```
CALL gds.pageRank.write('ArticlesGraph', {maxIterations: 20, dampingFactor: 0.85, writeProperty: 'pagerank'}) YIELD nodePropertiesWritten, ranIterations
```

Гистограмма популярности статей на основе вычисленного PageRank представлена на рисунке 3.

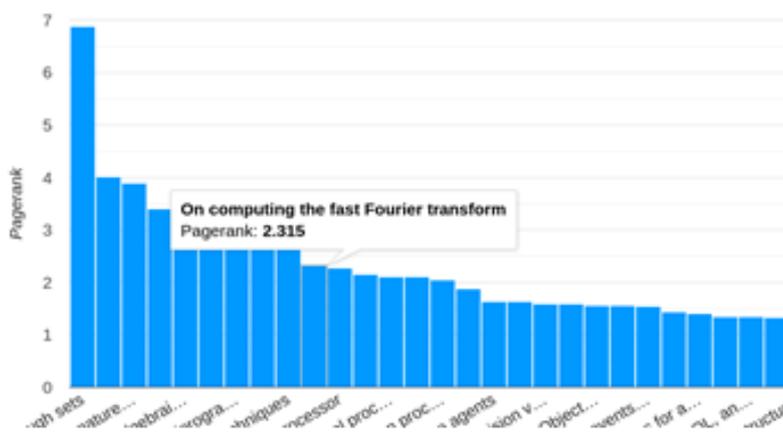


Рисунок 3 – Гистограмма популярности статей на основе вычисленного PageRank

Результатом научной работы является комплексная методологии и ее применение для глубокого анализа данных Web-сайтов.

Список использованных источников:

1. Medium [Электронный ресурс] / Режим доступа: <https://medium.com> Дата доступа: 10.03.23.
2. Neo4j [Электронный ресурс] / Режим доступа: <https://neo4j.com/labs/neosemantics/> Дата доступа: 10.03.23
3. Pyri [Электронный ресурс] / Режим доступа: <https://pyri.org/project/graphdatascience/> Дата доступа: 12.03.23

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

4. Neo4j. Node2Vec [Электронный ресурс] / Режим доступа: <https://neo4j.com/docs/graph-data-science/current/machine-learning/node-embeddings/node2vec/> / Дата доступа: 18.03.23

5. Neo4j. GraphSAGE [Электронный ресурс] / Режим доступа: <https://neo4j.com/docs/graph-data-science/current/machine-learning/node-embeddings/graph-sage/> / Дата доступа: 18.03.23

6. Neo4j. Node Similarity [Электронный ресурс] / Режим доступа: <https://neo4j.com/docs/graph-data-science/current/algorithms/node-similarity/> / Дата доступа: 19.03.23

ИНТЕРНЕТ ВЕЩЕЙ. МЕТЕОСТАНЦИЯ НА БАЗЕ ARDUINO

Богомаз К.С., студент гр. 953503, Юшкевич М.А., студент гр. 953503

Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь

Волорова Н.А. - канд. тех. наук, доцент

Технический прогресс сильно влияет на все аспекты нашей жизни, определяет развитие материальной, бытовой и духовной сфер, меняет систему коммуникаций и информации, воздействует на общественную и личную жизнь людей, коренным образом преобразуя их образ жизни. Была разработана умная гибко настраиваемая метеостанция и её программное обеспечение для профессионального и домашнего использования.

Основная часть. Целью данной научной работы является разработка, описание и документация архитектуры IoT, обеспечивающее безопасности передачи данных, разработка удобного и практичного пользовательского интерфейса и реализация на их основе “умного” метеорологического модуля.

Сейчас наша жизнь насыщена различными устройствами, позволяющими облегчать её. Эти устройства помогают нам с повседневными вещами: уборкой дома, готовкой еды, общение с родными и т.д.. Сейчас, казалось бы, мы не можем представить свою жизнь без мобильных телефонов и Интернета.

Техника занимает важнейшее место в современном обществе. Она воздействует на все стороны нашей жизни, в значительной мере определяя развитие материальной, бытовой и духовной сфер, радикально изменяя ее систему коммуникаций и информации, воздействуя на общественную и личную жизни людей, в корне преобразуя их образ жизни.

Интернет вещей — это концепция сети передачи данных между устройствами. Внутри IoT люди могут общаться с «вещами», а «вещи» — общаться между собой.

Список использованных источников:

1. AWS Documentation [Электронный ресурс]. - Режим доступа : <https://docs.aws.amazon.com/>
2. Arduino Documentation [Электронный ресурс]. - 2022.- Режим доступа : <https://docs.arduino.cc/>

ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ПОИСКА ОДИНАКОВЫХ ПОДГРАФОВ В ГРАФЕ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА

Бурко Л.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Иванюк А.А. – профессор кафедры информатики

Представлено программное средство для нахождения одинаковых подграфов в ориентированном графе с проверкой значений в вершинах. Предлагаемый алгоритм использует эффективный подход, базируемый на хешировании в процессе сопоставления вершин и ребер между ними, который позволяет быстро находить и сравнивать идентичные подграфы в больших графах. Приведены результаты экспериментов с реальными графическими данными, демонстрирующие эффективность и точность предложенного алгоритма.

Сумматоры и умножители являются основными арифметическими блоками во многих цифровых системах. Их оптимальное использование может существенно повысить эффективность проектирования цифровых схем. Поэтому вопрос переиспользования ресурсов является актуальной задачей.

Классические алгоритмы, такие как алгоритм Ульманна [1], VF2, Subgraph Isomorphism направлены на решение задачи поиска изоморфных подграфов, где входными данными являются два неориентированных графа. Поставлена задача – отыскать в графе вычислительного процесса повторяющиеся комбинации, состоящие из одинаковых блоков. Для ее реализации требуется найти максимальное количество различных подграфов с одинаковыми значениями в вершинах для дальнейшего объединения их в блочные структуры.

Однако, чтобы учитывать значения в вершинах, необходимо изменить алгоритмы проверки графов на изоморфизм. Например, к проверке наличия изоморфизма между вершинами двух графов нужно добавить проверку равенство значений в этих вершинах. Для проверки структуры графов на изоморфизм возьмем существующий алгоритм VF2[2] и модифицируем его.

В общем случае, поиск одинаковых подграфов с учетом значений в вершинах является вычислительно сложной задачей, и время ее выполнения может зависеть от размера графа и количества подграфов, которые необходимо найти [3].

В графе вычислительного процесса существует два вида вершин. Вершины, хранящие операнды – обозначены на рисунке 1 прямоугольниками и знаки операций – обозначены кругами [4]. Для поиска общей структуры необходимы только вершины с знаками операций.

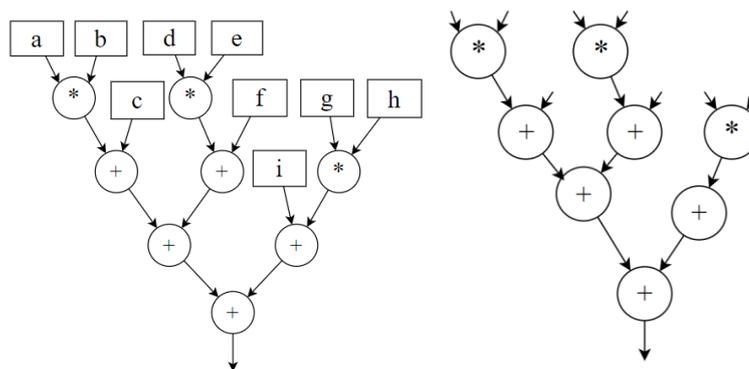


Рисунок 1 – Пример графа вычислительного процесса на 8 вершин (слева) и получившийся, из него граф G (справа).

Для описания графа будут использованы обозначения $M = \text{len}(V(G))$ и $N = \text{len}(E(G))$ для описания вершин и ребер графа G , где $V(G)$ – множество вершин заданного графа, M – количество вершин, $E(G)$ – множество ребер заданного графа, N – количество ребер. Каждой вершине графа присуще свойство (value) "sign", которое хранит знак операции в данной вершине. Паттерном будет называться такой подграф G' , который выполняет условия $G' \subseteq G$, $V(G') \subseteq V(G)$, $E(G') \subseteq E(G)$, $\text{values}(G) = \text{values}(G')$ и встречается в графе G более двух раз.

Чтобы найти абсолютно одинаковые подграфы графа G являющиеся паттернами, нужно разбить граф на компоненты связности по имеющимся ребрам. Всего существует $2^M - 1$ вариантов графов на вершинах M на множествах $V(G)$ и $E(G)$. Перебирать все $2^M - 1$ вариантов не оптимально, поэтому нужно выделить только те варианты, которые могут подходить для реализации задачи. Выделим некоторые условия:

1. Отсутствие изолированных вершин и наличие в подграфе более одного ребра.
2. Компонент связности должно быть больше одной.
3. Количество вершин в максимальной компоненте связности должно быть меньше $M/2$.
4. Количество вершин в компонентах связности должно быть одинаковым.
5. Значения в вершинах между компонентами связности должно совпадать.

Графы получившиеся после прошлого шага нужно проверить на изоморфизм. Для каждого графа вычисляется хеш для меток вершин и ребер. Если хеши равны, то графы - изоморфны. При дальнейшем переборе всех подходящих вариантов выбираются те, где количество компонент связности больше.

Для исследования результатов сгенерируем 3 графа с количеством вершин 8, 13 и 22 соответственно. Значения вершин данных графов может быть + или *. Граф с 8 вершинами представлен на рисунке 1, с 13 – на рисунке 2. Изменение количества подходящих графов в зависимости от добавления условий представлено в таблице 1.

Таблица 1 – изменения в количестве графов.

	Всего вариантов	Условие 1	Условие 2	Условие 3	Условие 4	Условие 5
Граф 1	255	120	87	75	27	8
Граф 2	8191	4083	3847	3478	420	151
Граф 3	4194303	2097130	2088129	2034983	20728	6601

Для просмотра результатов рассмотрим более подробно граф 2. При наличии нескольких паттернов объединяются между собой те, которые не будут пересекаться друг с другом, а именно паттерны [$*$, $*$] и [$*$, $+$, $+$]. Тем самым вместо использования 6 умножителей можно использовать трижды блок из двух. И конструкцию из двух сумматоров и умножителя так же можно использовать дважды.

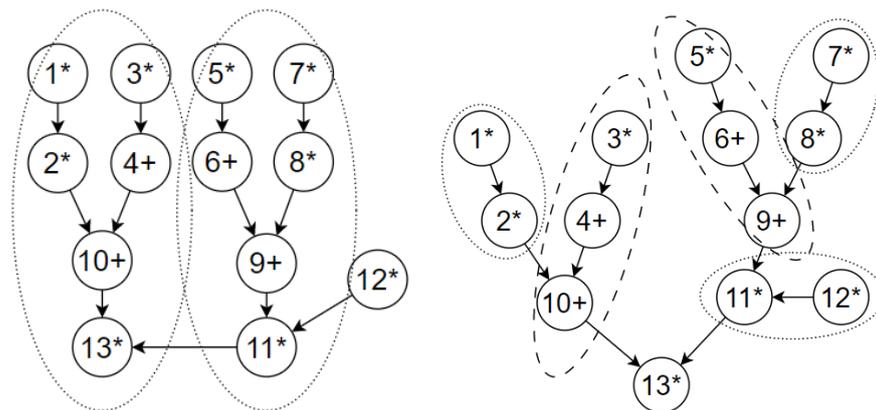


Рисунок 2 – Граф, содержащий 13 вершин, значения вершин $\in \{+,*\}$. Всего предложено 22 варианта замены. Варианты замены – 2 паттерна вида [$*$, $*$, $*$, $+$, $+$, $*$] или 3 паттерна вида [$*$, $*$] и 2 паттерна [$*$, $+$, $+$].

В графе на 22 вершины программным средством был найден 91 паттерн. Максимальное количество вершин в паттерне – 9, таким образом можно заменить 2 блока. Максимальное количество замен – 6, в данном случае можно заменить 2 вершины.

Таблица 2 – количество затрачиваемых ресурсов.

	Сумматоры до оптимизации	Сумматоры после оптимизации	Умножители до оптимизации	Умножители после оптимизации
Граф 1	5	3	3	1
Граф 2	4	2	9	4
Граф 3	12	5	10	7

Данное программное средство позволяет выделить множество повторяющихся подграфов в графе вычислительного процесса с целью переиспользования ресурсов, использующихся для реализации на RTL-уровне.

Список использованных источников:

1. Ullmann Julian R. An algorithm for Subgraph Isomorphism // Journal of the Association for Computing Machinery – 1976. – 23. – P. 31-4

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. Харари Ф. Теория графов / Ф. Харари – М.: Ленанд, 2018. – 304 с;
3. Crowe, J. E., Lynch, M. F., Town, W. G. *Analysis of Structural Characteristics of Chemical Compounds in a Large Computer-Based File. I: Non-Cyclic Fragments.* J. Chem. Soc 1970, (C), 990-996;
4. H. Bunke, P. Foggia, *A comparison of algorithms for maximum common subgraph on randomly connected graphs.*

ИССЛЕДОВАНИЕ ЗАДАЧ ОПТИМИЗАЦИИ РАБОТЫ ГРУЗОВЫХ ТЕРМИНАЛОВ

Голубович Ю.И.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Дугинов О.И. – канд. физ.-мат. наук

В настоящей работе рассматривается задача, в которой требуется разбить фиксированную долю реберно-взвешенного сбалансированного полного двудольного графа на заданное количество подмножеств и найти в этом графе совершенное паросочетание по критерию минимума максимального суммарного веса ребер паросочетания, инцидентных вершинам одного из подмножеств разбиения доли. Анализируются варианты этой задачи, в которых ограничены сверху мощности подмножеств разбиения доли или суммы весов ребер паросочетания, инцидентных любому из подмножеств разбиения доли. Выделены полиномиально разрешимые и NP-трудные случаи задач. Обсуждаются целочисленные модели.

Постановка задач. Приведем содержательные постановки рассматриваемых в работе задач. Пусть U, W – множества соответственно контейнеров на временной площадке и свободных мест на основной площадке в терминале; m – число погрузчиков, которые перемещают контейнеры с временной площадки на основную. Для любого контейнера u из множества контейнеров U и любого места w из множества W известно время перемещения c_{uw} контейнера u на место w . В задаче требуется, во-первых, каждому контейнеру u из множества U назначить место w из множества W и, во-вторых, каждому из m погрузчиков назначить перевозимые им контейнеры таким образом, чтобы время завершения работы последнего погрузчика было минимальным. Будем говорить об этой задаче как о задаче A . Добавив к задаче A дополнительное условие, а именно ограничение сверху на количество контейнеров, перевозимых i -м погрузчиком с временной площадки на основную ($i = 1, 2, \dots, m$), получим задачу B . Наконец, если к задаче A добавить дополнительное условие – ограничения сверху на общее время работы каждого из погрузчиков, то получим задачу C .

Представим формулировки рассматриваемых задач в теоретико-графовой терминологии. Пусть $G = (U \cup W, E)$ – сбалансированный полный двудольный граф с долями U, W и множеством ребер E . Для любого ребра графа задан неотрицательный целочисленный вес $\omega: E \rightarrow \{0, 1, 2, \dots\}$.

Также задано натуральное число m . Пусть $Y = \{U_1, U_2, \dots, U_m\}$ – это разбиение доли U на m подмножеств (U_i – это множество контейнеров, перевозимых i -м погрузчиком) и M – это совершенное паросочетание графа G , которое задает соответствие между контейнерами и позициями на основной контейнерной площадке. Стоимость совершенного паросочетания M относительно разбиения Y определим следующим образом:

$$c_Y(M) = \max\{\omega(\delta(U_1) \cap M), \omega(\delta(U_2) \cap M), \dots, \omega(\delta(U_m) \cap M)\},$$

где $\delta(U_i)$ – множество ребер графа G , инцидентных вершинам подмножества U_i . В задаче требуется найти разбиение множества U на m необязательно непустых подмножеств $Y = (U_1, U_2, \dots, U_m)$ и совершенное паросочетание M графа G с минимальной стоимостью $c_Y(M)$. Если мы дополнительно потребуем, чтобы выполнялось условие $|U_i| \leq b_i$ для каждого $i = 1, 2, \dots, m$, где b_1, b_2, \dots, b_m – это заданные целые неотрицательные числа, то получим задачу B . Если дополнительно потребовать, чтобы выполнялось условие $\omega(\delta(U_i) \cap M) \leq c_i$ для каждого $i = 1, 2, \dots, m$, то получится задача C .

Сформулируем задачу A в виде задачи распознавания. Условие: задан сбалансированный полный двудольный граф $G = (U \cup W, E)$ с весами на ребрах $\omega: E \rightarrow \{0, 1, 2, \dots\}$ и целое неотрицательное число k . Вопрос: существуют ли совершенное паросочетание M в графе G и разбиение Y доли U на m подмножеств такие, что $c_Y(M) \leq k$? Распознавательные версии задач B и C имеют схожий вид.

Цель работы заключается в исследовании вопросов, связанных со сложностью решения рассматриваемых задач. Мотивацией для исследований служит слабая изученность задач и актуальность их приложений в области оптимизации работ контейнерных терминалов. Отметим также, что в литературе изучаются и другие задачи, которые моделируют различные ситуации, возникающие при транспортировке грузовых контейнеров [1 – 3].

Переходим к описанию полученных результатов.

Задача А для сбалансированного полного двудольного графа с весами 0, 1 на ребрах.

Предлагается следующий алгоритм решения задачи А, ограниченной сбалансированными полными двудольными графами с весами на ребрах из множества $\{0, 1\}$:

Шаг 1. Найдем паросочетание минимального веса M с помощью любого известного полиномиального (псевдополиномиального) алгоритма [4 – 6].

Шаг 2. Разобьем множество вершин доли U графа G на m подмножеств $Y = (U_1, U_2, \dots, U_m)$ следующим образом:

Шаг 2.1. Упорядочим ребра паросочетания M в порядке неубывания их веса; пусть e_1, e_2, \dots, e_n – это список ребер паросочетания M такой, что $\omega(e_i) \geq \omega(e_{i+1})$ для каждого $i = 1, 2, \dots, n-1$;

Шаг 2.2. По упорядоченности ребер M циклично относительно подмножеств разбиения будем добавлять инцидентную очередному ребру вершину в подмножество разбиения Y (т.е. в U_1 добавим инцидентную ребру e_1 вершину u_1 , в U_2 – инцидентную ребру e_2 вершину u_2 и т.д.).

Таким образом, получим разбиение Y доли U . Методом от противного можно доказать, что полученное решение (M, Y) оптимально. Нетрудно видеть, что представленный алгоритм является полиномиальным. Таким образом, верна следующая теорема.

Теорема 1. Задача А решается за полиномиальное время в классе сбалансированных полных двудольных графов с весами 0, 1 на ребрах.

Задача В для сбалансированных полных двудольных графов с весами 0, 1 и 2 на ребрах. В работе [1] установлена сильная NP-трудность задачи В. При этом веса ребер графа в задаче не ограничены никакой константой. Следующая теорема усиливает этот результат.

Теорема 2. Задача В в классе сбалансированных полных двудольных графов с весами 0, 1 и 2 является NP-трудной.

В основе доказательства этой теоремы лежит полиномиальное сведение от NP-трудного случая задачи Наибольшая Часть Совершенного Паросочетания [7].

Целочисленная линейная модель задачи С. Сформулируем задачу С в виде задачи целочисленного линейного программирования.

Пусть $G = (U \cup W, E)$ – сбалансированный полный двудольный граф с заданными весами на ребрах $\omega: E \rightarrow \{0, 1, 2, \dots\}$, \tilde{c}_i – максимальное время работы i -го погрузчика, $i \in \{1, \dots, m\}$. Без ограничения общности можем полагать, что $U = \{1, 2, \dots, p\}$ и $W = \{1, 2, \dots, q\}$. Задача С в виде задачи ЦЛП имеет следующий вид:

$$\left\{ \begin{array}{ll} t \rightarrow \min & \\ t \geq \sum_{i \in U} \sum_{j \in W} c_{ij} z_{ijk}, & \forall k \in \{1, \dots, m\} \\ \sum_{i \in U} \sum_{j \in W} c_{ij} z_{ijk} \leq \tilde{c}_k, & \forall k \in \{1, \dots, m\} \\ z_{ijk} \geq y_{ik} + x_{ij} - 1, & \forall i \in U, j \in W, k \in \{1, \dots, m\} \\ z_{ijk} \geq 0, & \forall i \in U, j \in W, k \in \{1, \dots, m\} \\ \sum_{j \in W} x_{ij} \leq 1, & \forall i \in U \\ \sum_{i \in U} x_{ij} \leq 1, & \forall j \in W \\ \sum_{k=1}^m y_{ik} = 1, & \forall i \in U \\ x_{ij} \in \{0, 1\}, & \forall \{i, j\} \in E \\ y_{ik} \in \{0, 1\}, & \forall i \in U, k \in \{1, \dots, m\} \end{array} \right.$$

где z_{ijk} – переменная, принимающая значение 1, если в графе G ребро $\{i, j\}$ принадлежит паросочетанию M и i -я вершина доли U принадлежит подмножеству разбиения U_k , иначе – 0; c_{ij} – вес ребра $\{i, j\}$ графа G ; x_{ij} – переменная, соответствующая ребру $\{i, j\}$ графа G , принимает значение 1, если ребро принадлежит паросочетанию M , иначе – 0; y_{ik} – переменная, принимающая значение 1, если i -я вершина доли U графа G принадлежит подмножеству разбиения U_k , иначе – 0.

Список использованных источников:

1. Kress, D. The partitioning min-max weighted matching problem / D. Kress, S. Meiswinkel, and E. Pesch // Eur. J. Oper. Res. – 2015. Vol. 247. – P. 745-754.
2. Li X., Otto A., Pesch E. Solving the single crane scheduling problem at rail transshipment yards / X. Li, A. Otto, E. Pesch // Discrete Applied Mathematics. – 2019. – Vol. 264. – P. 134 – 147.
3. Meiswinkel, S. On Combinatorial Optimization and Mechanism Design Problems Arising at Container Ports. Springer Gabler, Wiesbaden, 2018. 123 p.

4. Ramshaw, L. *A weight-scaling algorithm for min-cost imperfect matchings in bipartite graphs* / L. Ramshaw, R. E. Tarjan // *In Proc. 53rd Annu. Symp. Foundations of Computer Science, New Brunswick, NJ, USA, Oct. 20–23, 2012 (IEEE, Piscataway, 2012)*, pp. 581–590.
5. H. N. Gabow and R. E. Tarjan, *Faster scaling algorithms for general graph matching problems* / H. N. Gabow, R. E. Tarjan // *J. ACM.* – 1991. Vol. 38 (4). – P. 815-853.
6. Gabow, H. N. *Faster scaling algorithms for network problems* / H. N. Gabow, R. E. Tarjan // *SIAM J. Comput.* – 1989. – Vol. 18. – P. 1013-1036.
7. Дугинов, О.И. *Взвешенное совершенное паросочетание с ограничениями на суммарный вес его частей* / О.И. Дугинов // *Дискретный анализ и исследование операций.* – 2021. – Т. 28., № 3. – С. 5-37.

ПРИМЕНЕНИЕ ФИЗИЧЕСКОЙ КРИПТОГРАФИИ В УСТРОЙСТВАХ НА БАЗЕ FPGA

Карбовский Д.В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Иванюк А.А. – д-р. техн. наук, доцент

Данная работа описывает применение методов физической криптографии для защиты софт-процессоров на базе FPGA от нелегального использования. Проводится исследование результатов применения физически неклонированной функции типа арбитр. Приводятся значения метрик для различных конфигураций.

Роль вычислительных устройств на базе полупроводниковых технологий неопределима в современном мире. Разработка и производство полупроводниковых кристаллов, однако, являются крайне дорогими процессами и связаны с большими рисками. Это привело к потребности в создании платформы для прототипирования полупроводниковых устройств, а также поиска альтернативных способов создания устройств. Таким решением стали устройства на базе FPGA (Field-Programmable Gate Array – программируемая логическая матрица). В отличие от устройств типа ASIC (Application-Specific Integrated Circuit – интегральные схемы конкретного предназначения), где архитектура микропроцессора предопределена его архитектурой, устройства типа FPGA содержат набор базовых элементов (элементы логики, памяти, коммуникации и т.д.), соединения между которыми устанавливаются конфигурацией. Такой подход позволяет использовать кристаллы с готовым набором элементов, лишь меняя конфигурацию устройства, но не его топологию. Это позволяет дешёвое прототипирование вычислительных устройств и даже использование FPGA в качестве базы для готовых устройств.

Одной из проблем такого подхода является проблема распространения файла конфигурации кристалла FPGA. Одна и та же конфигурация, загруженная в аналогичный кристалл, приведёт к созданию логической копии выпускаемого устройства. Поскольку конфигурация не хранится кристаллом, а передаётся из внешнего источника, такая архитектура устройства является крайне уязвимой к нелегальному копированию. Для защиты конфигурации устройства от нелегального использования применяются методы физической криптографии, в частности, физически неклонированные функции (ФНФ). Такие функции используют особенности физических объектов для формирования ответов для набора запросов. Набор таких пар запросов-ответов является уникальным для каждого устройства. Для определения качеств ФНФ были разработаны метрики оценки, такие как стабильность и уникальность.

Примером физически неклонированной функции является ФНФ типа арбитр. Данная функция использует разницу в длине физических путей на кристалле. Даже топологически одинаковые пути ввиду технологической несовершенности будут отличаться по длине. Это приводит к тому, что чаще всего по одному из путей сигнал будет проходить быстрее, чем по другому.

Разработка устройств на базе FPGA происходит в автоматизированных средах разработки. Инженер пользуется языками описания аппаратуры, код на которых транслируется в бинарный файл. Среда проектирования выбирает базовые элементы и линии для соединения на кристалле, используя алгоритмы оптимизации, длины путей и использования кристалла. Внесение даже незначительных изменений в описание устройства может повлечь создание совершенно другой конфигурации с отличным использованием элементов чипа. Как следствие этого, выбор соединительных линий может значительно измениться в сравнении с предыдущим описанием.

Нелегальное использование конфигурации устройства подразумевает различные способы копирования. Помимо обычного копирования злоумышленник может использовать дополнительный инструментарий для восстановления проекта из битового файла, внести изменения в проект и синтезировать устройство с собственным функционалом.

Использование физически неклонированных функций позволяет устройству на базе FPGA производить самоверификацию. ФНФ позволяет получить уникальный отпечаток для данного кристалла. Отпечаток представляет собой набор битов, которые устройство сравнивает с контрольным значением. Если значения не совпадают, устройство может не включаться. В таком случае, простое копирование описания на другой кристалл не позволит клонировать продукт.

В случае, если злоумышленник меняет описание устройства, повторный синтез конфигурации приведёт к изменению соединительных путей, и, как следствие, изменению времени прохода сигнала по ним. Это влечёт изменения в работе всех физически неклонированных функций. Как итог, множество их ответов изменяется и перестаёт совпадать с контрольными значениями. Это является эффективным методом защиты от нелегального копирования устройств на базе FPGA.

Для оценки работы физически неклонированных функций используются различные метрики. Самые важные метрики в данном контексте – это стабильность работы и уникальность функции. Стабильность работы необходима для обеспечения одинаковых ответов на запросы для большинства ситуаций. Она выражается относительной величиной и означает долю корректных ответов. Уникальность может быть разделена на две категории: внутрикристальную и межкристальную. Для расчёта уникальности анализируется набор ФНФ, реализованных в аппаратуре. Набор, расположенный на одном кристалле, используется для получения внутрикристальной уникальности, а на различных – межкристальной. Уникальность ФНФ рассчитывается как сумма удельных хемминговых расстояний между наборами ответов на один и тот же запрос.

Одна из реализаций ФНФ типа арбитр использует цепочку мультиплексоров и линии соединения между ними [1]. Изменение значения управляющего сигнала переключает используемые линии, изменяя длину полного пути. Схема ФНФ типа арбитр представлена на рисунке 1.

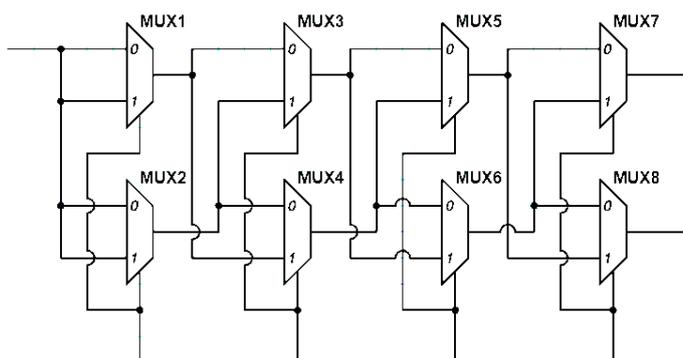


Рисунок 1 – Схема физически неклонированной функции типа арбитр на базе цепочки из мультиплексоров.

Данная схема была реализована на языке VHDL с использованием софт-процессора Microblaze в среде проектирования Vivado. Тесты производились на плате Nexys 4 на базе кристалла Artix 7. Описание аппаратуры было синтезировано в различных конфигурациях (количество реализованных ФНФ на кристалле, длина арбитра) и протестировано с различными параметрами. Целью тестов было исследование изменения метрик для разных конфигураций. Стабильность для всех конфигураций составила более 0.99 для всех тестов. Результаты тестов приведены в таблице 1.

Таблица 1 – Результаты тестирования

Длина арбитра (ширина запроса)	Количество запросов при тестировании	Внутрикристальная уникальность	Межкристальная уникальность
8	256	0.38 (38%)	0.0156 (1.56%)
10	256	0.40 (40%)	0.0195 (1.95%)
12	256	0.35 (35%)	0.0008 (0.08%)
14	256	0.35 (35%)	0.0040 (0.40%)
16	1024	0.36 (36%)	0.0156 (1.56%)
18	1024	0.37 (37%)	0.0137 (1.37%)
20	1024	0.43 (43%)	0.0117 (1.17%)
22	1024	0.44 (44%)	0.0186 (1.86%)
24	1024	0.38 (38%)	0.0117 (1.17%)
26	1024	0.41 (41%)	0.0225 (2.25%)
28	1024	0.41 (41%)	0.0225 (2.25%)
30	1024	0.41 (41%)	0.0098 (0.98%)
32	1024	0.38 (38%)	0.0215 (2.15%)

Полученная межкристальная уникальность для простой реализации низкая. Эксперименты показали, что увеличение длины ФНФ типа арбитр не приводят к однозначным улучшениям показателей, в то время как внутрикристальная уникальность достаточно высокая. Выгоднее всего использовать несколько копий более дешёвой реализации для достижения лучшей защиты.

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

Проектирование и производство вычислительных устройств – дорогостоящие процессы. Некоторые устройства выпускаются на базе FPGA и нуждаются в защите от нелегального копирования. Методы физической криптографии позволяют защитить конфигурацию от использования на нелицензированном кристалле. ФНФ типа арбитр требует подбора длины и других параметров для получения высокой уникальности, но недорого в реализации и обладает хорошей стабильностью, пригодной для уникальной идентификации полупроводниковых кристаллов.

Список использованных источников:

1. Ярмолик, В.Н. Физически неклонированные функции / В.Н. Ярмолик, Ю.Г. Вашинко // Информатика. -- 2011. -- № 2 (30). – С. 92-103.

ИСПОЛЬЗОВАНИЕ BIG DATA В КОМПЬЮТЕРНЫХ ИГРАХ

Протасов А.П., Стаселько И.Д., Позняков Т.Д.

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Анисимов В.Я. – Кандидат физико-математических наук, доцент

Большие данные стали актуальным понятием, так как интернет сфера постоянно расширяется и прогрессирует. В данной работе анализируется использование больших данных в видеоиграх, технологии баз данных для решения задач игровой отрасли. Нами рассматриваются примеры использования больших данных в видеоиграх и способы монетизации данных приложений. Проблемы разработки видеоигр касаются в основном игрового дизайна и количества различных характеристик.

Введение. Big data — это различные инструменты, подходы и методы обработки как структурированных, так и неструктурированных данных для того, чтобы их использовать для конкретных задач и целей. В современном мире Big data—социально-экономический феномен, который связан с тем, что появились новые технологические возможности для анализа огромного количества данных. Большие данные помогают разработчикам балансировать, тестировать и монетизировать свои игры. В будущем большие данные, скорее всего, будут иметь еще больше вариантов использования в видеоиграх и их разработке. Есть несколько примеров того, как игровые компании успешно внедряют решения больших данных с самого начала разработки своих продуктов, и есть некоторые компании, которые адаптировались к ним позже. Игровым компаниям не обязательно использовать большие данные, чтобы предоставлять свои услуги должным образом, в отличие от популярных социальных сетей, которые обрабатывают огромные объемы данных каждую секунду. Тем не менее, такие компании, как Electronic Arts, являются прекрасными примерами того, как большие данные могут помочь игровой компании добиться успеха.

Уравнивание шансов. Балансировка является важной частью разработки как одиночной, так и многопользовательской игры, т.к. игра должна оставаться честной и захватывающей. Сложность однопользовательских игр должна увеличиваться постепенно, чтобы игроки не выходили из игры, потому что игра слишком легкая или слишком сложная. Многопользовательские игры требуют балансировки для асимметричных функций, таких как игровые персонажи с разными атрибутами и способностями. Большие данные можно использовать для балансировки как одиночной, так и многопользовательской игры, но в рамках данной работы мы рассмотрим многопользовательский аспект.

Балансировка персонажа. Electronic Arts собрала более 35 миллионов игроков для игры в шутер от первого лица Apex Legends[1]. Игра была запущена в 2019 году с 22 различными игровыми персонажами, известными как герои, было представлено четыре различных режима игры и девять разных карт. Существует четыре разные категории героев, и все герои имеют разные навыки, которые Electronic Arts необходимо сбалансировать. Игры с таким количеством героев и способностей не могут быть полностью сбалансированы для всех режимов игры, карт и навыков игрока(от уровня начинающих до профессиональных игроков). Поэтому некоторые комбинации героев стали так называемым «мета», что означает, что эти герои и комбинации предпочтительнее других. В течение 2020 года игроки высказывали опасения, что некоторые, более мобильные герои стали слишком сильными, и так называемая «dive comp meta»(группировка некоторых персонажей с целью координации атак на противника) не позволяет полноценно играть другими героями, что делает игру предсказуемой и однозначной.

Балансировка игровой среды. В дополнение к балансировке героев, некоторые карты в играх также нуждаются в балансировке. Приведем соответствующий пример. В режиме игры под названием «Штурм» атакующая команда, состоящая из шести человек, имеет десять минут, чтобы захватить две команды защиты, также состоящих из шести человек, делающих все возможное для предотвращения захвата. На официальном игровом форуме игроки начали высказывать опасения по поводу режима игры, благоприятствующего защищающейся стороне.

Заключение. Индустрия видеоигр выросла больше, чем кто-либо мог себе представить, когда впервые были опубликованы коммерческие видеоигры. Этого не произошло бы без помощи Интернета, хотя игры имели свою справедливую долю игроков еще до настоящего Интернета и распространились по всему миру. Видеоигры могут быть запущены на различных видах устройств, и подключение к Интернету воспринимаются как должное для многих из этих новых игр, которые появились в 21 веке. Объем данных в интернете вырос неожиданно быстро, и в конечном итоге традиционные реляционные базы данных не могли больше идти в ногу с объемом, скоростью и разнообразием данных. В то время как реляционные базы данных все еще служат своей цели для небольших томов, скорости и разнообразия данных, концепция больших данных была необходима, чтобы справиться с этой проблемой с базами данных NoSQL и фреймворками, такими как Hadoop и HPCSS. Большие данные принесли больше в таблицу разработчиков игр с точки зрения балансировки, монетизации и тестирования. Решения по балансировке и монетизации все еще могут быть приняты с помощью догадки, но всегда есть данные, доступные для изучения эффектов должным образом. Разработчики также могут объяснить свои действия или бездействие, показав сырую статистику своим игрокам. С надеждой игроки все равно будут услышаны, хотя разработчики могут просто посмотреть на данные и увидеть, если кажется, что-то не так вместо того, чтобы читать мнения игроков на игровых форумах. Ведь не имеет значения, хороша ли игра, какая модель монетизации используется или нет, насколько точно можно получить статистику, если не хватает платящих клиентов для игры.

Список использованных источников

[1]. Electronic arts game site. [Online]. Available: <https://www.ea.com/ru-ru/games/apex-legends/about> .

ИСПОЛЬЗОВАНИЕ АЛГОРИТМА MFCC В ЗАДАЧЕ РАСПОЗНАВАНИЯ МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ

*Петровец В.Н., магистрант гр.256241, Мискевич П.Л., магистрант гр.256241,
Деменковец Д.В., аспирант*

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Деменковец Д.В. – старший преподаватель, аспирант

В данной работе описан алгоритм нахождения Mel-частотного кепстра и применения его в задаче идентификации индивидуальной музыкальной композиции. Также предложен алгоритм автоматизированного распознавания звуковых сигналов с использованием MFCC.

Распознавание музыки или возможность автоматически идентифицировать и классифицировать музыкальный контент в последние годы привлекли значительное внимание из-за распространения цифровой музыки и необходимости эффективного управления музыкой, рекомендаций и защиты авторских прав. Одним из ключевых методов, используемых в процессе распознавания музыки, является алгоритм кепстральных коэффициентов Mel-частоты (MFCC), который получил широкое распространение благодаря своей эффективности в захвате акустических характеристик музыкальных сигналов. Алгоритм MFCC основан на свойствах восприятия слуховой системы человека и применяется в различных задачах распознавания музыки, таких как классификация жанров, определение настроения и распознавание инструментов. Однако, несмотря на его популярность, все еще существует потребность во всесторонней оценке эффективности алгоритма MFCC в распознавании музыки, включая его сильные стороны, ограничения и потенциальные области для улучшения.

Исследования показали, что функции MFCC, извлеченные из аудиосигналов, могут эффективно представлять тембровые характеристики музыки, что имеет решающее значение для таких задач, как классификация жанров и определение настроения. MFCC также использовался в сочетании с другими методами, такими как скрытые марковские модели (HMM) и машины опорных векторов (SVM), для достижения высокой точности в задачах распознавания музыки. Однако следует также учитывать определенные ограничения алгоритма MFCC. Одним из основных ограничений является его чувствительность к шуму и изменчивости музыкальных сигналов, что может повлиять на точность и надежность системы распознавания. Кроме того, алгоритм MFCC может не захватывать музыкальные особенности более высокого уровня, такие как мелодическая или гармоническая информация, которые могут быть важны для определенных задач распознавания музыки. Кроме того, на производительность алгоритма MFCC может влиять выбор параметров, таких как количество кепстральных коэффициентов, размер набора фильтров Mel, а также длина кадра и перекрытие, используемые в процессе выделения признаков.

Особенность использования алгоритма MFCC для анализа музыкальных композиций заключается в термине Mel-частота. Mel – единица высоты звука, основанная на восприятии этого звука органами слуха человека. Амплитудно-частотная характеристика человеческого слухового аппарата нелинейна, как показывают исследования (рисунок 1).

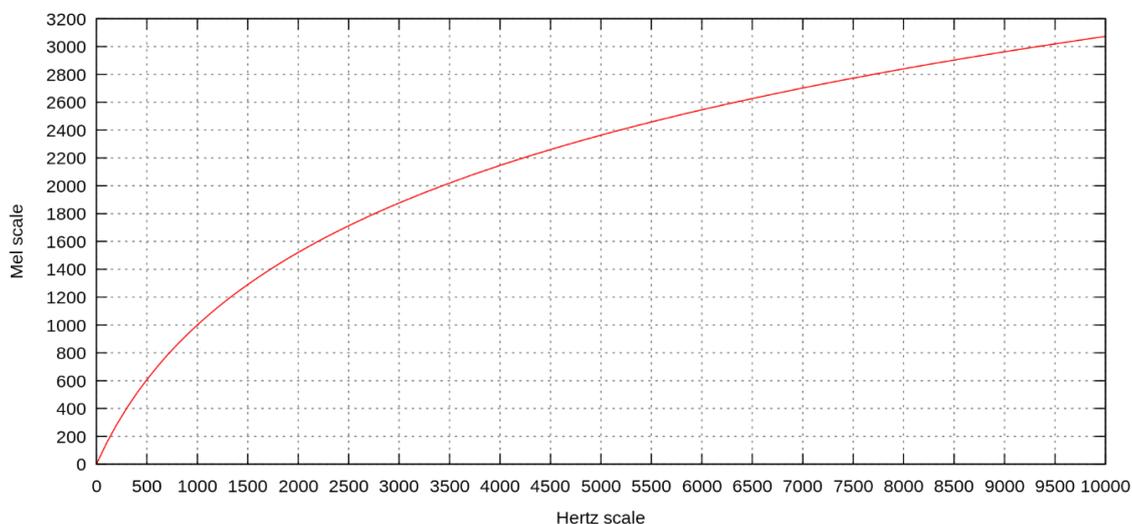


Рисунок 1 – Зависимость единицы измерения Mel-частот от Герц

Далее приведено описание алгоритма MFCC для участка музыкальной композиции:

1. Преобразование звукового кадра из временной в частотную область с помощью оконного преобразования Фурье.

2. Отображение частотных бинов на Mel-шкалу, используя треугольные окна с наложением

3. Расчет логарифмов отображенных значений Mel-диапазонов

4. Применение дискретного косинусного преобразования (DCT) к Mel-диапазонам.

Параметрами алгоритма являются: количество Mel-диапазонов (в литературе встречаются значения от 20 до 80), кол-во коэффициентов DCT (10-15), размер окна, коэффициент наложения окон.

Результат работы алгоритма MFCC – двумерная матрица, хранящая в себе векторы признаков для каждого окна исходного сигнала. Впоследствии набор данных векторов может использоваться для классификации исходной звуковой композиции.

Для использования алгоритма MFCC в процессе распознавания музыки необходимо выполнить следующий порядок действий:

1. Подготовка набора данных. Необходимо собрать разнообразный набор данных музыкальных сигналов, включающий различные жанры, настроения и инструменты, чтобы обеспечить всестороннюю оценку производительности алгоритма MFCC. Набор данных будет предварительно обработан для удаления любых артефактов (шума), нормализации уровней звука и разделен на наборы для обучения и тестирования.

2. Извлечение признаков. Алгоритм MFCC будет реализован для извлечения признаков из музыкальных сигналов в наборе данных. Различные конфигурации параметров, перечисленные при описании MFCC, должны быть протестированы для изучения их влияния на производительность алгоритма.

3. Распознавание музыки. Значения коэффициентов MFCC, извлеченные из аудиосигналов, используются в качестве входных данных для классификатора машинного обучения, такого как машина опорных векторов (SVM) или нейронная сеть, которая обучается на помеченном наборе данных музыкальных треков.

4. Оценка эффективности. Производительность алгоритма MFCC в задачах распознавания музыки должна оцениваться с использованием соответствующих показателей, таких как точность, воспроизводимость и оценка F1. Результаты должны быть проанализированы статистически, чтобы выявить любые существенные различия в производительности между алгоритмом MFCC и другими методами извлечения признаков.

MFCC широко используется в цифровой обработке сигналов для получения информации о звуковых композициях. Кроме того, данный алгоритм может быть использован в обработке изображений для аннулирования оттенков серого черно-белых изображений. Извлеченная из короткого окна, совокупность MFCC коэффициентов воспринимается как тембральный дескриптор звукового сигнала. В данной работе был предложен и описан подход для использования алгоритма MFCC в процессе анализа и индивидуального распознавания музыкальных композиций.

Список использованных источников:

1. Li, T.L.H. *Genre classification and the invariance of MFCC features to Key and Tempo* / T.L.H. Li, A.B. Chan // *International Conference on MultiMedia Modeling, Taipei, 2011*.

2. Prabakaran, D. *Speech Processing: MFCC Based Feature Extraction Techniques- An Investigation* / D. Prabakaran, S. Sriuppili // *Journal of Physics: Conference Series / IOP Publishing – London, 2020*.

3. Analytics Vidhya [Electronic resource] : *MFCC Technique for Speech Recognition*. – Mode of access: <https://www.analyticsvidhya.com/blog/2021/06/mfcc-technique-for-speech-recognition/>. – Date of access: 24.03.2023.

4. Gupta, S. *Feature extraction using MFCC* / S. Gupta [at et.] // *Signal & Image Processing : An International Journal (SIPIJ) / Indian institute of Information and Technology – Allahabad, 2013*.

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ШИФРОВ ЗАМЕНЫ

Бигеза Я. О., Рыбак Д. В.

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Стройникова Е. Д. – ассистент каф. информатики

В настоящее время данные являются самой ценной информацией, поэтому существует множество методов защиты. Их изучает наука криптография. Изначально она изучала методы шифрования информации.

Шифр — это сам алгоритм, по которому преобразуется сообщение, которое мы хотим передать. Шифрованное сообщение – это сообщение, прошедшее процесс шифрования. Расшифровать его можно с помощью ключа.

Криптостойкость шифра определяется по количеству времени, необходимому для его взлома. Шифры разделяются на **шифры перестановки**, **композиционные шифры** и **шифры замены**. Шифрами замены называются такие шифры, преобразования из которых приводят к замене каждого символа исходного сообщения на другие символы, причем порядок их следования совпадает с порядком следования соответствующих им символов исходного сообщения.

Математическая модель произвольного шифра замены:

$$S_A = (x, k, y, e, d)$$

где x — символ открытого текста в алфавите A , y — символ шифротекста в алфавите B , k — ключ шифра, E_k — преобразование шифра с ключом k_s , D_k — преобразование расшифрования с ключом k_p .

Шифров замены существует множество, мы выделили некоторые из них.

1. Шифр Вернама (XOR-шифр) (1890 г. - 1960 г.).

Шифр основан на бинарной логике и обладает криптографической стойкостью. Без знания ключа расшифровать его невозможно (доказано Клодом Шенноном(1916 - 2001)).

Исходный алфавит — латиница, символы шифротекста - двоичный код по таблице ASCII, ключ шифра - слово на языке исходного сообщения, где каждому символу также присвоен двоичный код. Ключ записывается столько раз, сколько потребуется для того, чтобы каждому символу исходного сообщения соответствовал символ ключа. Между символами сообщения и ключа проводится операция XOR - "исключающее или" (если все сигналы равны между собой: 0-0 или 1-1, то на выходе получаем 0, если сигналы не равны: 0-1 или 1-0, то на выходе получаем 1). Таким образом, мы получаем новую запись в виде двоичного кода.

С образцом программы, выполняющей алгоритм данного шифра на языке C++ вы можете ознакомиться по [ссылке](#).

2. Аффинный шифр.

Для реализации аффинного шифра необходимо выбрать 2 числа для использования в качестве ключа шифрования, обозначим их как a и b . Число a должно быть взаимно простым с 26. Пусть каждая буква текста, который необходимо зашифровать, имеет соответствующий числовой индекс от 0 до 25. Затем для каждой буквы текста нужно вычислить новый индекс, умножив ее индекс на число a и прибавив остаток от деления на 26. Полученный результат будет соответствовать другой букве в алфавите. Таким образом, каждая буква текста заменяется на другую букву, образуя зашифрованный текст.

Чтобы расшифровать текст, используется обратная формула:

$$y = a^{-1} * (x - b) \pmod{26}$$

где x — индекс зашифрованной буквы, a^{-1} — обратное число к a , y — индекс расшифрованной буквы. Используя таблицу соответствия букв и индексов, мы можем найти соответствующую букву для каждого вычисленного значения y , чтобы получить исходный текст.

С образцом программы, выполняющей алгоритм данного шифра на языке C++ вы можете ознакомиться по [ссылке](#).

3. Шифр «Квадрат Полибия» (ок. 200 до н. э., — ок. 120 до н. э.)

Квадрат Полибия представляет собой квадрат 5×5, столбцы и строки которого нумеруются цифрами от 1 до 5. В каждую клетку этого квадрата записывается по алфавиту одна буква.

Существует несколько методов шифрования с помощью квадрата Полибия. Ниже приведены три из них.

Метод 1: Для шифрования на квадрате находили букву текста и вставляли в шифр нижнюю от неё в том же столбце. Если буква была в нижней строке, то брали верхнюю из того же столбца.

Метод 2: Сообщение преобразуется в координаты по квадрату Полибия, координаты записываются вертикально. Затем координаты считывают по строкам: Далее координаты преобразуются в буквы по этому же квадрату:

Метод 3: Полученный первичный шифротекст шифруется вторично. При этом он выписывается без разбиения на пары. Полученная последовательность цифр сдвигается циклически влево на нечетное количество шагов. Эта последовательность вновь разбивается в группы по два. И по таблице заменяется на окончательный шифротекст.

Квадрат Полибия является примером шифра замены, поэтому неустойчив к частотной атаке.

4. Шифр Плейфера

Шифр Плейфера использует матрицу 5×5 для латинского алфавита, содержащую ключевое слово. Чтобы составить ключевую матрицу, нужно сначала заполнить пустые ячейки матрицы буквами ключевого слова без повторяющихся символов, потом заполнить оставшиеся ячейки матрицы символами алфавита, не встречающимися в ключевом слове, по порядку.

Для того чтобы зашифровать сообщение, необходимо разбить его на группы из двух символов. Два символа биграммы соответствуют углам прямоугольника в ключевой матрице. Определяем положения углов этого прямоугольника относительно друг друга. Затем, руководствуясь следующими 4 правилами, зашифруем пары символов исходного текста.

Если два символа биграммы совпадают (или если остался один символ), заменяем второй символ на «X», зашифровываем новую пару символов и продолжаем.

Если символы биграммы исходного текста встречаются в одной строке, то эти символы заменяются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящиеся непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Для расшифровки необходимо использовать инверсию этих четырёх правил, убрав из алфавита символы «X» (или «Q»), если они не несут смысла в исходном сообщении.

В настоящее время, в связи с быстрым развитием технологий, многие из шифров замены уже не способны обеспечить необходимую степень безопасности. Поэтому необходимо постоянно совершенствовать методы защиты информации, а в случае шифров замены необходимо еще правильно подбирать ключи. Также некоторые из них неустойчивы к некоторым атакам, что делает их менее надежными.

Список использованных источников:

1. Шнайер Б. Прикладная криптография. 2012. Гл. 1.4-1.5.
2. Криптографическая защита информации: учебное пособие / А.В. Яковлев, А.А. Безбогов, В.В. Родин, В.Н. Шамкин. Тамбов : ТГТУ, 2020. 140 с.
3. Алферов А.П., Зубов А.Ю. Основы криптографии. - М.: Гелиос АРВ, 2005.

ИССЛЕДОВАНИЕ СВОЙСТВ ДИЗЪЮНКТИВНЫХ ГРАФОВ И ИХ ПРИМЕНЕНИЕ В КАЧЕСТВЕ ИНСТРУМЕНТА ДЛЯ АНАЛИЗА СОЦИАЛЬНЫХ СЕТЕЙ

Давыдовский Д. В., студент гр.253504

*Белорусский государственный университет информатики и радиоэлектроники
г.Минск, Республика Беларусь*

Егорова Н.Г. - доцент, кандидат технических наук

Аннотация. Статья рассматривает исследование свойств дизъюнктивных графов и их применение в качестве инструмента для анализа социальных сетей. Дизъюнктивный граф представляет собой граф, где каждое ребро может быть либо присутствующим, либо отсутствующим, в зависимости от наличия связи между вершинами. Статья описывает различные метрики, используемые для анализа дизъюнктивных графов, включая центральность, кластеризацию и плотность сети. Дизъюнктивный граф может быть использован для определения наиболее важных узлов или групп узлов в социальных сетях, которые обладают определенными свойствами. Эта статья может быть полезна для исследователей, занимающихся анализом социальных сетей и использующих дизъюнктивные графы в своих исследованиях.

Ключевые слова. Анализ социальных сетей, структура социальных сетей, графовое представление социальных сетей, размер сети, плотность сети, среднее расстояние между вершинами в сети, диаметр сети, центральность, отдельные вершины в сети, коэффициент кластеризации, группы или сообщества в сети, дизъюнктивный граф, важность узлов в социальной сети, метрики центральности, взаимосвязи между людьми

Социальная сеть - это веб-платформа или приложение, которое позволяет людям взаимодействовать друг с другом онлайн, обмениваться информацией, содержанием, мнениями, интересами и связываться в различных целях. Социальные сети предоставляют своим пользователям инструменты для создания профилей, поиска и добавления друзей, отправки сообщений, обмена контентом и просмотра контента, созданного другими пользователями. На социальных сетях можно создавать и присоединяться к группам по интересам, участвовать в обсуждениях и событиях, делиться своими успехами, переживаниями и мнениями с другими. Социальные сети стали очень популярными в последние десятилетия и играют значительную роль в современной коммуникации и обмене информацией. Для того, чтобы связать тему социальных сетей и дизъюнктивных графов, дадим другое определение социальным сетям. Социальные сети – это сети взаимосвязей между людьми, группами людей, организациями или другими социальными единицами. Они могут быть представлены в виде графов, где вершины представляют социальные единицы, а ребра – связи между ними.

Дизъюнктивный граф (disjunctive graph) - это граф, в котором каждая вершина представляет множество дизъюнкций (логических ИЛИ) некоторых элементов. Другими словами, каждая вершина дизъюнктивного графа представляет собой набор элементов, и эти наборы могут пересекаться. Ребра между вершинами графа обозначают, что некоторые элементы могут принадлежать как одному набору, так и другому.

Существуют различные методы представления дизъюнктивных графов, в том числе методы, основанные на булевых матрицах, векторах и формулах. Данные графы обладают рядом интересных свойств, которые делают их полезными инструментами для решения различных задач. Некоторые из этих свойств включают:

1. Степень вершины: степень вершины в графе равна количеству элементов, которые принадлежат этой вершине. Это свойство может быть использовано для эффективного вычисления степеней вершин в дизъюнктивных графах.
2. Клики и независимые множества: в дизъюнктивном графе, максимальная клика (максимальное полное подмножество вершин) является максимальным независимым множеством (максимальным подмножеством вершин, никакие две из которых не соединены ребром). Это свойство может быть использовано для решения задач на графах, связанных с поиском максимальных клик или независимых множеств.
3. Паросочетания: в дизъюнктивном графе, каждое паросочетание (набор ребер, которые не имеют общих вершин) соответствует набору элементов, которые не могут быть выбраны вместе. Это свойство может быть использовано для решения задач на графах, связанных с паросочетаниями.
4. Хроматическое число: хроматическое число дизъюнктивного графа равно максимальному размеру набора элементов, которые не могут быть выбраны вместе. Это свойство может

быть использовано для определения минимального количества цветов, необходимых для раскраски вершин графа.

5. NP-полнота: многие задачи, связанные с дизъюнктивными графами, являются NP-полными, что означает, что нет эффективного алгоритма для их решения. Такие задачи включают в себя нахождение максимальной клики, минимальной вершинной оболочки, максимального независимого множества и др.

Дизъюнктивные графы широко используются в информатике для решения различных задач, таких как анализ социальных сетей, обработка естественного языка, задачи оптимизации, планирования, поиска путей и др. Ниже приведены некоторые примеры использования дизъюнктивных графов:

1. Задача максимальной клики. Максимальной кликой в графе называется подмножество вершин, каждая пара которых соединена ребром, и которое нельзя расширить путем добавления еще одной вершины. Для поиска максимальных клик в графе можно построить дизъюнктивный граф, где вершины соответствуют элементам множества вершин графа, а ребра указывают, какие вершины не могут быть выбраны вместе. Затем можно найти максимальные клики в этом дизъюнктивном графе, которые будут соответствовать максимальным кликам в исходном графе.
2. Задача планирования. При планировании задач на основе дизъюнктивных графов можно представить каждую задачу как вершину графа, а дуги между вершинами обозначают зависимости между задачами. Если две задачи не могут быть выполнены одновременно, то соответствующие вершины графа соединяются ребром в дизъюнктивном графе. Таким образом, можно найти оптимальный план выполнения задач, учитывая ограничения на их выполнение.
3. Задача поиска пути. В графах с дизъюнктивной структурой можно решать задачи поиска пути с ограничениями. Например, если требуется найти кратчайший путь между двумя вершинами графа, при этом некоторые вершины не могут быть посещены одновременно, то можно построить дизъюнктивный граф, где ребра соответствуют возможным переходам между вершинами графа, а ограничения на выбор вершин задаются ребрами дизъюнктивного графа. Затем можно применить алгоритм поиска пути, учитывая ограничения на выбор вершин.

Это лишь некоторые примеры использования дизъюнктивных графов для решения задач на графах. В зависимости от конкретной задачи, структура дизъюнктивного графа может отличаться, но основные идеи и принципы остаются применимыми.

Анализ социальных сетей – это процесс изучения структуры и свойств социальных сетей.

Одним из первых шагов в анализе социальных сетей является описание структуры сети. Структура может быть представлена, например, как число вершин и ребер в сети, а также как графическое представление сети, которое показывает, как вершины и ребра соединены между собой. Структура социальной сети может включать в себя различные характеристики, такие как размер, плотность, среднее расстояние между вершинами и диаметр сети.

Плотность сети – это мера того, насколько много связей в сети находится в сравнении с количеством возможных связей. Чем выше плотность, тем более связанной является сеть. Центральность – это мера того, насколько важными являются отдельные вершины в сети. Одним из способов определения центральности является анализ того, сколько других вершин зависит от данной вершины в сети. Коэффициент кластеризации – это мера того, насколько тесно связаны между собой вершины в небольшой области сети. Кластеризация – это свойство, которое указывает на то, что вершины в сети имеют склонность группироваться в группы или сообщества.

При анализе социальных сетей, дизъюнктивный граф может быть использован для определения наиболее важных узлов или групп узлов, которые обладают определенными свойствами. Он представляет собой граф, в котором каждый узел соответствует индивиду, а ребра соответствуют связям между индивидами. В дизъюнктивном графе ребра могут быть либо присутствующими, либо отсутствующими, в зависимости от того, имеются ли связи между индивидами или нет.

Для определения наиболее важных узлов в социальной сети можно использовать метрики центральности, такие как центральность посредничества (*betweenness centrality*) или центральность близости (*closeness centrality*). Центральность посредничества измеряет количество кратчайших путей, которые проходят через данный узел, что может указывать на то, что узел играет важную роль в передаче информации или контроле за потоками ресурсов. Центральность близости, с другой

стороны, измеряет, насколько близко каждый узел к другим узлам в сети, что может указывать на то, как быстро узел может обмениваться информацией с другими узлами.

Для определения наиболее плотно связанных групп узлов в социальной сети можно использовать алгоритмы выявления сообществ (community detection algorithms). Эти алгоритмы могут использовать различные критерии для определения сообществ, например, на основе плотности связей между узлами внутри сообщества и разреженности связей между узлами в разных сообществах. Выявление сообществ может помочь в определении групп узлов, которые имеют общие интересы, цели или характеристики. Можно использовать методы кластеризации (clustering methods), чтобы определить группы узлов, которые имеют схожие интересы или принадлежат к одной социальной группе. Это может помочь в определении групп, которые могут быть наиболее подвержены определенным воздействиям или иметь схожие поведенческие паттерны.

Использование дизъюнктивного графа для анализа социальных сетей может быть полезным при исследовании различных аспектов социальных взаимодействий и структуры сети. Он может помочь в выявлении наиболее важных узлов и групп узлов в сети, которые могут играть ключевую роль в передаче информации, контроле за потоками ресурсов или формировании сообществ.

Дизъюнктивный граф также может использоваться для анализа влияния узлов в социальной сети. Например, можно использовать алгоритмы распространения влияния (influence propagation algorithms), чтобы определить, какие узлы в сети могут оказывать наибольшее влияние на других узлах. Это может помочь в определении важных точек в сети, которые, если будут заинтересованы в распространении определенной информации, могут привести к широкому распространению этой информации в сети.

Дизъюнктивный граф также может использоваться для анализа изменений в социальной сети во времени. Например, можно создать несколько дизъюнктивных графов, представляющих социальную сеть в разные моменты времени, и сравнить их между собой. Это может помочь в определении тенденций изменения структуры сети, выявлении новых узлов или сообществ и определении причин этих изменений.

Использование дизъюнктивных графов при анализе социальных сетей может быть эффективным методом, так как дизъюнктивные графы могут помочь в моделировании различных аспектов социальных сетей и их свойств.

Например, дизъюнктивные графы могут быть использованы для моделирования неоднородных связей между узлами в социальной сети. В реальных социальных сетях связи между узлами могут иметь различные типы и свойства, и использование дизъюнктивных графов позволяет учитывать эту неоднородность в моделировании.

Дизъюнктивные графы также могут быть использованы для моделирования взаимодействия между различными сообществами в социальной сети. В социальных сетях обычно существуют сообщества, которые имеют сильные связи внутри них, но слабые связи между сообществами. Дизъюнктивные графы могут помочь в моделировании таких сообществ и взаимодействия между ними.

Кроме того, дизъюнктивные графы могут использоваться для моделирования динамики социальной сети. Социальные сети могут меняться со временем, и дизъюнктивные графы могут помочь в моделировании этих изменений и предсказании будущих состояний сети.

В целом, использование дизъюнктивных графов при анализе социальных сетей может быть полезным методом, который позволяет учитывать различные свойства социальных сетей и их сложную структуру. Однако, выбор метода для анализа социальных сетей зависит от конкретной задачи и доступных данных, и может потребовать использования других методов в дополнение к дизъюнктивным графам.

Существует несколько возможных направлений дальнейших исследований в этой области:

1. Развитие методов анализа дизъюнктивных графов для более точного определения наиболее важных узлов и групп узлов в социальных сетях. Это может включать различные подходы для определения критериев важности, таких как мощность, степень влияния и другие.
2. Исследование того, как различные аспекты социальной сети могут влиять на ее структуру и свойства, такие как общность, плотность, структура сообществ и другие. Это может включать исследование различных типов социальных сетей, таких как профессиональные, личные и академические.
3. Исследование того, как дизъюнктивный граф может быть использован для анализа социальных сетей на разных уровнях анализа, таких как индивидуальные узлы, сообщества, группы и т.д. Это может включать исследование взаимодействия между различными уровнями и анализ влияния узлов на различные уровни анализа.

4. Применение дизъюнктивных графов для анализа больших социальных сетей, таких как социальные сети на основе Интернета, чтобы определить наиболее важные узлы и группы узлов, которые обладают определенными свойствами. Это может включать разработку эффективных алгоритмов для обработки и анализа больших объемов данных.

Анализ социальных сетей является активной и интересной областью исследований, и применение дизъюнктивных графов может предоставить новые инструменты для более глубокого и точного анализа этих сетей. Дальнейшие исследования в этой области могут привести к новым инсайтам о социальных сетях и их влияние на общество, а также к развитию более эффективных методов анализа и управления этими сетями.

INVESTIGATION OF THE PROPERTIES OF DISJUNCTIVE GRAPHS AND THEIR APPLICATION AS A TOOL FOR THE ANALYSIS OF SOCIAL NETWORKS

Davidovsky D. V., student gr.253504

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Egorova N.G. - Associate Professor, Candidate of Technical Sciences

Annotation. *The article considers the study of the properties of disjunctive graphs and their application as a tool for the analysis of social networks. A disjunctive graph is a graph where each edge can be either present or absent, depending on whether there is a connection between the vertices. The article describes various metrics used to analyze disjunctive graphs, including centrality, clustering, and network density. A disjunctive graph can be used to identify the most important nodes or groups of nodes in social networks that have certain properties. This article may be useful for researchers engaged in the analysis of social networks and using disjunctive graphs in their research.*

Keywords. *Social network analysis, social network structure, graph representation of social networks, network size, network density, average distance between vertices in the network, network diameter, centrality, individual vertices in the network, clustering coefficient, groups or communities in the network, disjunctive graph, importance of nodes in the social network, centrality metrics, relationships between people*

ПРОГНОЗИРОВАНИЕ ЦЕН АКЦИЙ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

Клебанов Д.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Сиротко С.И. – канд. физ.-мат. наук, доцент

Разработано программное обеспечение для прогнозирования курса акций, использующее в качестве вычислительного механизма сверточные, рекуррентные и гибридные нейросетевые модели: CNN, LSTM, CNN-LSTM.

Применение нейронных сетей в задачах финансового прогнозирования ориентировано на использование свойства нелинейности и самообучения сетей к таким трудно формализуемым задачам, как прогнозирование рынков [1].

Котировки акций представляют собой сложный нестационарный временной ряд, подверженный влиянию большого количества разнообразных факторов, как внутренних, так и внешних. На его динамику оказывают влияние компоненты, выделение и отдельный анализ которых может быть затруднен или невозможен, а их взаимодействие зачастую нивелирует или искажает ключевые закономерности. Проблема точности предсказания цен акций является актуальной и в настоящее время, в связи с чем, при всем многообразии количественных методов анализа, на данный момент не существует общепринятой модели прогнозирования, которая позволила бы получить абсолютно достоверные предсказания.

Для обучения нейронных сетей использовался набор данных цен акций с веб-сайта Yahoo Finance в стабильный период с января 2011 года по январь 2020 года. В качестве типов акций были выбраны акции компаний с наибольшей капитализацией на данный момент – Apple (AAPL), Google (GOOG), Amazon (AMZN), Microsoft (MSFT). Загрузка, обработка и анализ данных осуществлялась с помощью библиотеки Pandas на языке программирования Python.

При реализации нейронных сетей использовалась библиотека Keras для языка Python, основной особенностью которой является встроенная в абстракции слоев нейронных сетей реализация алгоритма обратного распространения ошибки, позволяющая автоматически считать градиенты функции ошибки по отношению к весам слоев нейронных сетей. В качестве алгоритма оптимизации скорости обучения нейронной сети был выбран алгоритм Adam, который использует возможности методов адаптивной скорости обучения, чтобы найти индивидуальную скорость обучения для каждого параметра.

Обучение производилось на 80% исходного набора данных, валидация на 20% исходного набора данных. В качестве гиперпараметров была подобрана вероятность Dropout-а от 0.2 до 0.5 с шагом 0.1 с целью борьбы с переобучением. Также количество эпох было выбрано в пределах 40, а количество примеров при обучении (batch_size), для которых считается функция потерь при работе алгоритма обратного распространения, было выбрано в пределах 16 штук.

Все модели смогли полностью обучиться, однако, модель на основе LSTM смогла это сделать за меньшее число итераций и быстрее, чем модель на основе CNN. В свою очередь медленнее всего сходилась модель на основе CNN-LSTM.

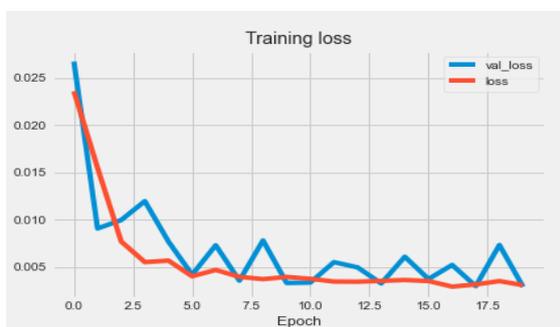


Рисунок 4.3 – Кривая обучения для модели на основе LSTM при вероятности Dropout-а 0.5, размера batch_size 16 сетей на примере акций компании Google

По метрике качества MAE гибридная модель CNN-LSTM позволяет делать прогноз с более высоким качеством.

Список использованных источников:

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

1. Subasi, A. *Stock Market Prediction Using Machine Learning* / A. Subasi, F. Amir, K. Bagedo, A. Shams, A. Sarirete // *Procedia Computer Science*. – № 194. – 2021. – P. 173–179.

СЕКЦИЯ 5
«ФИЗИКА»

ТЕХНОЛОГИЯ RFID. СПОСОБЫ ЗАЩИТЫ ДАННЫХ

Васютин А.Д., Бондарь Е.Н., студенты гр.263001

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Храмович Е.М. – канд. физ.-мат. наук, доцент

Аннотация. Цель работы – изучение технологии радиочастотной идентификации физических объектов, используемой в бесконтактных системах. В работе исследована зависимость скорости чтения RFID-метки от расстояния между RFID-считывателем и RFID-меткой, экспериментально проверены некоторые способы защиты данных RFID-метки от неправомерного считывания. Рассмотрены примеры использования RFID-систем в повседневной жизни.

Ключевые слова. Радиочастотная идентификация, RFID-система, RFID-метка, бесконтактная карта, NFC, защита данных.

Сегодня многие используют бесконтактные карточки для оплаты покупок, проезда в общественном транспорте, пропуска в различные организации. Но как это всё работает? Каким образом считывающее устройство получает информацию с карты? Как передаётся эта информация? Сколько данных можно хранить на такой карте? Можно ли их защитить? Ответы на эти вопросы рассмотрены в нашей работе.

Radio frequency identification (RFID) дословно переводится с английского языка как «радиочастотная идентификация». RFID – это способ автоматической идентификации физических объектов, в котором с помощью радиоволн считываются или записываются данные, хранящиеся в специальных RFID-метках. Технология RFID помогает идентифицировать физический объект. Другие примеры технологии автоматической идентификации: штрих коды, сканеры отпечатков пальцев или сетчатки глаза, идентификация голоса [1].

Система RFID состоит из считывающего/записывающего устройства (ридера) и специальной RFID-метки (транспондера), в которой хранится некоторая записанная в неё информация. Считывающие/записывающие устройства могут быть как мобильными (переносной терминал), так и встроенными в какой-либо объект (например, в дверь или турникет).

RFID-метка всегда состоит как минимум из двух важных компонентов: микрочипа и антенны (катушки). Именно антенна в основном определяет физические характеристики метки. У считывающего/записывающего устройства примерно такой же набор компонентов, но добавляется программное обеспечение для того, чтобы можно было считывать, записывать и изменять информацию в метке. На рисунке 1 показаны основные компоненты RFID-ридера.



Рисунок 1 – RFID-ридер

Транспондеры встраивают в брелоки, идентификационные карты, стикеры, браслеты.

Классифицируют три вида RFID-меток по типу источника питания: пассивные, активные и полупассивные (полуактивные).

Пассивные метки получают энергию для обмена информацией с помощью электромагнитного поля, которое создаёт считывающее/записывающее устройство, и не могут работать без него.

Позволяют вместить информацию размером от одного до сотен бит. Примеры таких меток приведены на рисунке 2.



Рисунок 2 – Примеры пассивных RFID-меток

Активные RFID-метки имеют встроенный источник питания и в большинстве случаев могут передавать данные на большие расстояния (30,5 м и более).

Существуют также полупассивные (полуактивные) RFID-метки. Они имеют внутренний источник питания, но используется он только для выполнения некоторых специализированных задач. В отличие от пассивной метки, полупассивная позволяет передать данные на большее расстояние, так как для этого использует энергию электромагнитного поля RFID-ридера.

Точный предел вмещаемых в метку данных зависит от её физического размера и функциональных возможностей.

Также различают несколько видов транспондеров по типу используемой памяти: только с чтением (read only (RO)), с многократной перезаписью (read and write (RW)), с однократной записью и многократным чтением (write once read many (WORM)). RO-метки могут быть только прочитаны. Информация в них записывается единожды (при изготовлении) и в дальнейшем её изменить нельзя. RW-метки можно считывать и записывать неограниченное количество раз. В WORM-метки информацию пользователь может записать только один раз, а в дальнейшем считывать неограниченное количество раз [1].

В зависимости от диапазона считывания RFID-системы разделяют на системы с близкой связью, системы с удалённой связью и системы дальнего действия. RFID-системы с близкой связью могут обмениваться данными на расстоянии до 10 сантиметров между RFID-меткой и считывающим/записывающим устройством, системы с удалённой связью – до 100 сантиметров, дальнего действия – более 100 сантиметров.

Классифицируют несколько типов RFID-меток в зависимости от частоты, на которой они передают сигнал: низкочастотные (НЧ), высокочастотные (ВЧ) и ультравысокочастотные (УВЧ).

Низкочастотные RFID-метки работают на частоте 125 - 134 кГц и характеризуются низкой скоростью и небольшим расстоянием передачи данных. Относятся к RFID-системам с близкой связью и RFID-системам с удалённой связью. Преимущество низкочастотных транспондеров перед другими типами меток заключается в том, что они устойчивы к жидкостям и металлам, а это значит, что их можно использовать в различных рабочих средах.

Высокочастотные RFID-метки работают на частоте 13,56 МГц, что даёт им высокую скорость передачи информации, но расстояние, на которое её можно передать, всё равно остаётся довольно небольшим. Относятся к RFID-системам с близкой связью и RFID-системам с удалённой связью. Имеют хорошие рабочие характеристики для считывания в присутствии металлов и жидкостей. Так как работа на данной частоте не требует большого числа мотков катушки внутри RFID-метки, то транспондер становится более компактным и удобным для использования.

Ультравысокочастотные транспондеры работают на частотных полосах от 300 МГц до 1 ГГц и отличаются очень высокой скоростью и большим расстоянием передачи данных. Относятся к RFID-системам дальнего действия. Недостатком является снижение характеристик работы RFID-системы при использовании в присутствии металлов и жидкостей. В настоящее время данный тип меток не сильно распространён, так как он не принят в мировом масштабе [1].

В таблице 1 мы представили классификацию RFID-меток в зависимости от рабочей частоты.

Таблица 1 – Классификация RFID-меток в зависимости от частоты, на которой передаётся сигнал.

Тип RFID-метки	Рабочая частота	Расстояние считывания
НЧ	125 – 134 кГц	до 100 см
ВЧ	13,56 МГц	до 100 см
УВЧ	300 МГц – 1 ГГц	более 100 см

Самыми широко используемыми типами транспондеров являются низкочастотный и высокочастотный.

Рассмотрим несколько примеров использования RFID-систем в повседневной жизни.

В магазине на товары размещают специальные RFID-метки, в которых записана информация о том, что товар ещё не приобретён. На кассе товар размагничивают и удаляют из метки эту информацию. В случае, если неоплаченный товар попробуют вынести из магазина, начнёт «пищать» турникет, так как обнаружит в метке запись о том, что товар ещё не приобретён.

В общественном транспорте Минска можно оплатить проезд с помощью бесконтактной смарт-карты (БСК). При пополнении карты поездками, на неё записывается определённая информация. Когда в транспортном средстве БСК подносится к валидатору, он считывает с неё информацию. Происходит оплата проезда [2].

При оплате покупок в магазине покупатель прикладывает свою банковскую карту к платёжному терминалу, который идентифицирует карту и проводит банковскую операцию.

Банковские системы создали свои технологии, которые основываются на технологии RFID: Visa PayWave и Mastercard PayPass [3].

На некоторых автомобильных дорогах необходимо оплачивать проезд. Для этого в транспортное средство на лобовое стекло крепится специальная метка. Когда транспортное средство проезжает по дороге возле специального считывающего устройства, происходит обмен данными и оплата проезда.

Принцип работы RFID-системы заключается в том, что считывающее/записывающее устройство создаёт электромагнитное поле. В RFID-метке под воздействием этого поля возникает индукционный ток, с помощью которого запитывается микрочип и отправляет или принимает информацию от считывающего/записывающего устройства [1].

В процессе развития технологий у высокочастотного типа RFID-систем появился специализированный подвид: технология Near Field Communication (NFC). Главным её отличием от высокочастотных RFID-систем стал режим P2P, с помощью которого два устройства могут обмениваться информацией между собой. Ранее такая функция была недоступна [4].

Модуль NFC устанавливают во многие модели смартфонов. Он позволяет использовать смартфон вместо банковской карты или пропуска. Данный модуль даёт возможность смартфону выполнять роль как считывающего/записывающего устройства, так и самой RFID/NFC-метки. Обычно производители располагают такой модуль как можно ближе к задней крышке смартфона. Связь NFC имеет небольшой радиус действия, поэтому необходимо подносить карту или смартфон как можно ближе к считывающему/записывающему устройству. NFC также можно использовать для быстрого подключения внешних устройств: NFC позволяет прописывать в метке некоторые последовательности действий (сценарии), которые выполняются при её считывании [5].

Данные, в соответствии с технологией RFID, передаются без проводов, поэтому следует рассмотреть вопрос о безопасности её использования: могут ли злоумышленники похитить ценную информацию? Как её защитить?

Мы провели несколько экспериментов с RFID-системой, работающей на частоте 13,56 МГц. Использовали RFID-метку и модуль RFID-RC522, предварительно подключив его к микроконтроллеру Arduino. Состав RFID-системы представлен на рисунке 3.

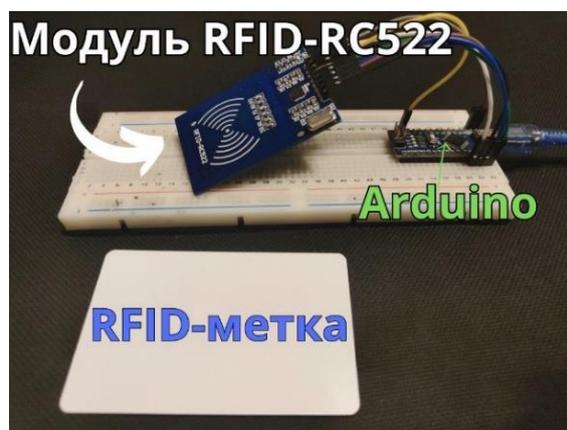


Рисунок 3 – RFID-система

Мы исследовали зависимость скорости чтения RFID-метки от расстояния между считывающим устройством и этой меткой. В результате получили закономерность: чем больше расстояние между меткой и считывающим устройством, тем меньше скорость передачи

информации. В нашем случае метка считывалась на расстоянии до 5 сантиметров между меткой и считывающим устройством. При увеличении этого расстояния устройство не обнаруживало метку.

Мы также провели эксперимент сразу с двумя RFID-метками. Две метки сложили вместе и поднесли к считывающему устройству. В результате RFID-модуль считал только одну из карт, при этом время считывания незначительно увеличилось.

Мы попробовали клонировать данные с одной RFID-метки на другую. Может быть метка не позволит себя клонировать? К сожалению, нет. Считывание прошло быстро и не вызвало каких-либо трудностей.

Были сделаны предварительные выводы:

- 1) нужно поместить что-то между меткой и считывателем, чтобы её не получилось считать;
- 2) нужно зашифровать данные в метке, чтобы, даже если данные смогут похитить, никто ими не смог воспользоваться.

Мы проверили первое предположение. Многие носят бесконтактные карты в карманах одежды и сумок. Может быть, ткань не позволит считать метку? Поместили ткань между RFID-меткой и считывающим устройством. Попытались считать метку. Метка считалась без каких-либо трудностей. Может быть нужно использовать другой материал? Работа RFID-системы основана на принципе электромагнитной индукции, поэтому возникло предположение, что если обмотать метку фольгой (поместить между меткой и считывающим устройством тонкий слой металла), то её невозможно будет считать. Мы обмотали метку фольгой и поднесли максимально близко к считывателю. В итоге метку прочесть так и не удалось. А это значит, что хорошим способом защиты RFID-метки от считывания злоумышленниками может быть размещение метки, когда она не используется, в защитной оболочке или футляре, имеющим в составе тонкий слой металла. Мы провели эксперимент ещё раз, поместив между меткой и считывающим устройством тонкую металлическую пластину. Получили ожидаемый результат: метку считать не удалось.

Второе предположение будем считать верным. Если зашифровать данные в метке, то после её считывания понять информацию не получится без знания применённого алгоритма шифрования.

На сегодняшний день существует множество различных способов, с помощью которых можно обеспечить защиту бесконтактных карт или меток. Например, с помощью тонкого слоя металла можно заблокировать передачу данных по RFID-технологии. В качестве иного способа защиты можно носить несколько бесконтактных карт вместе. При считывании информации RFID-ридер будет получать отклики от разных микрочипов, что значительно затруднит получение информации. Однако данный способ сработает только в случае терминалов, а специальные программные коды легко его обойдут. Но носить обернутую фольгой карту или несколько бесконтактных карт вместе непрактично. Лучше использовать специальные приспособления, такие как экранирующие бумажники, кошельки и чехлы. Они более надёжные и удобные в использовании.

А как же защитить данные карты, привязанной к модулю NFC смартфона? Может быть, стоит отключать модуль NFC, когда он не используется? Отключение модуля NFC в смартфоне не является обязательным. Однако, если вы не пользуетесь данной технологией, её рекомендуется отключить, так как это минимизирует вероятность кражи информации мошенниками [5].

Таким образом, технология RFID получила широкое распространение в нашей повседневной жизни. Возможно, в будущем данная технология будет активно развиваться и появятся новые методы защиты данных. В скором времени могут получить широкое распространение умные RFID-системы, взаимодействующие с искусственным интеллектом и другими датчиками. Для нас данная тема представляет большой интерес с профессиональной точки зрения.

Список использованных источников:

1. Сандип Лахири *RFID. Руководство по внедрению* / Дудников С. – М.: Кудиц-Пресс, 2007. – 312 с.
2. ОПЛАТА ПРОЕЗДА С ПОМОЩЬЮ БСК. [Электронный ресурс]. – Режим доступа: <https://minsktrans.by/oplata-proezda/oplata-proezda-bsk/>. – Дата доступа: 02.04.2023.
3. Что такое бесконтактные банковские карты PayPass и payWave. [Электронный ресурс]. – Режим доступа: <https://nfcexpert.ru/takoe-payave-paypass>. – Дата доступа: 02.04.2023.
4. Что такое RFID? Описание RFID технологии, принципа работы и сфер применения. [Электронный ресурс]. – Режим доступа: <https://nfcexpert.ru/cto-takoe-rfid-opisanie-rfid-tehnologii-printsipa-raboty-i-sfer-primeneniya>. – Дата доступа: 02.04.2023.
5. NFC в смартфоне: что это такое, зачем нужно и как настроить бесконтактную оплату. [Электронный ресурс]. – Режим доступа: <https://www.samsung.com/ru/explore/life-hacks/nfc-in-a-smartphone-what-is-it-why-is-it-needed-and-how-to-set-up-contactless-payment/>. – Дата доступа: 02.04.2023.

RFID TECHNOLOGY. DATA PROTECTION METHODS

Vasiutich A.D., Bondar E.N.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Khramovich E.M. – PhD in Physics and Mathematics

Annotation. The purpose of the work is to study the technology of radio frequency identification of physical objects used in contactless systems. The paper investigates the dependence of the RFID tag reading speed on the distance between the RFID reader and the RFID tag, experimentally tested some ways to protect RFID tag data from unauthorized reading. Examples of the use of RFID systems in everyday life are considered.

Keywords. Radio frequency identification, RFID system, RFID tag, Contactless card, NFC, Data protection.

УДК 621.3.049.77–048.24:537.2

ОПТИМИЗАЦИЯ РАБОТЫ УЛЬТРАЗВУКОВОЙ ОЧИСТИТЕЛЬНОЙ ВАННЫ

Родевич Р.А., Шебеко В.П.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Савилова Ю.И.—доцент кафедры физики

Аннотация. В ходе работы были проведены опыты с использованием кавитометра, датчика и анализатора спектра. Были установлены зависимости активности кавитации от уровня жидкости, температуры и содержания газа.

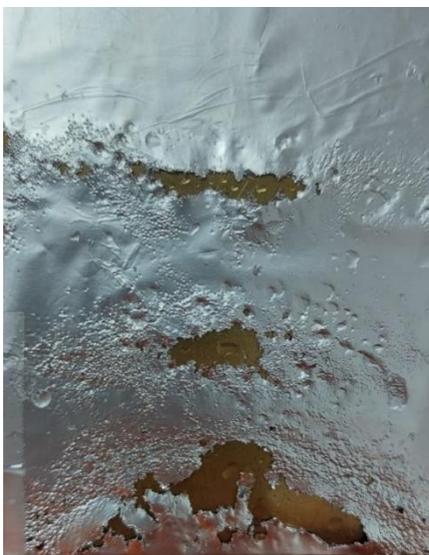
Ключевые слова: кавитация, активность, ультразвуковые волны, ультразвуковая ванна.

Введение.

Кавитация – процесс образования пузырьков в жидкостях и их последующее схлопывание, при котором высвобождается большое количество энергии и другие физические процессы. Сам процесс сопровождается гидроударами и характерным шумом. Кавитация возникает в результате местного понижения давления в жидкости, которое может происходить либо при увеличении скорости, (гидродинамическая кавитация), либо при прохождении акустической волны большой интенсивности во время полупериода разрежения (акустическая кавитация).

Принцип кавитации применяется в многих областях медицины таких как нейрохирургия, офтальмология, онкология, отоларингология, кардиология и урология^[1]. Также в промышленных целях применяется очистка при помощи ультразвуковой ванны. При очистке таким способом требуется правильно поместить деталь в ванну. При неправильном размещении детали есть вероятность неравномерной очистки или разрушения детали(рисунок 1).

Рисунок 1



Основная часть.

Для регистрации кавитации используются специализированное оборудование. К подобному можно отнести кавитометры, специализированные датчики и анализаторы спектра.

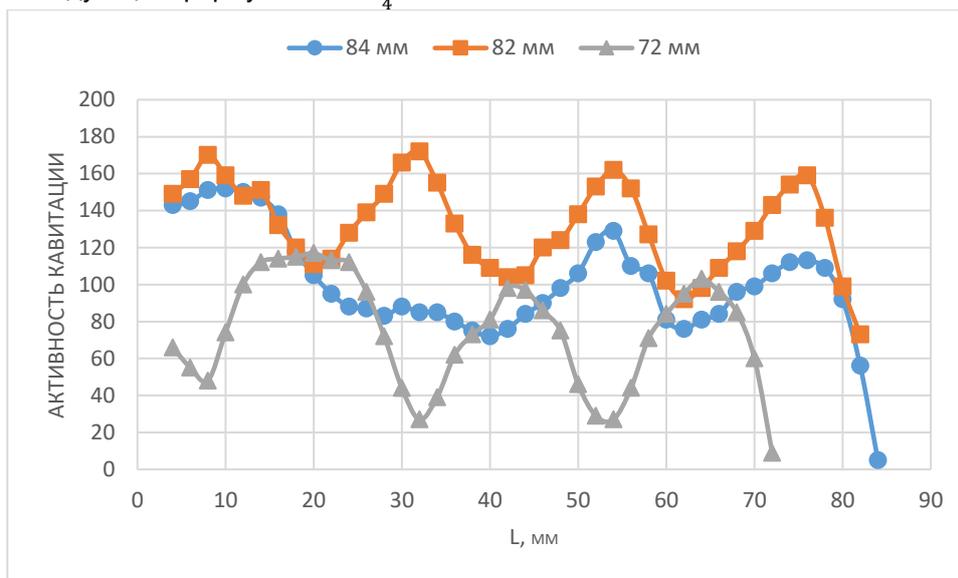
Принцип работы ультразвуковой ванны следующий: пьезоэлектрические излучатели преобразует электрические колебания, вырабатываемые транзисторным генератором, в упругие механические колебания соответствующей частоты. Через плоское дно ванны колебания передаются в рабочую среду. Воздействие на физико-химические процессы в жидкостях осуществляется в основном за счет вторичных акустических эффектов – кавитации и акустических течений.

Ванна состоит из емкости в которую ко дну монтируется держатель преобразователя. В нем предварительно установлен преобразователь в этом случае происходит непосредственный контакт преобразователя с жидкостью.

На распределение кавитации влияют несколько основных факторов: уровень жидкости, температура и газосодержание.

Уровень жидкости:

На рисунке 2 представлены зависимости активности кавитации от расстояния L между датчиком и излучателем для разных уровней жидкости h в ванне. Из представленных данных видно, что при изменении уровня жидкости существенно меняется активность кавитации и характер её распределения в объёме жидкости. Это обусловлено скорее всего изменением условий образований стоячей компоненты звукового поля. Также меняется звук и невооружённым взглядом видно изменение на поверхности воды. При любом уровне можно наблюдать эффект стоячей волны. При расстоянии 82 мм наблюдается самая большая активность кавитации. Это обусловлено тем что для данного h условия образования стоячей волны наиболее близки к идеальным, определяемым следующей формулой $h = \lambda \frac{\lambda}{4}$

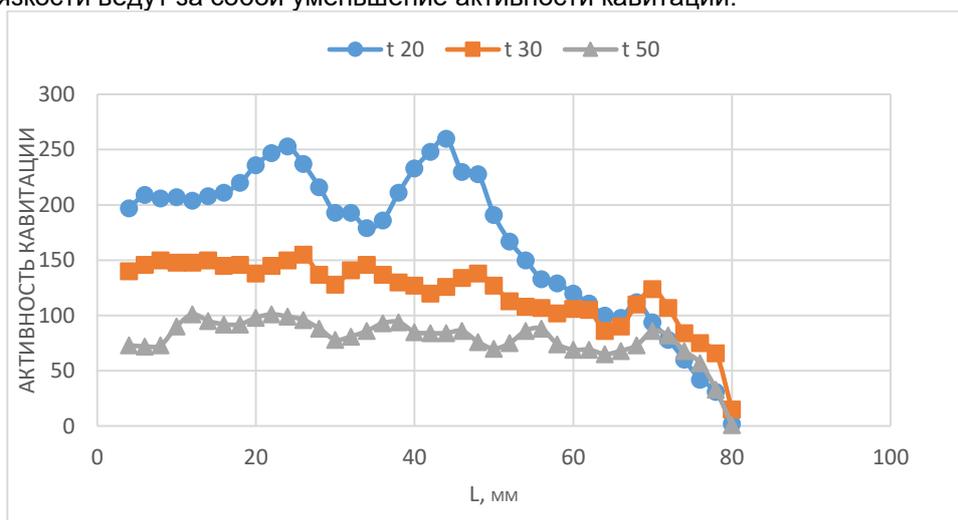


Уровень жидкости в ванне, мм: 1(●)—84 ; 2(■)—82; 3(▲)—72

Рисунок 2 – Распределение активности кавитации вдоль оси излучателя

Температура:

При изменении температуры повышается плотность вязкость жидкости. Увеличение плотности и вязкости ведут за собой уменьшение активности кавитации.



Температура жидкости в ванне, С°: 1(●)—20 ; 2(■)—30; 3(▲)—50

Рисунок 3—изменение активность кавитации в зависимости от температуры

Газосодержание:

Для исследования зависимости содержания газа в жидкости от распределения мы использовали минеральную воду.

На графике представлен изменение активности кавитации в процессе дегазации минеральной воды. В самом начале опыта наблюдается выделение большого количества пузырьков газа (рисунок 2). Примерно на 272 и 350 секундах происходят изменения.

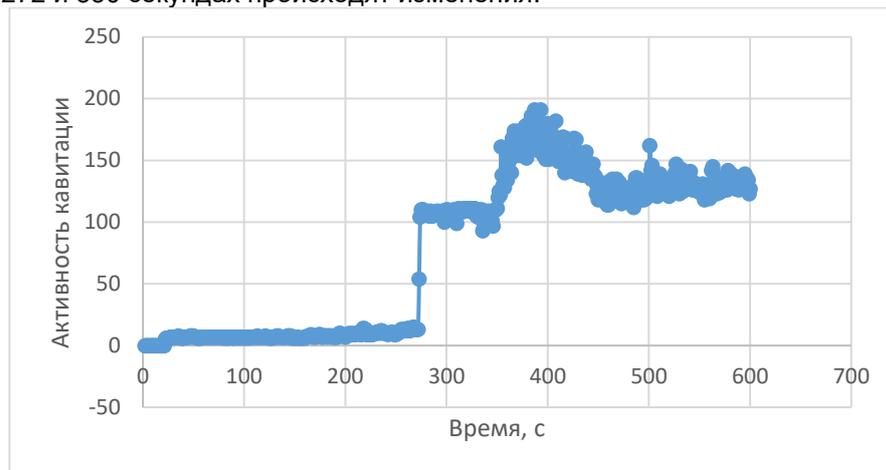


Рисунок 4 –изменение активности кавитации в зависимости от содержания газа

Заключение.

Выполнено исследование влияния температуры жидкости, ее уровня в реакторном стакане и дегазации на активность кавитации *A* в неоднородном ультразвуковом поле.

Показано, что при уровнях жидкости, наиболее близко соответствующих идеальным условиям формирования стоячей волны, распределение активности кавитации вдоль оси излучателя представляет собой чередующиеся максимумы и минимумы, расстояние между которыми примерно равно четверти длины волны ультразвука на данной частоте

При увеличении температуры жидкости активность кавитации уменьшается, меняется также характер ее распределение в объеме жидкости. При повышении температуры максимумы и минимумы проявляются все слабее, т.е. распределение становится более однородным.

В экспериментах с водой, пересыщенной углекислым газом, показано, что по мере дегазации, т.е. при уменьшении газосодержания активность кавитации повышается. Выделено две стадии дегазации: переход от одной стадии к другой сопровождается скачкообразным повышением активности кавитации.

Полученные результаты будут использоваться при оптимизации ультразвуковых технологических процессов в жидкости.

Список использованных источников:

1. Л.Р. Гаврилов **ФОКУСИРОВАННЫЙ УЛЬТРАЗВУК ВЫСОКОЙ ИНТЕНСИВНОСТИ В МЕДИЦИНЕ.**
2. **КАВИТАЦИЯ В ВОДНЫХ РАСТВОРАХ УГЛЕКИСЛОГО ГАЗА** А.В. КОТУХОВ, Н.А. ЖАРКО, В.С. МИНЧУК, Н.В. ДЕЖКУНОВ.

OPTIMIZING THE PERFORMANCE OF THE ULTRASONIC CLEANING BATH

Rodevich R. A., Shabeka V.P.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Savilova J.I. – assistant professor of the Department of Physics

Annotation. . In the course of the work, experiments were carried out using a cavitometer, a sensor and a spectrum analyzer. The dependences of cavitation activity on the liquid level, temperature and gas content were established.

Keywords. cavitation, activity, ultrasonic waves, ultrasonic bath.

НОВЫЕ МАТЕРИАЛЫ В НАНОЭЛЕКТРОНИКЕ

Абушкевич А.А., Чапля М.Д., Агель А.А., ст. группы 221702

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Родин С.В. – доцент, канд. физ.-мат. наук, доцент кафедры физики

Аннотация. В данной работе представлены материалы, которые применяются в нанoeлектронике, их краткое описание, область применения и перспективы внедрения и развития.

Ключевые слова. Материалы, нанoeлектроника, материалы в нанoeлектронике.

Во второй половине прошлого века электроника прочно вошла в нашу повседневную жизнь (мобильные телефоны, персональные компьютеры, калькуляторы, системы управления двигателем автомобиля), и теперь сложно себе представить день, прожитый без электронных устройств. И даже день отдыха на природе вдали от цивилизации многие проводят с электронным спутниковым навигатором в кармане.

Стремительное развитие микроэлектроники и ее концептуальная трансформация в нанoeлектронику в 1999 году, когда в промышленном производстве был преодолен рубеж минимальных размеров элементов интегральных схем в 100 нм, означает не просто уменьшение размеров интегральных полупроводниковых структур, но также возникновение новых физических эффектов в них. Это потребовало огромных усилий в области интеграции новых материалов в наноструктуры ИС, которые нарастали по мере продвижения в область нано- размеров.

Говоря об используемых материалах, стоит упомянуть, что доминирование кремния в микроэлектронике обуславливалось не только его уникальными свойствами как полупроводника, но и замечательными диэлектрическими свойствами его оксида SiO_2 , а также нитрида. Поэтому чип интегральной схемы, с точки зрения применяемых материалов, ранее можно было охарактеризовать всего несколькими словами: Si, SiO_2 , Si_3N_4 плюс легирующие примеси, как правило, бор/фосфор, а также Al в качестве проводников схемы.

Нанoeлектроника - это область электроники, которая занимается разработкой и использованием наноматериалов и наноструктур (тонких слоев, кластеров, кубитов и т.д.) для создания электронных компонентов и устройств с нанометровыми размерами и новыми электронными свойствами. Это включает в себя разработку нанотехнологий для создания более производительных, меньших и энергоэффективных устройств, таких как микропроцессоры, флэш-накопители, высокочастотные компоненты, датчики, биомедицинские устройства и прочее.

Фактически, нанoeлектроника является логическим развитием микроэлектроники. Таким образом, с одной стороны, нанoeлектроника является неким продолжением развития традиционных идей микроэлектроники, связанным со схемотехническими методами обработки информации. С другой стороны, совершенно новые идеи в квантовых явлениях и эффектах открывают совершенно новые возможности создания приборов нового поколения.

Развитию нанотехнологий способствует постоянное стремление уменьшить составляющие элементы полупроводниковых приборов и интегральных схем. Основным направлением являются новые уникальные свойства материалов и структур, создаваемых по нанотехнологиям. При этом нанотехнологические приемы постоянно совершенствуются, появляются новые методы.

В нанoeлектронике используются различные материалы, которые могут иметь нанометровые размеры и обладать особыми электронными свойствами. Некоторые из них:

1. Графен: это плоский слой углерода толщиной в один атом, обладающий высокой электропроводностью и светопропусканием. Графен является одним из самых обещающих материалов для создания электронных компонентов, таких как транзисторы и сенсоры. Кроме того, он может использоваться в качестве покрытия для повышения прочности и стойкости к износу.

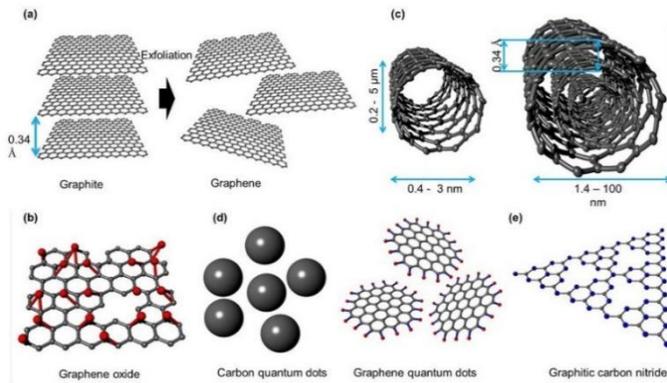


Рисунок 1- графен

2. Углеродные нанотрубки: цилиндрические структуры, образованные из углеродных атомов, с высокой прочностью и электропроводностью. Нанотрубки могут использоваться в качестве электродных материалов для литий-ионных батарей, а также для создания ультрафильтров и мембран для различных промышленных приложений.

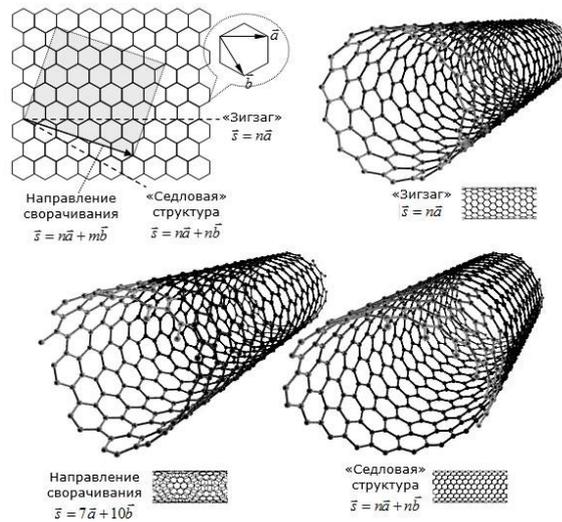


Рисунок 2- углеродные трубки

3. Нанокристаллы: кристаллы, размер которых составляет несколько нанометров, обладают уникальными электронными свойствами и могут использоваться в качестве чувствительных элементов для различных типов датчиков. На рисунке 3 представлены нанокристаллы некоторых веществ.

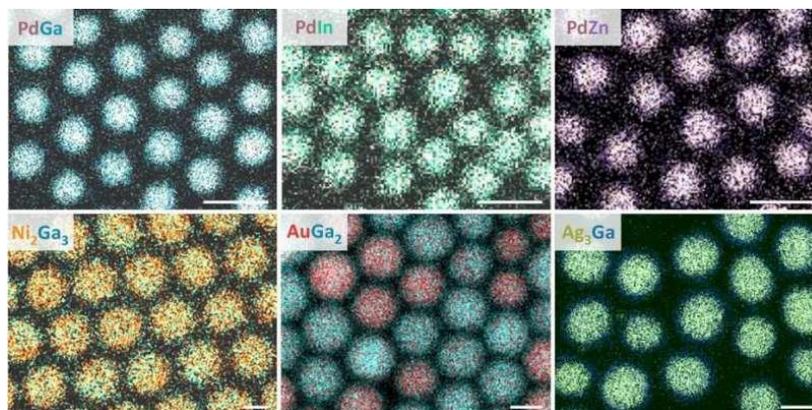


Рисунок 3- нанокристаллы из некоторых веществ

4. Квантовые точки: наногабаритные полупроводниковые частицы с размерами от нескольких до нескольких десятков нанометров, имеют спектральные свойства и применяются в оптике. Квантовые точки могут использоваться для создания светодиодов и солнечных батарей с высокой эффективностью.

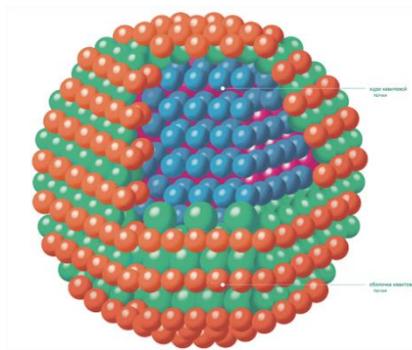


Рисунок 4- квантовая точка

5. Нанопроволоки: прямоугольные или цилиндрические структуры, состоящие из металлов, полупроводников или диэлектриков, с размерами несколько нанометров. Нанопроволоки могут использоваться для создания эффективных сенсорных систем, биомедицинских приборов и нанокomпьютеров.

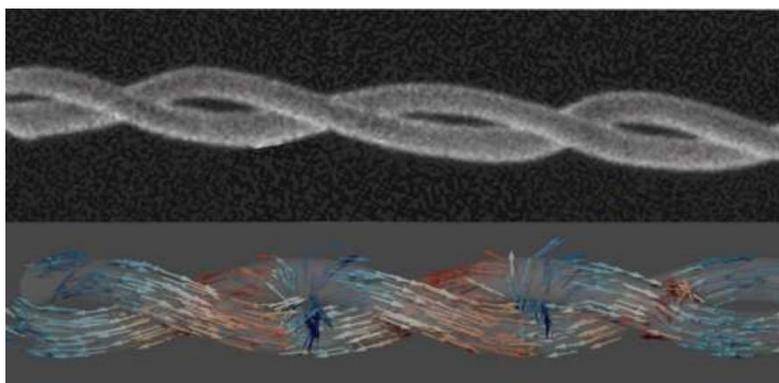


Рисунок 5- нанопроволока

6. Металлические наночастицы: частицы металлов, размер которых составляет до нескольких десятков нанометров, имеют особые оптические и магнитные свойства. Металлические наночастицы могут использоваться для создания плазмонных устройств, улучшающих эффективность процессов передачи данных, и наномантенн для облегчения работы беспроводных сетей. На рисунке 6 представлены наночастицы некоторых металлов, таких как золото, серебро, медь, кобальт, никель и др.

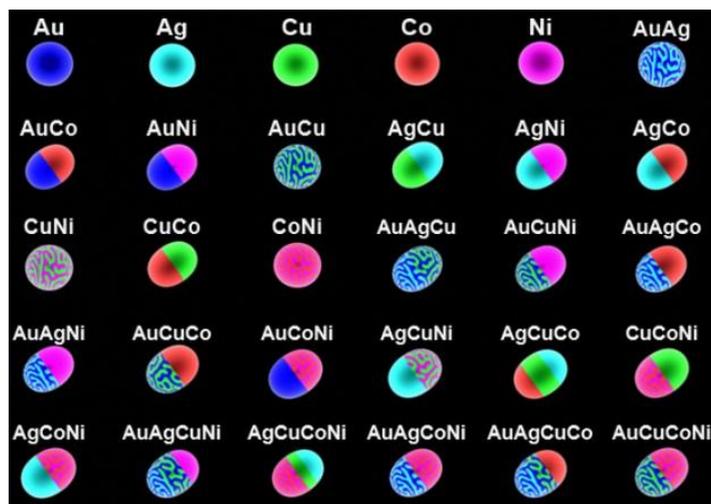


Рисунок 6- металлические наночастицы

На основании проведенного анализа можно сделать вывод о перспективности использования графена, углерода, полупроводников и диэлектриков в наноэлектронике, наноплазмонике и других применениях.

В заключении, переход к наноэлектронике в определенной степени базируется на достижениях микроэлектроники – использование уменьшающихся до атомарных размеров транзисторов и диодов и собранных из них схем. В то же время будущее сулит новые достижения на основе новых принципов работы на уровне отдельных атомов – использование квантовых эффектов, волновых свойств электрона и других явлений наномира.

В настоящее время уже создаются заделы будущих устройств и технологий. Перспективными в этом отношении являются различные объекты: углеродные (а возможно и другие) нанотрубки, супрамолекулярные соединения. Вероятней всего, что приборы следующих поколений будут гибридами, построенными из фрагментов применяемых в нынешней полупроводниковой электронике и новых элементов. Очевидно, что преобладающим подходом будет создание устройств путем синтеза из атомов и молекул (down up way).

Список использованных источников:

1. ИССЛЕДОВАНИЕ ХАРАКТЕРИСТИК УЛЬТРАТОНКИХ ПЛЁНОК MoS₂ ДЛЯ НАНОЭЛЕКТРОНИКИ [Электронный ресурс]. –: <https://studvesna.ru/>
2. НАНОТЕХНОЛОГИИ В СОВРЕМЕННОМ МИРЕ [Электронный ресурс]. <https://core.ac.uk/download/pdf/328379709.pdf#page=413>
- 3 СИНТЕЗ И ИССЛЕДОВАНИЕ НОВЫХ МАТЕРИАЛОВ В МДПСТРУКТУРАХ ДЛЯ РАЗРАБОТКИ ФИЗИЧЕСКИХ ОСНОВ КМОПТЕХНОЛОГИЙ НАНОЭЛЕКТРОНИКИ [Электронный ресурс]. – А. В. Зенкевич, Ю. Ю. Лебединский, Ю. А. Матвеев, Н. С. Баранцев, Ю. А. Воронов, А. В. Сокоян, В. Н. Неволин, В. И. Чичков¹, С. Спига², М. Фанчулли, 2010г.

UDC 538.945

NEW MATERIALS IN NANOELECTRONICS

Abushkevich A.A., Chaplya M.D., Agel A.A., et. группы 221702

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Rodin S.V – assistant professor, PhD in Physics and Mathematics, associate professor of the department of physics

Annotation. This paper presents materials that are used in nanoelectronics, their brief description, scope of application and prospects for implementation and development

Keywords.. Materials, Nanoelectronics, Materials in Nanoelectronic.

ЧИСЛЕННОЕ МОДЕЛИРОВАНИЕ ЭЛЕКТРОМАГНИТНЫХ ПОЛЕЙ

Бушенко В. А., студент гр. 272301, Демидовец В. В., студент гр.272302

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Смирнова Г. Ф. – доцент, канд. физ.-мат. наук, доцент кафедры физики

Аннотация. В данной работе рассмотрены методы численного моделирования электромагнитных полей, а также программы, используемые для расчетов в данной области.

Ключевые слова. Метод конечных элементов, метод конечных разностей, автоматизация численного моделирования, электромагнитные поля.

Электромагнитное поле — это важное понятие во многих областях науки и техники, начиная от электроники и заканчивая мощными электростанциями. Оно представляет собой особый вид материи, посредством которой осуществляется взаимодействие между электрическими заряженными частицами.

Создание точных математических моделей электромагнитных полей является непростой задачей, требующей учета множества факторов, влияющих на эти поля. Численное моделирование электромагнитных полей можно считать необходимым инструментом для проектирования и оптимизации различных устройств и систем.

Одним из главных преимуществ численного моделирования является возможность учитывать множество факторов, влияющих на электромагнитные поля, и получать результаты с высокой точностью.

В данной работе рассмотрены методы численного моделирования электромагнитных полей, а также программы, используемые для расчетов в данной области. Вначале будет обзор различных методов моделирования, таких как метод конечных элементов, метод конечных разностей, метод конечных объемов и другие. Будут проанализированы их особенности, преимущества и ограничения. Далее будут рассмотрены программы, используемые для расчетов в данной области. Будут представлены примеры таких программ, как CST Studio Suite, Ansys HFSS. Кроме того, будет описан процесс автоматизации расчетов, который позволяет ускорить процесс моделирования и сделать его более точным и эффективным.

Численное моделирование электромагнитных полей является мощным инструментом для различных областей науки и техники, и его преимущества, такие как высокая точность и возможность автоматизации, делают его неотъемлемой частью современного инженерного процесса.

Существует несколько способов решения задач электромагнитного, теплового, прочностного и других типов анализа. Первый представляет собой аналитический метод. Его целесообразно применять в том случае, если геометрия системы достаточно проста, т. к. он включает себя упрощенные аналитические приближения с множеством ограничений и низкой точностью. Поэтому в последнее время все более часто приходит ему на смену второй способ – высокоэффективные численные методы. Каждый из этих методов имеет определенные преимущества и может использоваться один или совместно с другими[1].

Классификация этих методов включает в себя:

- метод сеток (метод конечных разностей);
- метод конечных элементов;
- метод конечных объемов;
- метод моментов;
- метод граничных уравнений;
- интегрально-дифференциальный метод.

Также отдельно можно сказать про гибридный. Суть его в том, что он включает в себя несколько методов.

Рассмотрим подробнее некоторые из методов.

Метод конечных разностей или, как часто его называют, метод сеток – наиболее старый и относительно простой метод, поскольку требует минимальной математической работы. МКР довольно успешно применялся и применяется в настоящее время для решения широкого круга расчета электромагнитных и тепловых полей. Идея метода конечных разностей (метода сеток) известна давно, с соответствующих трудов Эйлера. Однако практическое применение этого метода было тогда весьма ограничено из-за огромного объема ручных вычислений, связанных с

размерностью получаемых систем алгебраических уравнений, на решение которых требовались годы. В настоящее время, с появлением быстродействующих компьютеров, ситуация в корне изменилась. Этот метод стал удобен для практического использования и является одним из наиболее эффективных при решении различных задач математической физики.

Общая идея МКР состоит в том, что мы сводим исходную задачу к более простой задаче решения систем линейных уравнений или нелинейных алгебраических уравнений. Вид получаемой системы зависит от вида исходного дифференциального уравнения.

МКР в общем случае включает в себя три этапа:

1. Этап построения сетки. На плоскости в заданной плоскости строится сеточная область, состоящая из одинаковых ячеек определенного размера.

2. Этап, при котором исходное дифференциальное уравнение системы заменяется соответствующим конечно-разностным уравнением, описывающим их функциональные связи между соседними узлами сетки;

3. Этап решения системы из конечно-разностных с n неизвестными одним из численных методов.

Число неизвестных n (порядок решений системы) соответствует числу узловых значений, в которых определяется значение искомой функции. Выбор сеточной области зависит от конкретной задачи, но при решении мы стремимся к тому, чтобы контур сеточной области наилучшим способом аппроксимировал контур заданной области.

К недостаткам метода можно причислить сложность действий с матрицами, а также сложность описания криволинейных поверхностей тел из-за использования прямоугольной конечно-разностной сетки.

Идея метода конечных элементов (МКЭ) возникла в СССР в 1936 году, но из-за того, что вычислительная техника была недостаточно развита, был впервые применен только в 1944 году. Достоинствами этого метода являются: точность вычислений, возможность описания криволинейных границ области любой сложности, легкость учета граничных условий различных типов[2].

Основная идея МКЭ заключается в том, что любую непрерывную величину можно аппроксимировать дискретной моделью, которая строится на множестве кусочно-непрерывных функций, определенных на конечном числе подобластей. Кусочно-непрерывные функции определяются с помощью значений непрерывной величины в конечном числе точек рассматриваемой области.

При построении дискретной модели необходимо поступать следующим образом:

В рассматриваемой области фиксируется конечное количество точек (узлов);

Значение непрерывной функции в каждой точке считается переменной, которую необходимо определить;

Область определения непрерывной величины разбивается на конечное число подобластей (элементов). Эти элементы имеют общие узлы и в совокупности аппроксимируют форму области;

Непрерывная величина аппроксимируется на каждом элементе полиномом, который определяется с помощью узлов этой величины. Для каждого элемента определяется свой полином. Они подбираются таким образом, чтобы сохранилась непрерывность величины вдоль границ элемента.

МКР и МКЭ являются сеточными методами. Они обладают примерно равными возможностями с точки зрения теоретических оценок точности. Однако между этими методами есть и существенные отличия. В первую очередь, в МКР аппроксимируются производные искомых функций, а в МКЭ – зависимость исходных функций от пространственных координат и времени. В МКР строятся регулярные сетки и особенности геометрии области учитываются только в пограничных узлах, поэтому метод применяется для задач с простой геометрией. В МКЭ при разбиении на элементы учитываются геометрические особенности области. У этих методов так же есть общая проблема: высокая размерность результирующей системы алгебраических уравнений, которая частично решается с использованием высокопроизводительных ЭВМ.

Численные методы моделирования были разработаны задолго до создания ЭВМ. Тогда все расчеты велись вручную. Однако ритм разработки все время увеличивается, поэтому появилась необходимость в ускорении процесса численного моделирования. Следующим шагом к автоматизации стала разработка систем автоматизированного проектирования и пакетов прикладных программ. Все эти системы были созданы для определенных предметных областей, в первую очередь наукоемких, чтобы уменьшить время, затрачиваемое на моделирование. Однако подобные пакеты создавались годами, так как их разработка требовала знаний в предметной области, численном моделировании и программировании. При этом процесс построения модели оказался наиболее сложным для автоматизации.

Построение модели вручную является трудоемким процессом, поскольку придется столкнуться с огромным количеством различных преобразований и существует вероятность

запутаться и допустить ошибку. Использование систем автоматизированного моделирования позволяет увеличить производительность труда, уменьшить количество ошибок и затрачиваемого времени. При этом исчезает необходимость привлечения программистов для этой работы, так как используя системы автоматизированного моделирования специалист в конкретной предметной области может самостоятельно построить достаточно сложные модели[3].

Существует огромное количество методов численного моделирования электромагнитных полей. Все они разрабатывались десятилетиями и сейчас известны и применяются во всем мире. Однако в современном мире происходит повышение требований к качеству и скорости разработки электрических аппаратов и снижение требований к производственным затратам. Поэтому сейчас широкое распространение получают программные пакеты, которые позволяют оптимизировать процесс численного моделирования.

CST Studio Suite – пакет программного обеспечения, который используется для проектирования, анализа и оптимизации электромагнитных систем. Он применяется в ведущих инженерных компаниях по всему миру. Единый пользовательский интерфейс делает доступными программы для всего спектра электромагнитных полей. Благодаря этому создается возможность эффективно анализировать системы, состоящие из большого количества компонентов. CST Studio Suite позволяет оптимизировать производительность устройства, выявлять и устранять на ранних этапах проектирования несоответствия требованиям, свести к минимуму риск ошибок.

Ansys HFSS – программное обеспечение, используемое для трехмерного моделирования. Оно предназначено для проектирования и моделирования высокочастотных электронных продуктов: антенн, микроволновых компонентов, фильтров и т.д. В программе используется понятный графический интерфейс, который помогает получить высокую производительность и понимание всех проблем, связанных с трехмерной электромагнитной моделью. Благодаря возможности использования инструментов, отслеживающих состояние окружающей среды (например, данные о температуре и гидродинамике), пользователь получает полный анализ, обеспечивающий тепловую и структурную надежность.

Таким образом, в настоящее время существует необходимость автоматизации численного моделирования электромагнитных полей. Для оптимизации этого процесса разработано много различных прикладных пакетов, которые делают численное моделирование менее трудозатратным и более эффективным.

Численное моделирование является одним из наиболее эффективных методов изучения сложных систем. Численные модели проще и удобнее в использовании, так как они дают нам возможность проводить вычислительные эксперименты, когда проведению физических экспериментов мешает финансовые или физические препятствия. Численное моделирование заключается в проведении серии вычислительных экспериментов при помощи ЭВМ. Их целью является анализ, интерпретация и сопоставление результатов моделирования и реального поведения объекта с последующим уточнением модели, если это будет необходимо.

Мы рассмотрели несколько методов численного моделирования. Каждый из них имеет свои преимущества и недостатки.

К преимуществам метода конечных разностей можно отнести:

- высокую универсальность.
- относительную простоту построения решающего алгоритма.

К числу недостатков МКР разностей можно отнести следующие характеристики:

- проблематичность использования на нерегулярных сетках.
- быстрый рост вычислительной трудоемкости при увеличении числа неизвестных переменных.
- сложность аналитического исследования свойств разностной системы.

Преимуществами метода конечных элементов являются:

- высокая универсальность метода.
- возможность использования разнообразных сеток.
- сравнительная простота и единообразие способов построения схем.

К недостаткам МКЭ относятся:

- отсутствие свойства локальной консервативности.
- выполнение законов сохранения гарантируется только на глобальном масштабе.

Важным вопросом в численном моделировании электромагнитных полей так же является выбор программного обеспечения. В настоящее время существует огромный выбор программных пакетов, которые упрощают процесс численного моделирования. Пользователь может выбрать те программы, которые подходят для его целей. Благодаря этому результат его работы будет более надежным, в процессе будут вовремя устранены несоответствия требованиям и сведено к минимуму количество ошибок, что сделает процесс численного моделирования более успешным.

Список использованных источников:

1. Основные методы численного моделирования [Электронный ресурс]. – Режим доступа : <https://lektsii.org/9-43773.html>

2. Метод конечных элементов [Электронный ресурс]. – Режим доступа : <https://portal.tpu.ru/>

3. Автоматизация численного моделирования [Электронный ресурс]. – Режим доступа : <https://www.comsol.ru/blogs/automate-modeling-tasks>

UDC 519.85:537.63

NUMERICAL SIMULATION OF ELECTROMAGNETIC FIELDS

Bushenko V. A., Demidovez V. V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Smirnova G. F. – assistant professor, PhD in Physics and Mathematics, associate professor of the department of physics

Annotation. In this paper, methods of numerical modeling of electromagnetic fields, as well as programs used for calculations in this field are considered.

Keywords. Finite element method, finite difference method, automation of numerical modeling, electromagnetic fields.

ИССЛЕДОВАНИЕ КОСМИЧЕСКИХ ТЕЛ С ПОМОЩЬЮ ГАММА-ИЗЛУЧЕНИЯ

Войченко М.М., Василевский В.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Леонович А.А. – канд. физ.-мат.наук, доцент кафедры физики

Из всех известных науке электромагнитных волн, гамма-излучение обладает самой малой длиной волны (меньше $5 \cdot 10^{-3} \text{ нм}$), что в свою очередь позволяет ему переносить большое число энергии (отдельные кванты имеют энергию, превышающую 10^5 эВ). Именно из-за данной энергии гамма-излучение является опасным для жизни на земле, так как оно способно проходить сквозь почти любого материала, вызывая ионизацию внутри молекул.

Лишь благодаря атмосфере Земли, гамма-излучение не может достичь её поверхности. Это и усложняет использование гамма-излучения для исследования космоса, так как волны попросту не могут выйти за пределы Земли. Мало того, волны почти свободно проникают сквозь стекло, что затрудняет их фиксацию в телескопах. Однако, после различных открытий в области физики и химии, NASA удалось решить обе эти проблемы, соорудив тем самым аппарат, называемый "GLAST" ("Gamma-Ray Large Area Space Telescope", или "Космический гамма-телескоп широкого поля действия"), в 2008 году 11 июня. В том же году 28 августа проект был переименован в "FGST" ("Fermi Gamma-Ray Space Telescope", или "Космический гамма-телескоп Ферми"), в честь итальянского физика XX века Энрико Ферми.

На рисунке 1 показано изображение, полученное данным телескопом. Светлая линия в середине изображения – гамма-излучение, приходящее на землю из центра Млечного Пути. Отдельные яркие точки – сверхновые звёзды [1-2].

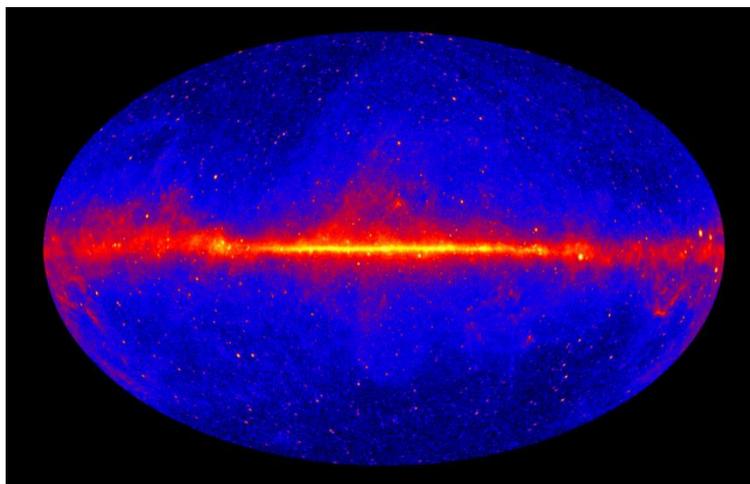


Рисунок 1 – Изображение, полученное FGST

С помощью данного телескопа удалось провести множество открытий и исследований, в частности:

– В ноябре 2010 года было объявлено об обнаружении двух эллиптических плазменных образований, излучающих гамма- и рентгеновское излучения и простирающихся на 25000 световых лет по обе стороны от центра галактики. Впоследствии они были названы «пузырями Ферми» и исследованы более подробно.

– 16 сентября 2008 года был обнаружен сильнейший по энергии замеченный когда-либо гамма-всплеск, названная GRB 160908C. Энергия взрыва оценивается в 9000 взрывов сверхновых, а скорость выброса звездного вещества как минимум в 99,9999% скорости света.

– Пульсар, светящийся только в гамма-диапазоне у сверхновой СТА1, расположенной на расстоянии 4600 световых лет от земли. Является первым крупным исследованием, связанным с телескопом Ферми.

Телескоп включает такие приборы как (рис. 2):

– Large Area Telescope (LAT) – гамма-телескоп, способный фиксировать отдельные излучение энергией от 20 МэВ до 300 ГэВ. Для решения проблемы фиксирования гамма-излучения он использует слой вольфрама в своих детекторах. Когда гамма-излучение, несущее в себе большое количество энергии, в него врывается, оно производит пары частиц электронов и позитронов. Используя направления этих пар, становится возможным определить источник гамма-излучения. А отдельный прибор, называемый калориметром, подсчитывает общую энергию этих пар, которая в свою очередь, равна энергии изначальной волны. Чтобы предотвратить попадание других волн, перед слоем вольфрама размещен ещё один детектор. [1]

– Gamma-ray Burst Monitor (FermiGBM) – прибор для обнаружения неожиданных вспышек гамма-излучения, вызванных гамма-всплесками и солнечными вспышками. Его диапазон чувствительности составляют рентгеновские и гамма-лучи энергией от 8 кэВ до 40 МэВ.[3] Включает в себя два набора детекторов: двенадцать сцинтилляторов на основе иодида натрия, способных регистрировать излучение от нескольких КэВ до МэВ, и два сцинтиллятора на основе германата висмута, способных регистрировать излучение примерно от 150 кэВ до 30 МэВ. GBM способен обнаружить в районе 200 гамма-вспышек в год.[1]

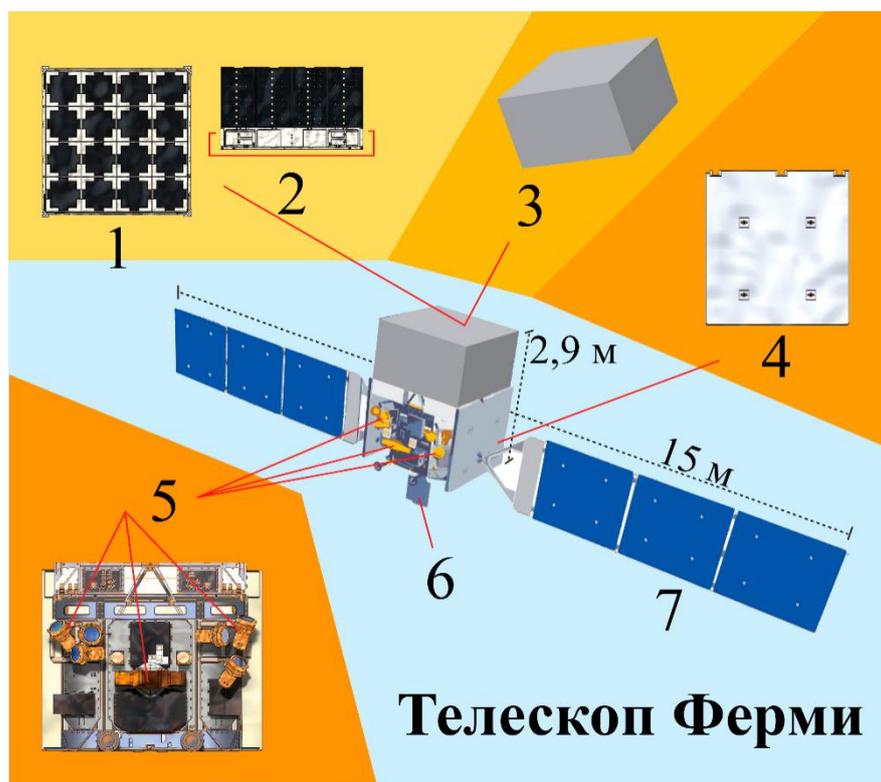


Рисунок 2 – Схематичное изображение приборов телескопа Ферми

На рисунке обозначены следующие приборы: 1 – трекер LAT; 2 – калориметр LAT; 3 – детектор несовпадений LAT; 4 – радиатор LAT; 5 – GBM; 6 – антенна ки-частот; 7 – солнечные панели.

Таким образом, гамма-излучение, имея самую короткую длину волны, помогает ученым видеть космические объекты под совершенно новым ракурсом, недоступным для человеческого глаза.

Список использованных источников:

- 1.Официальный сайт гамма-телескопа Ферми [Электронный ресурс]/ Режим доступа: <https://fermi.gsfc.nasa.gov>
- 2.Официальный сайт НАСА [Электронный ресурс]/ Режим доступа: <https://www.nasa.gov/>
- 3.Астрофизика гамма излучения в Национальном Центре Космоса, Технологий и Науки [Электронный ресурс] / Режим доступа: <https://gammaray.nsstc.nasa.gov/>

ЭЛЕКТРОЛИЗ ВОДЫ ДЛЯ ПОЛУЧЕНИЯ ЭКОЛОГИЧЕСКИ ЧИСТОГО ТОПЛИВА

Гембар С.В., Шиян А.Г.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Савилова Ю.И. – доцент кафедры физики, канд. тех. наук, доцент

Аннотация. В работе представлено описание устройства для получения водородного топлива путем электролиза.

Ключевые слова. Электролиз воды, электролизер, водородное топливо, КПД.

Введение. В наше время огромную роль играет транспорт. Самый распространенный вид топлива для транспорта является продукты нефтепереработки. Они обладают высоким КПД и небольшой стоимостью, однако существенный их недостаток – это большой выхлоп загрязняющих веществ. Решить эту проблему можно путем использования иных видов топлива. Одним из этих видов и является водород

Основная часть. Целью работы является описание устройства «Электролизер», его компонентов для получения водорода и способы применения.

При разработке данного устройства используется принцип электролиза, первооткрывателем которого считается известный британский ученый М. Фарадей. Электролиз -- это окислительно-восстановительная реакция, которая протекает только под действием электричества. В промышленных [генераторах водорода](#) для получения водорода и кислорода проводят электролиз воды. Для протекания реакции необходимо поместить в электролит два электрода, подключенных к источнику питания постоянного тока.

Для данного устройства необходим постоянный электрический ток, закрытое помещение, на которое не оказывает влияние внешняя среда.

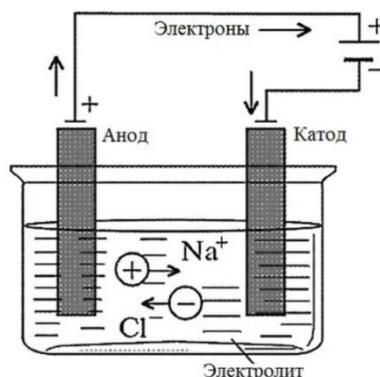


Рис. 1 – принцип работы электролизера

Электролизер работает благодаря внешнему источнику питания, который подает электрический ток. Упрощенно агрегат выполнен в виде U-образного корпуса, в который вмонтировано два или несколько электродов. Внутри корпуса находится электролит. При подаче электрического тока происходит разложение раствора на требуемые составляющие. Положительно заряженные ионы водорода направляются к отрицательно заряженному электроду (а ионы кислорода направляются к положительно заряженному электроду) и попадают в цилиндр, в котором аккумулируются и в дальнейшем по выводной трубке извлекаются.

Сама по себе дистиллированная вода не является проводником поэтому в нее добавляют электролиты с меньшим катионным потенциалом, чтобы избежать конкуренцию с катионами водорода: KOH или NaOH. Электрохимическая реакция протекающая на электродах выглядит следующим образом:

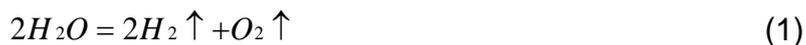
Реакция на аноде: $2H_2O \rightarrow O_2 + 4H^+ + 4e^-$ - выделение кислорода.

Реакция на катоде: $2H_2O + 2e^- \rightarrow H_2 + 2OH^-$ - выделение водорода.

В качестве материала для электродов нужно использовать химически нейтральные вещества, например нержавеющая сталь высокого качества или графитовые электроды.

Далее будут представлены расчёты, необходимые для устройства.

Формула электролиза:



Пусть у нас x молей воды, тогда из уравнения (1) по пропорции получаем $v(H_2)=x$, $v(O_2)=\frac{x}{2}$.

По закону $p = \frac{vRT}{V}$ найдем V для каждого цилиндра:

$$P_{O_2} = \frac{v(O_2)RT}{V(O_2)}, P_{H_2} = \frac{v(H_2)RT}{V(H_2)}$$

Для стабильной работы необходимо чтобы давление внутри устройство было стабильным и одинаковым в обоих цилиндрах.

$$\frac{P_{O_2}}{P_{H_2}} = 1 \Rightarrow \frac{v(O_2)}{V(O_2)} \cdot \frac{v(H_2)}{V(H_2)} = \frac{x}{2V(O_2)} \cdot \frac{V(H_2)}{x} = \frac{V(H_2)}{2V(O_2)} = 1$$

$$2V(O_2) = V(H_2) \quad (2)$$

Из уравнения (2) нам известна разница объемов цилиндров.

Так как при разложении воды на кислород и водород ее объем уменьшается, в устройстве присутствуют клапаны, которые стабилизируют ее объем, чтобы не нарушить баланс давления.

На рисунке 1 представлена схема устройства. Описание его составляющих:

1. Катод, на нем собираются ионы водорода;
2. Анод, на нем собираются ионы кислорода;
3. Клапаны, запускающие раствор электролита, для стабилизации давления
4. Трубка, через которую выходит водород;
5. Трубка, через которую выходит кислород;
6. Пластика, не подпускающая смешивание водорода и кислорода;

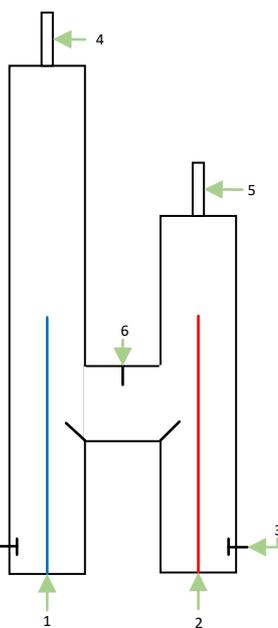


Рис. 1 – схема устройства «Электролизёр»

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

Удельная теплота сгорания водорода составляет примерно 140 МДж/кг, что в несколько раз превышает удельную теплоту сгорания углеводородных топлив (для метана – около 50 МДж/кг)

Преимущества водородных двигателей внутреннего сгорания

1. Главное неоспоримое преимущество автомобилей на водороде – это высокая экологичность, так как продуктом горения водорода является водяной пар. Конечно, при этом сгорают еще различные масла, но токсичных выбросов гораздо меньше, чем у бензиновых выхлопов.
2. Простая конструкция.
3. Отсутствие дорогостоящих систем топливоподачи, которые к тому же опасны и ненадежны.
4. Бесшумность.
5. КПД электродвигателя на водородном топливе намного выше, чем у ДВС.

В настоящее время многие мировые производители транспорта переходят на водородное топливо, например:

- **Toyota** — модель Toyota Highlander FCHV;
- Ford Motor Company проводит испытания с концептом Focus FCV;
- Honda со своей моделью Honda FCX;
- Hyundai выпускает Tucson FCEV;
- И др.

Список литературы

1. Мычко, Д. И. Учебная программа факультативных занятий «Электричество и химия» [Электронный ресурс] – Режим доступа: http://adu.by/images/2016/08/15_elektric_i_himia.docx – Дата доступа: 02.04.2017.
2. Матулис, Вадим Э. Электролиз водных растворов и расплавов солей / Вадим Э. Матулис, Виталий Э. Матулис, Т. А. Колевич [Электронный ресурс] / Национальный образовательный портал. – Минск, 2016. – Режим доступа: <http://www.adu.by/images/2016/11/dop-materelektroliz-solej-11kl-kolevich.pdf>. – Дата доступа: 12.06.2017.
3. [Электронный ресурс] -- Плюсы и минусы водородного топлива. Режим доступа: <https://www.monsterauto.ru/articles/plyusy-i-minusy-vodorodnogo-topliva.html>

UDC 654.147.6:621.6.028

ELECTROLYSIS OF WATER TO PRODUCE ENVIRONMENTALLY FRIENDLY FUEL

Hembar S. V., Shiyan A.G.

*Belarusian State University of Informatics and Radioelectronics
Minsk, Republic of Belarus*

Scientific adviser:

Annotation. The paper presents a description of a device for producing hydrogen fuel. ива.

Keywords. Water electrolysis, electrolyzer, hydrogen fuel, efficiency.

ОПТИМИЗАЦИЯ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ И АНАЛИЗА ДАННЫХ НА ПРИМЕРЕ ЭФФЕКТА ЗЕЕБЕКА

Глинский И.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Григорьев А.А. – канд. физ-мат. наук, доцент

Аннотация. В данной работе исследуется применение машинного обучения и анализа данных для оптимизации эксперимента по изучению эффекта Зеебека. Цель состоит в разработке методики определения оптимальных параметров термопар для максимизации коэффициента Зеебека. Используя методы кластеризации, предобработки данных и машинного обучения, успешно выявлены закономерности в экспериментальных данных, что позволило определить оптимальные условия для проведения эксперимента. Результаты данного исследования могут быть полезны для разработки новых материалов и устройств, основанных на эффекте Зеебека.

Эффект Зеебека (также известный как термоэлектрический эффект) - это физический явление, при котором температурная разница между двумя разными материалами, соединенными в контактные точки, приводит к возникновению электродвижущей силы (напряжения). Это явление было впервые обнаружено Томасом Зеебеком в 1821 году и с тех пор находит широкое применение в различных областях науки и техники, особенно

Ключевым параметром, характеризующим эффект Зеебека, является коэффициент Зеебека. Его значение зависит от свойств материала и температурного градиента.

$$E = \alpha(T_2 - T_1) \quad (1),$$

Где E - коэффициент Зеебека; α - термоэлектрическая способность пары, T_2, T_1 - температуры контактов.

Одним из наиболее распространенных применений эффекта Зеебека являются термопары - простые и недорогие устройства для измерения температуры. Термопары состоят из двух проводников из разных материалов, соединенных на одном конце. При нагреве этого соединительного конца возникает термоэлектрическое напряжение, которое можно измерить и преобразовать в соответствующее значение температуры.

В данной работе был проведен анализ данных по семи различным типам термопар (Т, К, J, Е, В, R и S) с целью оптимизации их использования. В ходе исследования были применены различные методы машинного обучения и кластеризации, такие как линейная регрессия, случайный лес, градиентный бустинг и метод опорных векторов, метод K-средних и агломеративная иерархическая кластеризация.

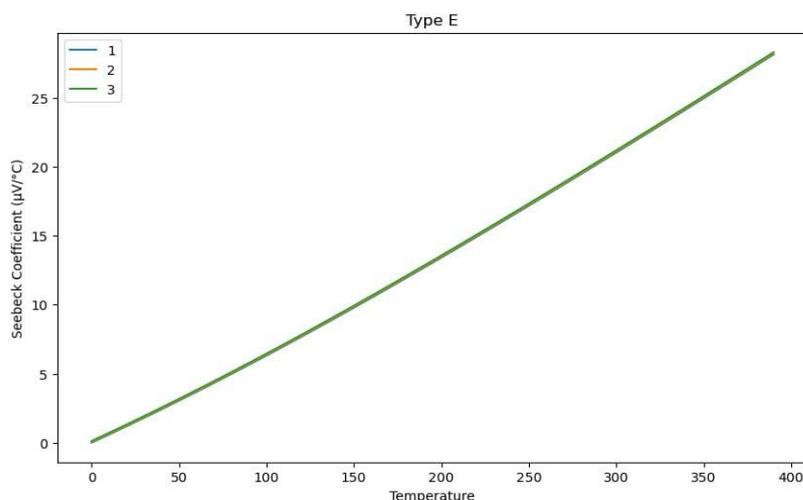


Рисунок 1 – график зависимости коэффициента Зеебека от разности температур термопары типа E

Для нахождения коэффициента Зеебека были использованы следующие методы: линейная регрессия, случайный лес, градиентный бустинг и метод опорных векторов. Производительность этих моделей оценивалась с использованием показателей среднеквадратичной ошибки (MSE) и коэффициента детерминации (R-squared), а также перекрестной проверки для оценки надежности каждой модели. Результаты были следующими:

Линейная регрессия показала наилучшие результаты с MSE 0,22 и значением R-squared 0,9996. Результаты перекрестной проверки показали MSE 4,43 и значение R-squared 0,9511, что указывает на высокий уровень согласованности и точности.

Кроме того, были применены методы кластеризации (обучение без учителя), такие как K-средних и агломеративная иерархическая кластеризация, для группировки термопар по их характеристикам и поведению в разных температурных диапазонах. Эти методы позволяют определить сходства между различными типами термопар и выделить группы с похожими свойствами.

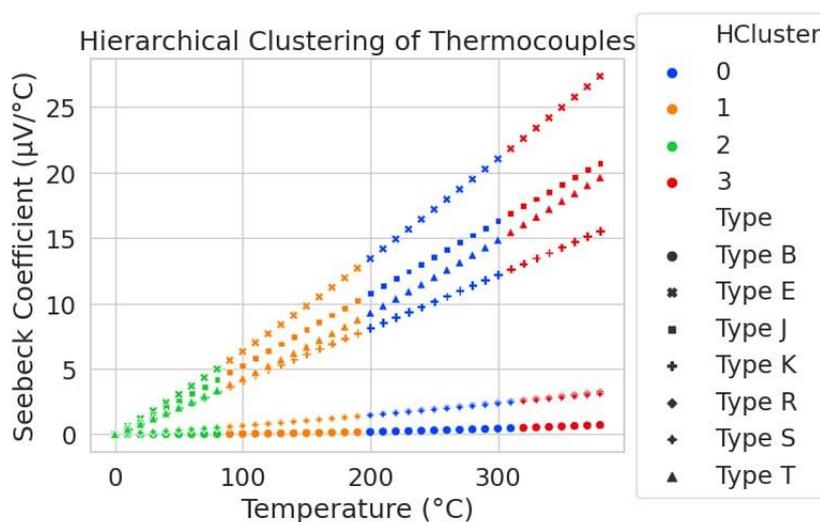


Рисунок 2 – иерархическая кластеризация термопар.

Агломеративная иерархическая кластеризация выявила четыре кластера, сгруппированные в соответствии с температурными диапазонами и коэффициентами Зеебека. В частности, первый кластер включает термопары, которые демонстрируют схожие характеристики при температуре в диапазоне 1 кластера. Это свидетельствует о том, что в определенных диапазонах температур разные типы термопар могут вести себя аналогично.

Таким образом, можно сделать вывод о том, что в некоторых случаях использование более дешевых термопар может быть предпочтительным, так как они ведут себя аналогично более дорогим моделям. Например, если наше устройство работает в диапазоне 1 кластера, то имеет смысл выбирать более экономичные варианты термопар, так как они обеспечивают сопоставимую точность и характеристики.

Помимо этого, важно отметить, что результаты данного исследования могут быть использованы не только для определения наиболее подходящих термопар, но и для разработки новых материалов и термоэлектрических устройств с оптимальными характеристиками. Ведь понимание закономерностей и связей между температурой и коэффициентом Зеебека может способствовать созданию новых термоэлектрических материалов с повышенной эффективностью и надежностью.

Результаты данного исследования могут быть использованы для оптимизации выбора термопар в зависимости от конкретных условий их применения. Применение методов машинного обучения и кластеризации позволяет выявить закономерности и сходства между различными типами термопар, что может помочь пользователям принимать обоснованные решения при выборе наиболее подходящего типа термопары для своих нужд.

Список использованных источников:

1. Савельев И. В., Курс физики : учеб. В 3 т. Т. 3: Квантовая оптика. Атомная физика. Физика твердого тела. Физика атомного ядра и элементарных частиц / И. В. Савельев. – М. : Наука. Гл. ред. физ.-мат. лит., 1989. – 304 с.
2. Епифанов, Г. И. Физика твердого тела : учеб. пособие / Г. И. Епифанов. – 4-е изд., стер. – СПб. : Издательство «Лань», 2011. – 288 с.

ИССЛЕДОВАНИЕ ПЕРЕКЛЮЧАТЕЛЕЙ СВЕРХПРОВОДИМОСТИ И ГРАФЕНА С МАГИЧЕСКИМ УГЛОМ НАКЛОНА

Лицкевич В.В., Гришель Д.Г., Дроздович В.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Леонович А.А. – канд. физ.-мат. наук, доцент

Аннотация. В данной работе мы всесторонне исследуем переключения сверхпроводимости и графен с магическим углом. Исследование основано на последних достижениях в области изучения графена и открытии эффекта магического угла. Наша работа включает в себя сочетание экспериментальных и теоретических подходов, чтобы лучше понять принципы, лежащие в основе переключений сверхпроводимости и графена с магическим углом.

Магический угол — это определенный угол, под которым два слоя графена или других двумерных материалов наложены друг на друга, что приводит к появлению новых проводящих свойств. Магический угол возникает, когда два слоя поворачиваются относительно друг друга на определенный угол, который обычно составляет около 1,1 градуса. Под этим углом слои графена создают муаровый узор, который заставляет электроны в материале вести себя уникальным образом, что приводит к появлению сверхпроводимости, состояния изолятора Мотта и других интересных свойств.

Открытие переключений сверхпроводимости и графена с магическим углом привело к новому волнению в области физики конденсированных сред. Переключения сверхпроводимости относятся к способности некоторых материалов переключаться между сверхпроводящим и несверхпроводящим состояниями при определенных условиях. Под графеном с магическим углом понимается расположение графеновых листов под определенными углами, что приводит к появлению новых электронных свойств.

Экспериментальная работа, проведенная в исследовании MIT [1], включала изготовление графеновых устройств с магическим углом с использованием сканирующей туннельной микроскопии высокого разрешения и атомно-силовой микроскопии. Затем устройства были подвергнуты различным электрическим и магнитным измерениям, включая транспортные измерения и сканирующую туннельную спектроскопию.

Теоретическая работа включала использование расчетов на основе первых принципов для моделирования электронных свойств графена с магическим углом. Моделирование проводилось с использованием теории функционала плотности и приближения. Результаты сравнивались с экспериментальными измерениями для подтверждения теоретических моделей.

Результаты показывают, что переключение сверхпроводимости в графене с магическим углом обусловлено наличием плоских электронных полос на уровне Ферми. Плоские полосы создают высокую плотность состояний, что приводит к образованию куперовских пар и возникновению сверхпроводимости.

Устройства на основе графена с магическим углом наклона проявляют и другие интересные свойства, такие как наличие состояний изолятора Мотта и топологических фаз. Эти свойства возникают благодаря уникальной электронной структуре графеновых листов под магическим углом.

Уникальная электронная структура графеновых листов с магическим углом наклона приводит к появлению различных других интересных свойств, помимо сверхпроводимости, таких как наличие состояний изолятора Мотта и топологических фаз [2]. Состояния изолятора Мотта возникают, когда электроны в материале становятся сильно коррелированными, в результате чего материал не проводит электричество даже при низких температурах. В графене с магическим углом такие состояния возникают из-за совместного эффекта электронных взаимодействий и муарового узора, образуемого слоями, уложенными друг на друга. Аналогично, топологические фазы означают наличие особых электронных состояний на краях материала, которые защищены от рассеяния примесями или дефектами. Таким образом было показано, что графен под магическим углом демонстрирует различные топологические фазы, включая квантовый аномальный эффект Холла и дробный квантовый эффект Холла [3]. Обнаружение этих свойств в графене с магическим углом открыло новые возможности для изучения фундаментальной физики и разработки новых электронных устройств.

В заключение следует отметить, что наше исследование дает полное представление о переключателях сверхпроводимости и графене под магическим углом. Результаты демонстрируют потенциал графена для разработки новых электронных устройств и приложений. Мы считаем, что данное исследование вдохновит дальнейшие исследования в области физики конденсированных сред и проложит путь к разработке новых технологий на основе графена.

Список использованных источников:

1. Superconductivity switches on and off in “magic-angle” graphene [Электронный ресурс]. – Режим доступа: <https://news.mit.edu/2023/study-superconductivity-switches-and-magic-angle-graphene-0130>. – Дата доступа: 20.03.2023.
2. Magic-angle graphene switches from superconductor to ferromagnet [Электронный ресурс]. – Режим доступа: <https://physicsworld.com/a/magic-angle-graphene-switches-from-superconductor-to-ferromagnet/>. – Дата доступа: 18.03.2023.
3. MIT Scientists Uncover Switchable Superconductivity in Magic-Angle Graphene, Paving Way for Neuromorphic Computing [Электронный ресурс]. – Режим доступа: <https://www.sciencetimes.com/articles/42135/20230131/mit-scientists-uncovers-switchable-superconductivity-magic-angle-graphene-paving-way.htm>. – Дата доступа: 17.03.2023.

ФИЗИЧЕСКИЕ ПРИНЦИПЫ МАГНИТНОЙ ЗАПИСИ И СЧИТЫВАНИЯ ИНФОРМАЦИИ

Жоголь А.О

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Григорьев А.А. – кандидат физико-математических наук, доцент

Аннотация. В данной работе рассматривается магнитная запись информации. Цель работы состоит в том, чтобы изучить физические принципы, по которым работает магнитная запись информации.

Магнитная запись – способ записи информации путем изменения магнитного состояния носителя и создания в нем распределения намагниченности, соответствующего записываемому сигналу. Намагниченность каждого участка носителя определяется совокупностью намагниченности отдельных частиц (доменов), расположенных в нем. Под действием магнитного внешнего поля (поля головки записи) происходит поворот векторов намагниченности этих частиц и эта ориентация (информация) сохраняется относительно бесконечно долго [1-4].

Явления магнетизма основаны на существовании магнитных моментов в атомах. В таких элементах, например, как железо, кобальт, никель, магнитные моменты возникают в результате спина электронов, а в редкоземельных элементах — в результате кругового вращения электронов вокруг ядра атомов. Электроны, определяющие магнитные свойства железа и его сплавов, находятся на третьей электронной оболочке атома, а электроны редкоземельных элементов — на четвертой электронной оболочке.

Электроны на внешней оболочке 4s, как у всех металлов, являются свободными. В ансамбле атомов кристаллической структуры они являются носителями зарядов и обуславливают наличие электронной проводимости. На внутренних оболочках 1s-3p размещается одинаковое количество электронов с положительным и отрицательным зарядами, вследствие чего их магнитные моменты взаимно компенсируются. На оболочке 3d находятся пять электронов с положительным спином и только один — с отрицательным. В результате спинового взаимодействия все оси вращения параллельны между собой. Таким образом, сумма моментов спинов не равна нулю и атом представляет собой — как бы постоянный магнит.

Под действием сил обменного взаимодействия магнитные моменты соседних атомов ориентируются параллельно или антипараллельно друг другу. Если атомы кристаллической структуры располагаются на минимальном расстоянии один от другого, то множество атомов объединяется в домен, который называют также областью Вейса. В пределах домена намагниченность постоянна. Возможность образования доменов является предпосылкой возникновения ферромагнетизма.

Для каждого домена (или для каждой однодоменной частицы) напряженность магнитного поля, вызывающего скачок намагниченности, может быть различной. Обычно скачки намагниченности почти одновременно совершаются в нескольких доменах (однодоменных частицах), векторы намагниченности которых после этого почти совпадают с направлением внешнего поля. Если теперь опять выключить внешнее поле, то большинство доменов магнитотвердых материалов (постоянных магнитов или материалов, используемых в качестве носителей магнитной записи) остается в новом, принятом ими положении; это изменение уже имеет необратимый характер.

Если магнитный материал повторно поместить в поле H_2 , то результирующая намагниченность достигнет примерно прежнего значения, но кривая намагничивания будет иметь другую форму, чем при включении поля H_2 . При этом считают, что в пределах этой замкнутой петли свойства материала постоянны, а материал ведет себя магнитно стабильно. Если продолжать увеличивать напряженность внешнего поля, то в направлении поля будут выстраиваться векторы намагниченности все большего числа доменов. Результирующая намагниченность медленно возрастает и, наконец, достигает намагниченности насыщения M_s , значение которой в однородной среде совпадает со спонтанной намагниченностью домена, а в порошкообразных материалах меньше и приблизительно пропорциональна коэффициенту объемной концентрации частиц. Ясно,

что состояние полной ориентации во всех доменах (частицах) сохраняется стабильным только на время существования поля насыщения H_s . Если выключить поле, то некоторые домены займут наиболее благоприятное для них положение наступает состояние остаточной намагниченности насыщения M_r .

Рассмотрим теперь, как петля гистерезиса от достигнутой точки остаточной намагниченности насыщения M_r пойдет дальше, если усиливать внешнее магнитное поле в обратном направлении. Результирующая намагниченность образца будет уменьшаться. Происходящие в этом случае процессы совершенно аналогичны рассмотренным ранее. Напряженность поля, при которой намагниченность образца достигает нуля, называется коэрцитивной силой и обозначается H_c .

После увеличения напряженности поля обратного направления до значения $-H_c$, соответствующего насыщению, затем переключения его опять в первоначальное направление и увеличения до $+H_s$ кривая зависимости намагниченности M от приложенного поля H приобретает вид симметричной замкнутой петли, называемой предельной петлей гистерезиса. Если намагничивание ведется не до насыщения, то для каждого цикла получается своя частная петля внутри предельной, при этом вершины всех этих петель располагаются на начальной кривой. Соответственно и значения остаточной намагниченности частных петель M_0 лежат внутри области между значениями остаточных намагниченностей насыщения от $+M_s$ до $-M_s$. Это и дает возможность дискретно различать градации записанного сигнала на носителе. Если построить график зависимости остаточной намагниченности M_0 от максимальной напряженности действующего поля H , то получим кривую остаточной намагниченности материала. Кривая остаточной намагниченности представляет собой одну из наиболее важных характеристик носителей магнитной записи. Воспроизводимые сигналы получаются наибольшими, когда в процессе записи кривая остаточной намагниченности используется до насыщения. Линейный же участок кривой позволяет вести запись без искажений.

Если образец магнитного материала подвергается воздействию переменного поля, то петля гистерезиса полностью описывается за каждый период. Требуемая для совершения такого цикла работа перемагничивания пропорциональна площади петли и частоте перемагничивания. Эта работа в основном представляет собой тепловые потери. Поэтому для устройств, работающих в переменных полях (магнитные головки, трансформаторы и т.д.), применяют магнитные материалы с малой площадью петли гистерезиса, главным образом с малой коэрцитивной силой H_c .

В процессе магнитной записи изменяющиеся во времени электрические величины преобразуются в локальные изменения намагниченности носителя. Временные соотношения сигналов сохраняются преимущественно вследствие движения носителя при неподвижной записывающей или воспроизводящей головке. Чисто физическое значение имеет только относительное перемещение.

В качестве записывающей или воспроизводящей головки преимущественно используются головки с сердечником из высокопроницаемого магнитомягкого материала, имеющим рабочий зазор, перед которым передвигается носитель записи.

Спад поля головки с увеличением расстояния до сердечника (зависимость от y) и слабая концентрация поля записи $H(x, y, t)$, а также конечная крутизна петли гистерезиса в значительной степени препятствуют желаемому однозначному преобразованию сигнала в намагниченность носителя записи. На рис. 7 доказана записывающая головка, по обмотке которой протекает ток записи $I(t)$. Создаваемое этим током магнитное поле рассеяния пронизывает магнитный рабочий слой носителя и намагничивает его.

Изменения намагниченности в рабочем слое $M(\xi)$ только приблизительно пропорциональны изменениям тока в головке. Поле рассеяния намагниченного носителя, движущегося с постоянной скоростью v в направлении стрелки, создает в сердечнике воспроизводящей головки магнитный поток $\Phi_1(t)$. При этом на выводах обмотки головки возникает напряжение $U_w(t)$.

Из сказанного можно видеть, что рабочие свойства носителя записи достаточно полно характеризуются толщиной рабочего слоя и гистерезисными свойствами:

- остаточной магнитной поляризации;
- коэрцитивной силой;

- начальной проницаемостью и дифференциальной максимальной проницаемостью.

Список использованных источников:

1. Бургов В.А. Теория фонограмм. – М.: Искусство, 1984. -302 с.
2. Гитлиц М.В. Магнитная запись сигналов: Учебное пособие для высших учеб. заведений. – М.: Радио и связь, 1990. - 231 с.
3. Василевский Ю.А. Носители магнитной записи. – М.: Искусство, 1989. – 287 с.
4. Подгорная Е.А. Магнитная аналоговая запись: Учебное пособие. - СПб: Изд. СПбГУКиТ, 2004. – 48с.
5. Сиаккоу М. Физические основы записи информации. Перевод с немецкого И.Д.Гурвица. Под редакцией В.Г. Королькова., Москва, Связь, 1980 – 188с.

СРАВНЕНИЕ ДВУМЕРНЫХ МАТЕРИАЛОВ ДЛЯ ПРИМЕНЕНИЯ В ТРАНЗИСТОРАХ

Кудин И. С. гр.272301, Громыко М. Р. гр.272601

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Смирнова Г.Ф. – доцент кафедры физики, канд. физ.-мат. наук, доцент

Аннотация. Данный доклад посвящен вопросу будущего использования двумерных материалов в нанoeлектронике на примере графена и силицена, описываются перспективы и препятствия в интеграции этих материалов в полевые транзисторы. Также обращается внимание на сравнение графена и силицена, различие и схожесть их свойств, делается вывод о более перспективном материале для применения в ближайшем будущем.

В условиях активного процесса миниатюризации даже ведущие производители вычислительных чипов вплотную подошлись к фундаментальным ограничениям в размере транзистора, после которого его элементы не смогут эффективно работать из-за преобладания нежелательных квантовых явлений [1]. По этой причине активно исследуются возможности использования двумерных неметаллов в качестве полупроводников для транзисторов нового типа, способных значительно превзойти в производительности традиционные транзисторы на кремнии.

Одним из таких материалов является графен. Графен – это аллотропная модификация углерода, представляющая собой двумерную решетку, состоящую из правильных шестиугольников. Он является основой для распространенных форм углерода, таких как графит, алмаз или фуллерены. Также графен используется для создания нанотрубок, которые обладают огромной прочностью натяжения. Получить графен удалось еще в 1970 году с помощью наращивания на металлической поверхности, но только в 2004 году этот материал был изолирован и экспериментально описан с точки зрения физики[2].

Графен является перспективным кандидатом для использования в транзисторах благодаря высокой подвижности зарядов в решетке. Используя тонкие графеновые ленты в каналах транзистора, возможно добиться ощутимого прироста переключаемости затвора, а следовательно и огромного прироста тактовой частоты, вплоть до нескольких терагерц при правильном подборе напряжения на затворе[3]. Более того, такие транзисторы будут иметь значительно большую энергоэффективность в сравнении с транзисторами на кремнии из-за меньшего электрического сопротивления графена[4].

Однако главная проблема по-прежнему остается в крайне дорогом процессе получения графена, так как до сих пор не были изобретены дешевые, эффективные и экологичные методы его производства в промышленных масштабах. По нынешний день основными способами выработки графена являются химическое расщепление графита галогенами и солями металлов и окисление графита и последующим его восстановлением. Оба способа требуют использования большого количества воды и сильных химикатов, что ограничивает дальнейшее распространение производства. Тем не менее, активно разрабатываются новые способы его получения, к примеру точное иглофрезерование графита и расщепление его в центрифугах[5].

Тем временем, параллельно с графеном идут исследования по применению другого двумерного материала с похожими свойствами – силицена.

Силицен, представляет собой атомарный слой кремния. Впервые силицен был получен методом молекулярно-лучевой эпитаксии на серебряной подложке. Свойства силицена схожи со свойствами его углеродного «родственника» графена. Например, носители заряда обладают таким же линейным законом дисперсии, как и в графене. Важным свойством силицена является большая химическая стабильность по сравнению с графеном. Другими словами, у графеновой полоски атомы углерода, находящиеся на краях, химически более активны, чем такие же атомы в полоске кремния. Это означает, что разрушить силицен намного сложнее, чем графен[6].

Силицен является аллотроп кремния, точно так же, как графен является аллотропом углерода, и состоит только из атомов кремния (одноатомных). В отличие от других двумерных материалов, таких как графен, силицен не является строго плоским, он имеет изогнутую сотовую поверхность.

Сравнив силицен с графеном мы можем наблюдать, что в отличие от графена силицен не является полностью плоским.

Силицен имеет запрещенную зону, что делает его пригодным для использования в новых транзисторах. Графен, с другой стороны, является полуметаллом и не имеет запрещенной зоны.

Силицен удобнее интегрировать в электронные устройства по сравнению с графеном, особенно для небольших и дешевых электронных устройств. Это связано с тем, что его можно легко интегрировать в линии по производству кремниевых чипов.

Однако, по сравнению с другими 2D-материалами, силицен широко не изучается и не производится, потому что его гораздо сложнее создать, чем любой 2D-материал, такой как графен.

Исходя из этого мы можем сделать, что развитие материалов не стоит на месте, однако на примере силицена, который имеет множество преимуществ перед графеном, мы видим, что графен остаётся той основной базой к которой продолжают ссылаться.

Список использованных источников:

1. Бреус, А.И. Оценка минимальных размеров локальных областей наноразмерного кристалла, при которых применимы классические понятия о работе транзистора / *Инфокоммуникационные технологии*, 2007. – С.22-25
2. Дьяковская, А.В., Телекова Л.Р. Графен: свойства, получение, перспективы применения / *Достижения науки и образования*, 2018. – 3 с.
3. Давидович М.В., Глухова О.Е., Слепченко М.М. Терагерцевый транзистор на основе графена / *Известия саратовского университета. Новая серия. Серия: физика*, 2017. – С.44-54
4. Муравьев В.В., Мищенко В.Н. Влияние одиночного слоя графена на характеристики полевого транзистора / *Материалы и структуры современной электроники*, 2018. – С.82-85
5. Журавлев В.В., Дудаков В.Б. Способ получения графена / *Патент: RU 2570069 C1*, 2015.
6. УДК 538.958 DOI 10.25587/SVFU.2022.89.60.006 Е. П. Шарин, Н. Я. Муксунов Структурные и электронные свойства гетероструктуры графен/силицен

ФОТОПРИЁМНИК НА ОСНОВЕ ФОТОТРАНЗИСТОРА

Казеко Д.А., Лойко А.И.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Иванов М.А. – канд. физ.-мат. наук

Фотоприёмник – это датчик, который генерирует электрический сигнал, зависящий от света, падающего на это устройство. То есть по мере того, как это электромагнитное излучение действует в большей или меньшей степени, оно будет генерировать тот или иной сигнал, который можно интерпретировать.

Фототранзистор — оптоэлектронный полупроводниковый прибор. Отличается от классического транзистора тем, что область базы доступна для светового облучения, при попадании которого на базу в полупроводнике базы происходит внутренний фотоэффект: образуются пары электрон-дырка. Фототранзистор имеет структуру n-p-n или p-n-p транзистора. Дырки электронно-дырочных пар, рождённых излучением, находятся в базе, а электроны переходят в эмиттер или коллектор.

Мы собрали следующую схему (Рис. 1). Принцип работы схемы заключается в следующем: на фототранзистор попадает свет, собранный линзой от источника света. Поскольку фототранзистор расположен на фокусном расстоянии линзы, весь свет от источника попадает на него. Фототранзистор преобразовывает световой сигнал в электрический. В результате чего изменяется напряжение на резисторе, которое мы считываем вольтметром.

а) б)

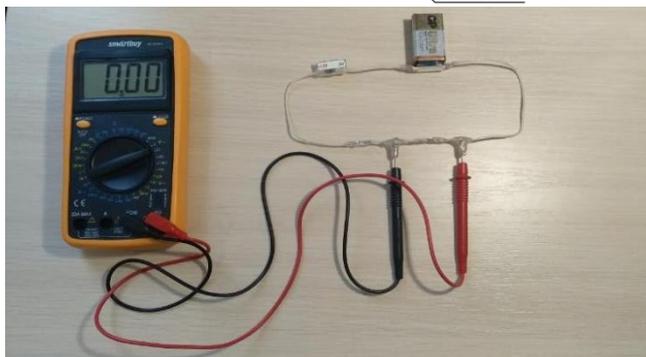
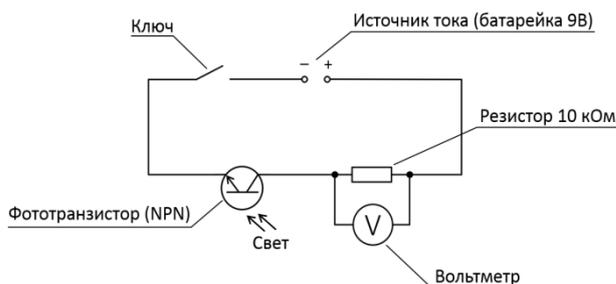


Рис 1 – а) схема и б) фото электрической цепи.

Мы поместили наш фотоприёмник в цилиндрический корпус из фольги, чтобы изолировать от внешнего светового излучения. Собранный фотоприёмник показан на Рис. 2а. Мы собрали спектрометр на основе монохроматора МУМ-2 (Рис. 2б).

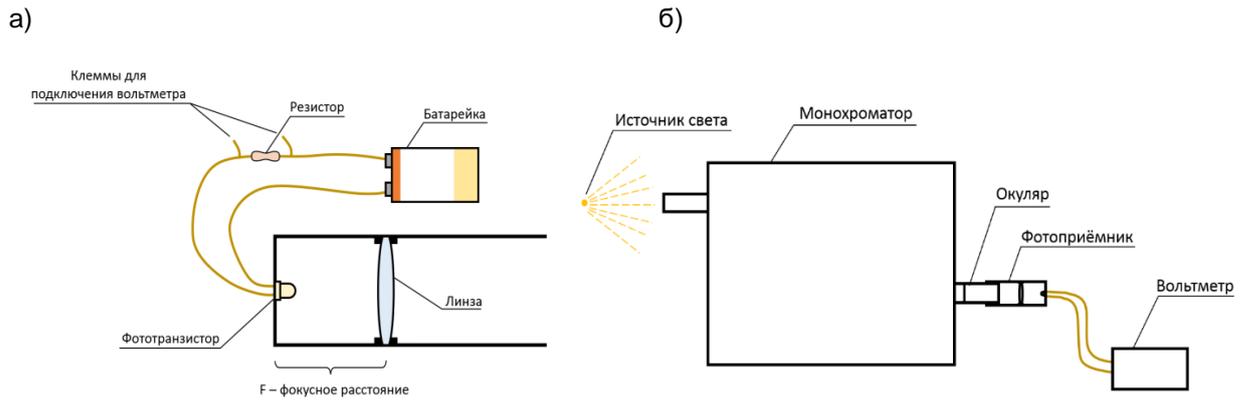


Рис. 2 – а) схема фотоприёмника б) схема спектрометра

В ходе эксперимента мы получили зависимость напряжения от длины волны светового излучения. И для получения спектра мы делили полученные значения на относительную спектральную чувствительность транзистора.

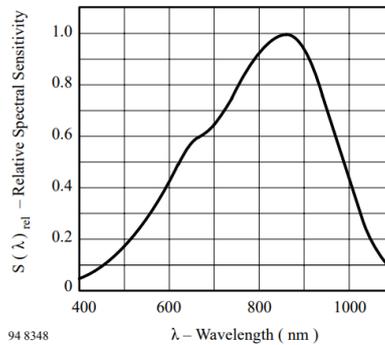


Рис. 3 – относительная спектральная чувствительность транзистора.

[1] Document Number 81531, www.vishay.com

Мы использовали спектрометр, чтобы измерить спектр теплового излучения лампы накаливания (на Рис. 4б расположен график, получившийся в результате предварительного эксперимента, форма графика может быть незначительно искажена из-за непрофессиональных условий проведения эксперимента). Также дополнительно измерили спектр светодиодного фонарика (смартфона).

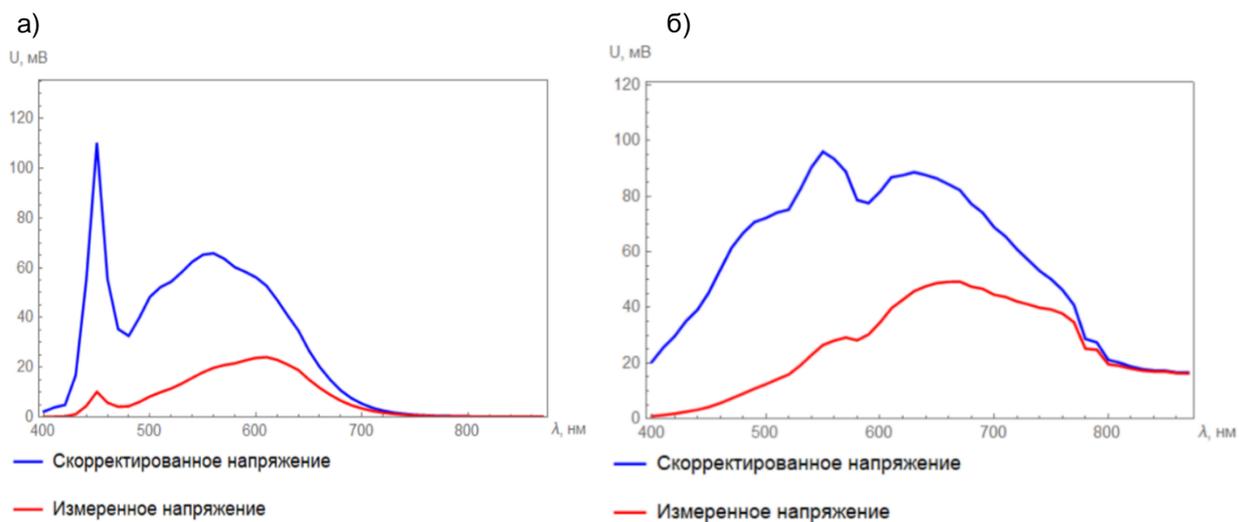


Рис. 4 – спектры а) светодиодного фонарика телефона б) лампы накаливания

Представленный нами прибор для получения спектра светового излучения может быть использован для определения температуры излучающего тела с помощью закона смещения Вина.

УДК 534.08

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ФИКСАЦИИ ПЕРИОДА ГАРМОНИЧЕСКИХ КОЛЕБАНИЙ ФИЗИЧЕСКОГО МАЯТНИКА

Михнюк А.И., Пашковец М.В.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

*Научные руководители: Кисель В.В. – канд. физ-мат. наук, доцент кафедры
физики,*

Широчин С.В. – вед. инж. кафедры физики

Аннотация. Разработана бюджетно реализуемая автоматизированная система фиксации временных характеристик механических колебаний, обеспечивающая высокую точность фиксации результатов, их стабильность, имеющая высокую степень надежности в процессе эксплуатации и позволяющая оптимизировать процесс проведения лабораторных исследований.

Ключевые слова: интегральная микросхема, колебания, физический маятник

Введение. При проведении лабораторных опытов студенты сталкиваются с проблемой нерационального использования учебного времени. Это обстоятельство выдвигает задачу оптимизации учебного процесса с целью повышения качества образования.

В данной статье авторами предложено техническое решение, которое способно оптимизировать процесс снятия показаний с лабораторных установок. Эта оптимизация повышает качество учебного процесса, снижает утомляемость студента, а также повышает качество итоговых результатов.

Основная часть. При создании автоматизированной системы фиксации колебаний, необходимо учесть следующие требования к техническому решению:

- точность измерений;
- стабильность результатов;
- эргономичность использования;
- бюджетность реализации;
- надёжность эксплуатации.

Для фиксации колебаний физического маятника рассмотрены два варианта:

- датчик Холла;
- оптопара.

На основе требований к техническому решению была выбрана оптопара. Оптический метод прост в реализации и эксплуатации. Он основан на прерывании оптического луча стержнем физического маятника.

Использование датчика Холла является нецелесообразным, так как для его работы понадобилась бы система магнитных полей. Создание подобной системы – трудоёмкий процесс. Кроме того датчик Холла не обладает достаточной степенью точности и для его функционирования требуется сложная усилительная схема.

Для фиксации выбрана транзисторная оптопара со светодиодом красного цвета, в частности, с длиной волны 800 нМ.

Период колебаний физического маятника фиксируется двукратным прерыванием оптического луча, что соответствует двум импульсам на выходе оптопары. Для предотвращения ложных срабатываний D-триггера сформирован цифровой сигнал с крутыми фронтами переключения. Для этого используется схема преобразования и стабилизация сигнала, которая является связующим элементом между оптопарой и частотомером.

Блок-схема автоматизированной системы фиксации колебаний физического маятника выглядит так (рисунок 1):



Рисунок 1 – Блок-схема автоматизированной системы фиксации колебаний

В качестве стабилизатора в этой работе выступила интегральная микросхема (ИМС) К561ЛЕ5. Она состоит из четырёх логических элементов 2ИЛИ-НЕ, которые скоммутированы по схеме инверторов.

Согласно электрической схеме (рисунок 2) после К561ЛЕ5 сформированный сигнал поступает на ИМС К561ТМ2. В этом устройстве данная микросхема выступает в качестве делителя входного сигнала с коэффициентом, равным 2.

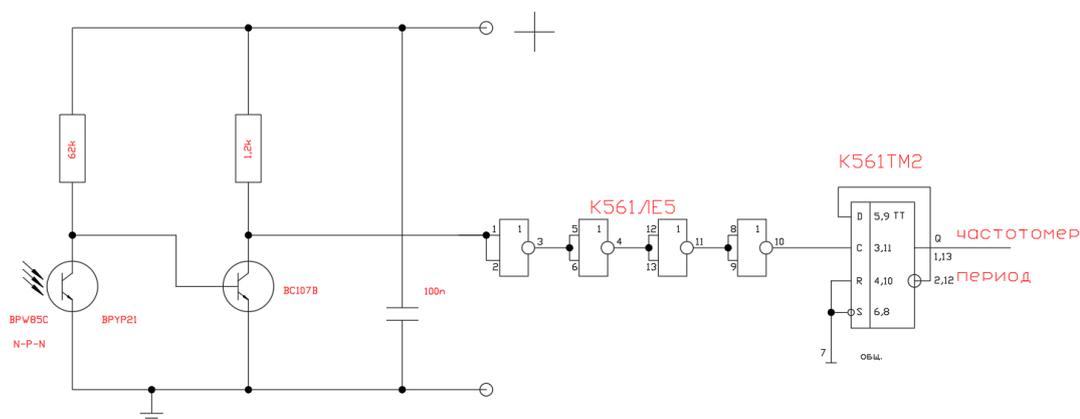


Рисунок 2 – Электрическая схема установки

К561ТМ2 имеет: два входа — *D* – информационный вход, *C* – тактовый вход; два канала — *RESET* – *R*, *SET* – *S* – каналы предустановки, причём *reset* обнуляет хранимое значение, а канал *set* устанавливает значение 1 (в представленной работе они не используются и подключены к шине *GND*).

Для того, чтобы двойной *D*-триггер выполнял функцию делителя, инверсный выход подключен к информационному входу *D*. При этом формирование сигналов осуществляется согласно схеме, представленной на временной диаграмме (рисунок 3).

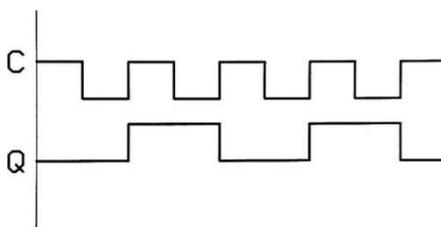


Рисунок 3 – Временная диаграмма К561ТМ2

С блока стабилизации и преобразования сигнал поступает на частотомер, который работает в режиме измерения периода импульса.

Топологическая схема разработана на базе графического редактора *AutoCAD* (рисунок 4), по ней изготовлен фотошаблон на бумажном носителе. Контактным методом фотошаблон перенесён на фольгированный однослойный стеклотекстолит. Травление проведено в химическом растворе хлорного железа. При помощи припоя и флюса произведён монтаж радиоэлементов.

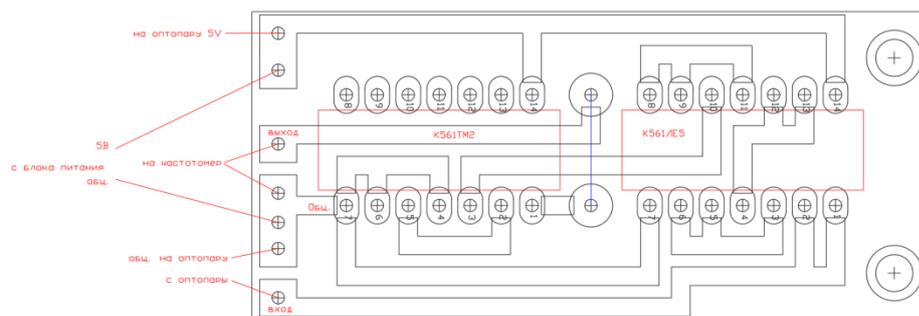


Рисунок 4 – Топологическая схема блока стабилизации и преобразования сигнала

С целью защиты технического устройства от механических повреждений и стороннего воздействия был изготовлен металлический корпус.

Заключение. Разработанное устройство способно фиксировать период колебаний с точностью вплоть до миллисекунды. Эксплуатация устройства очень проста. Техническое решение эргономично и студент без всякого труда способен использовать разработанную модернизацию. Техническая модернизация обладает высокой степенью надежности. При ее использовании достигается высокая пропускная способность учебных бригад. Подобные системы автоматизированной фиксации можно размножить и популяризировать их использование в учебных целях.

По результатам наблюдений за работой экспериментальной группы студентов, которая использовала автоматизированную систему при выполнении лабораторного опыта, установлено, что подобная разработка эффективно сказывается на динамике проведения опытов, техническое решение экономит значительное количество учебного времени, а также положительно сказывается на итоговой погрешности измерений и результативно повышает качество учебного процесса.

Указанное обстоятельство однозначно свидетельствует о целесообразности внедрения разработанной автоматизированной установки в ход выполнения лабораторных работ по разделу «Механика» курса физики.

Список использованных источников:

1. Савельев, И. В. Курс общей физики : в 5-ти кн. : кн. 1 : механика : учеб. пособие для вузов / И. В. Савельев. - М. : АСТ: Астрель, 2006. - 336 с. : ил.
2. Гусев В.Г. Электроника и микропроцессорная техника: Учеб. для вузов / В.Г. Гусев, Ю.М. Гусев. –М.: Высш.шк., 2005.
3. Манаев Е.И. Основы радиоэлектроники. –М. : Радио и связь, 1985.
4. Валенко В.С., Хандогин М.С. Электроника и микросхемотехника: Учебное пособие.–Мн.: Бестпринт, 2003.
5. Схемотехника электронных систем. Аналоговые и импульсные устройства/В.И. Бойко, А.Н. Гуржий, В.Я. Жуйков, А.А. Зори, В.М. Спивак. –СПб.: БХВ-Петербург, 2004.

UDC 534.08

AUTOMATED SYSTEM FOR RECORDING THE PERIOD OF HARMONIC OSCILLATIONS OF A PHYSICAL PENDULUM

Mikhnyuk A.I., Pashkovets M.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Supervisors: Kisel V.V. - Candidate of Physical and Mathematical Sciences, Associate Professor of the Physics Department,

Shirochin S.V. - Leading Engineer of the Physics Department

Annotation. An affordable automated system for recording the temporal characteristics of mechanical oscillations has been developed, which ensures high accuracy of results, their stability, has a high degree of reliability during operation, and allows optimising the process of conducting laboratory research.

Keywords: integrated circuit, oscillations, physical pendulum.

МАШИННОЕ ОБУЧЕНИЕ И АНАЛИЗ ДАННЫХ В ФИЗИЧЕСКИХ ИССЛЕДОВАНИЯХ

Прокофьев Н.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Григорьев А.А. – канд. физ.-мат. наук, доцент

Аннотация. Машинное обучение и анализ данных имеют широкое применение в физических экспериментах. Применение методов машинного обучения включает в себя подготовку данных, выбор модели, обучение и оценку качества модели. Методы анализа данных, такие как описательная статистика, корреляционный анализ, регрессионный анализ и анализ временных рядов, позволяют выявлять зависимости между переменными и определять их статистическую значимость. Применение методов машинного обучения и анализа данных в физическом эксперименте имеет перспективы для оптимизации экспериментов и решения научных и технологических задач.

Машинное обучение является мощным инструментом для обработки и анализа данных в физическом эксперименте. Оно позволяет автоматически находить скрытые зависимости в данных, выявлять закономерности и строить прогнозы. Применение методов машинного обучения в физическом эксперименте включает в себя следующие шаги:

Подготовка данных. На этом этапе производится очистка данных от шумов и выбросов, а также преобразование данных в удобный формат для обработки.

Выбор модели машинного обучения. В зависимости от задачи и характеристик данных выбирается модель машинного обучения, которая наилучшим образом подходит для решения поставленной задачи.

Обучение модели. На этом этапе модель обучается на обучающей выборке данных. Для этого производится минимизация функции ошибки, которая показывает расхождение между предсказаниями модели и фактическими значениями.

Оценка качества модели. После обучения модели ее необходимо оценить на тестовой выборке данных. Это позволяет оценить качество предсказаний модели и выявить ее слабые места.

Примеры применения методов машинного обучения в физическом эксперименте включают в себя распознавание частиц в адронных коллайдерах, анализ сверхпроводников и определение параметров волн в океане.

Методы анализа данных в физическом эксперименте позволяют выявить зависимости между различными переменными и определить статистическую значимость этих зависимостей. Применение методов анализа данных в физическом эксперименте включает в себя следующие шаги

Описательная статистика. Этот метод позволяет описать данные численными значениями, такими как среднее значение, медиана, стандартное отклонение и др. Это позволяет получить общее представление о данных и выявить их основные характеристики.

Корреляционный анализ. Корреляционный анализ позволяет выявить связь между различными переменными и оценить ее статистическую значимость. Этот метод может использоваться для исследования зависимостей между различными параметрами в физическом эксперименте.

Регрессионный анализ. Регрессионный анализ позволяет оценить зависимость между зависимой переменной и одной или несколькими независимыми переменными. Это может быть полезным при изучении взаимосвязей между различными параметрами в физическом эксперименте.

Анализ временных рядов. Анализ временных рядов позволяет исследовать изменения переменных во времени и выявить закономерности в их изменении. Это может быть полезным при изучении динамических процессов в физическом эксперименте.

Примеры применения методов анализа данных в физическом эксперименте включают в себя анализ данных измерений температуры, давления и скорости в атмосфере, а также анализ данных спектроскопии и рентгеновской дифракции.

Методы машинного обучения и анализа данных становятся все более важными инструментами в физическом эксперименте. Они позволяют автоматически обрабатывать и анализировать большие объемы данных, выявлять скрытые зависимости и строить прогнозы. Применение этих методов в физическом эксперименте позволяет получить новые знания о природе и улучшить качество и точность измерений.

Список использованных источников:

1. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.
2. Alpaydin, E. (2010). *Introduction to machine learning* (2nd ed.). MIT Press.
3. Bishop, C. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.
4. Higgs, P. W. (2012). *The Large Hadron Collider: past, present and future*. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1972), 741-758.
5. National Academies of Sciences, Engineering, and Medicine. (2018). *A decadal survey of the physics of the universe*. The National Academies Press.

ТУННЕЛЬНЫЙ ЭФФЕКТ ДЛЯ РАБОТЫ ТВЕРДОТЕЛЬНЫХ НАКОПИТЕЛЕЙ

Бахмат А.Д., Столбун Е.А. студенты гр.272301

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Смирнова Г.Ф. – доцент кафедры физики, канд. физ.-мат. наук, доцент

Аннотация. Данная статья посвящена практическому применению туннельного эффекта в работе твердотельных накопителей. В ней излагается суть такого квантового явления, как туннельный эффект, дается описание его работы и применения в различных сферах жизни, а также рассматривается его значимость для работы твердотельных накопителей.

Ключевые слова. Туннельный эффект, твердотельный накопитель, «память с ловушкой заряда», квантовые явления.

Актуальность темы данной работы объясняется тем, что в настоящее время твердотельные накопители стали неотъемлемой частью нашей жизни. Они используются в ноутбуках, настольных ПК и серверах. Благодаря своей скорости и надежности твердотельные накопители являются отличным вариантом для новых сборок ПК, серверов и сборщиков систем. SSD-накопитель — это носитель данных, в котором используется флеш-память для хранения данных и доступа к ним. Данные устройства имеют ряд преимуществ, ведь они долговечны, надежны и энергоэффективны, работают быстрее жестких дисков, а также имеют небольшой вес и компактный размер.

Исходя из вышеперечисленного, становятся очевидны достоинства твердотельных накопителей. Однако стоит отметить, что их работа была бы невозможна без квантовых явлений, в частности, туннельного эффекта.

Туннельным эффектом называют преодоление частицей потенциального барьера в случае, когда её энергия (остающаяся при этом неизменной) меньше высоты барьера. Квантовое туннелирование или туннелирование относится к квантово-механическому явлению, когда частица туннелирует через барьер, который она не могла преодолеть в классическом понимании [1].

В 20–30-х годах нашего века во время появления квантовой механики, появилось предположение о существовании туннельного эффекта. В дальнейшем за счет туннельного эффекта были объяснены некоторые важные явления, обнаруженные экспериментально в различных областях физики.

Туннельный эффект – квантово-механический эффект, не имеющий альтернативы в классической механике. Это основная особенность туннельного эффекта. Если опираться на классическую механику, понятно, что любое материальное тело, которое обладает энергией E , не может преодолеть потенциальный барьер высотой V_0 , если $V_0 > E$.

В соответствии с законом сохранения энергии при падении на барьер тело отражается от него. Для электрона всё происходит иначе. Ему присущи как корпускулярные, так и волновые свойства

Туннелирование часто объясняется с использованием принципа неопределенности Гейзенберга и корпускулярно-волнового дуализма материи. При ширине потенциального барьера $R \leq \lambda_D$, где λ_D длина волны де Бройля для материального тела, электрон при падении на барьер с некоторой вероятностью окажется с другой его стороны, он пройдет через барьер без изменений энергии. В этом качественно заключается смысл туннельного эффекта. Это объясняет работу туннельного эффекта.

В 20-х годах XX века независимо Г.А. Гамовым, а также Е. Кондоном и Р. Герни была выдвинута туннельная теория α -распада. Это позволило достаточно точно описать периоды распада различных ядер в диапазоне их величин. Без туннельного эффекта термоядерные реакции были бы невозможны. Чтобы произошло слияние, ядра, участвующие в реакции, должны приблизиться друг к другу. Однако им препятствует кулоновский потенциальный барьер, который частично преодолевается за счет высоких скоростей (высокая температура ядер) и отчасти в результате туннельного эффекта. В начале XI века успех теории туннелирования α -частиц из ядер доказал справедливость основ новой квантовой физики.

Туннельный эффект представлен и в оптике. Если угол падения волны больше предельного угла, происходит полное отражение, так же как и при отражении частицы от потенциального барьера при $E < V$ в рамках классической механики. В результате экспериментов, физики выяснили, что свет проникает за границу раздела на глубину порядка длины волны, экспоненциально ослабляясь во второй среде. Наиболее наглядным является опыт одного из основателей отечественной радиофизики, Л.И. Мандельштама (1879–1944), в котором он изучал отражение света от

поверхности люминесцирующего раствора. Наглядным примером является эксперимент Л.И. Мандельштама, где он исследовал отражение света от поверхности люминесцирующего раствора. Проникновение света в раствор наблюдали по возникновению свечения в тонком приповерхностном слое раствора.

Многие примеры туннельного эффекта можно найти в физике твердого тела. Таким образом, полевая эмиссия, т. е. эмиссия электронов из металла или полупроводника под действием сильного электрического поля, происходит путем туннелирования. Туннельный эффект объясняет различные явления в полупроводниках, помещенных в сильное электрическое поле. В эффекте Джозефсона пары электронов туннелируют через тонкий изолирующий барьер между двумя сверхпроводящими материалами [2].

Сказанное позволяет заключить, что туннельный эффект играет существенную роль в самых различных областях физики и техники. Более того, данное квантовое явление обеспечивает работу твердотельных накопителей.

Для того, чтобы иметь представление о взаимосвязи туннельного эффекта и работы твердотельных накопителей, необходимо понимать, как кодируется и хранится информация. В современных компьютерах информация хранится в виде данных, зашифрованных в виде нулей и единиц. Что же касается твердотельных накопителей, принцип их работы несколько иной. Твердотельный накопитель состоит из кремниевых чипов, внутри которых находится множество ячеек памяти. Каждая ячейка, по сути, представляет собой «память с ловушкой заряда» [3]. «Память с ловушкой заряда» — это технология полупроводниковой памяти, которая используется при создании энергонезависимой NOR и NAND флэш-памяти. Когда на твердотельный накопитель записывается информация, «память с ловушкой заряда» заполняется электронами. Во многих современных твердотельных накопителях каждая «ловушка заряда» использует свой собственный уровень заряда электрона для представления 3-х битов информации.

Поскольку запись информации на твердотельный накопитель подразумевает заполнение «памяти с ловушкой заряда» электронами, обратимся к тому, как это происходит. «Память с ловушкой заряда» состоит из регулирующего затвора, канала, слоя диэлектрика и непосредственно «ловушки заряда», представленных на рисунке 1.

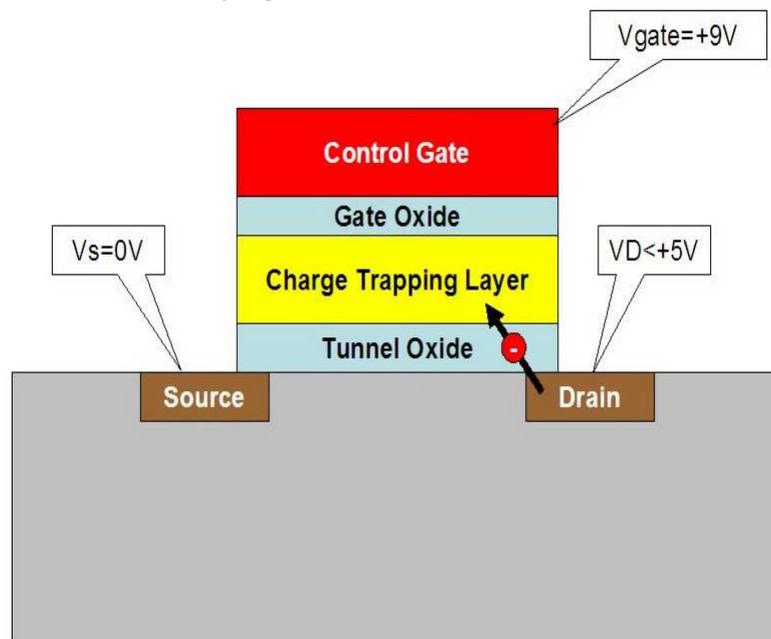


Рисунок 1 – устройство «памяти с ловушкой заряда»

Источником электронов является канал. Диэлектрик, в свою очередь, разделяет канал и «ловушку заряда», так как его электропроводность мала, а значит электронам из канала трудно попасть в «ловушку заряда».

Именно здесь и происходит туннельный эффект. Когда на затвор над слоем оксида действует напряжение, электрическое поле перемещает электроны из канала к затвору. При применении подходящего напряжения, электроны, находящиеся в канале, могут при помощи туннельного эффекта переместиться через затвор в «ловушку заряда». Как только электроны попадают в «ловушку заряда», они остаются там.

Каждый раз, когда на твердотельный накопитель записывается информация, происходят вышеупомянутые процессы. Квантовое туннелирование заполняет «ловушки заряда», расположенные в ячейках памяти устройства.

Однако стоит подчеркнуть, что результат работы квантового туннелирования сильно зависит от ширины затвора. Если диэлектрические стенки слишком плотные, напряжение, которое нужно применить для возникновения туннельного эффекта, будет слишком высоким. Чем выше применимое к ячейке памяти напряжение, тем больше ущерба ей наносится. В случае, если диэлектрические стенки слишком тонкие, электроны могут туннелироваться из «ловушки заряда». В итоге записанные на твердотельном накопителе данные будут повреждены.

Таким образом, туннельный эффект играет существенную роль в самых различных областях физики и техники. Одной из главных особенностей туннельного эффекта является его безальтернативность в классической механике, так как он является принципиально квантово-механическим эффектом. На своём примере туннельный эффект доказывает основополагающее положение квантовой механики – корпускулярно-волновой дуализм свойств элементарных частиц. Также из всего вышперечисленного можно заключить, что без туннельного эффекта была бы невозможна работа популярных на сегодняшний день твердотельных накопителей.

Список использованных источников:

1. Эффект туннелирования и квантовая механика / А. Ахмаров // Сборник статей Международной научно-практической конференции МЦИИ ОМЕГА САЙНС 22 августа 2017 г., 2017. – С. 14-16.
2. Quantum Tunnelling in Small-Capacitance Josephson Junctions in a General Electromagnetic Environment / G. Falci [et al.] Delft University of Technology, 1991. – P. 2-4.
3. Solid state drives data reliability and lifetime / A. R.Olson [et al.] // Imation White Paper, 2008. – P. 15-16.

UDC 530.145

TUNNEL EFFECT FOR SOLID STATE DRIVES

Stolbun E.A., Bahmut A.D.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Smirnova G. F. – assistant professor, PhD in Physics and Mathematics, associate professor of the department of physics

Annotation. This article is devoted to the practical application of the tunnel effect in the operation of solid state drives. It outlines the essence of such a quantum phenomenon as the tunnel effect, describes its work and application in various areas of life, and also considers its significance for the operation of solid-state drives.

Keywords. Tunnel effect, solid state drive, "charge trap memory", quantum phenomena.

ОПТИМИЗАЦИЯ ФИЗИЧЕСКОГО ЭКСПЕРИМЕНТА С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ И АНАЛИЗА ДАННЫХ НА ПРИМЕРЕ ЭФФЕКТА ЗЕЕБЕКА

Глинский И.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Григорьев А.А. – канд. физ-мат. наук, доцент

Аннотация. В данной работе исследуется применение машинного обучения и анализа данных для оптимизации эксперимента по изучению эффекта Зеебека. Цель состоит в разработке методики определения оптимальных параметров термопар для максимизации коэффициента Зеебека. Используя методы кластеризации, предобработки данных и машинного обучения, успешно выявлены закономерности в экспериментальных данных, что позволило определить оптимальные условия для проведения эксперимента. Результаты данного исследования могут быть полезны для разработки новых материалов и устройств, основанных на эффекте Зеебека.

Эффект Зеебека (также известный как термоэлектрический эффект) - это физический явление, при котором температурная разница между двумя разными материалами, соединенными в контактные точки, приводит к возникновению электродвижущей силы (напряжения). Это явление было впервые обнаружено Томасом Зеебеком в 1821 году и с тех пор находит широкое применение в различных областях науки и техники, особенно

Ключевым параметром, характеризующим эффект Зеебека, является коэффициент Зеебека. Его значение зависит от свойств материала и температурного градиента.

$$E = \alpha(T_2 - T_1) \quad (1),$$

Где E - коэффициент Зеебека; α - термоэлектрическая способность пары, T_2, T_1 - температуры контактов.

Одним из наиболее распространенных применений эффекта Зеебека являются термопары - простые и недорогие устройства для измерения температуры. Термопары состоят из двух проводников из разных материалов, соединенных на одном конце. При нагреве этого соединительного конца возникает термоэлектрическое напряжение, которое можно измерить и преобразовать в соответствующее значение температуры.

В данной работе был проведен анализ данных по семи различным типам термопар (Т, К, J, Е, В, R и S) с целью оптимизации их использования. В ходе исследования были применены различные методы машинного обучения и кластеризации, такие как линейная регрессия, случайный лес, градиентный бустинг и метод опорных векторов, метод K-средних и агломеративная иерархическая кластеризация.

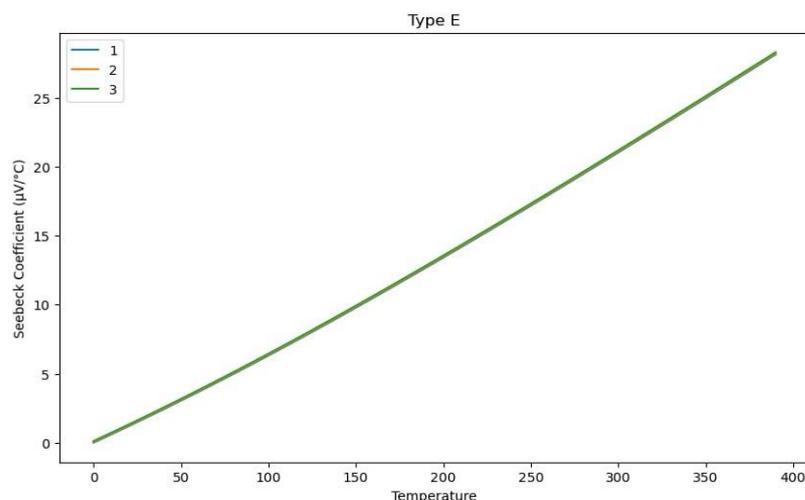


Рисунок 1 – график зависимости коэффициента Зеебека от разности температур термопары типа E

Для нахождения коэффициента Зеебека были использованы следующие методы: линейная регрессия, случайный лес, градиентный бустинг и метод опорных векторов. Производительность этих моделей оценивалась с использованием показателей среднеквадратичной ошибки (MSE) и коэффициента детерминации (R-squared), а также перекрестной проверки для оценки надежности каждой модели. Результаты были следующими:

Линейная регрессия показала наилучшие результаты с MSE 0,22 и значением R-squared 0,9996. Результаты перекрестной проверки показали MSE 4,43 и значение R-squared 0,9511, что указывает на высокий уровень согласованности и точности.

Кроме того, были применены методы кластеризации (обучение без учителя), такие как K-средних и агломеративная иерархическая кластеризация, для группировки термопар по их характеристикам и поведению в разных температурных диапазонах. Эти методы позволяют определить сходства между различными типами термопар и выделить группы с похожими свойствами.

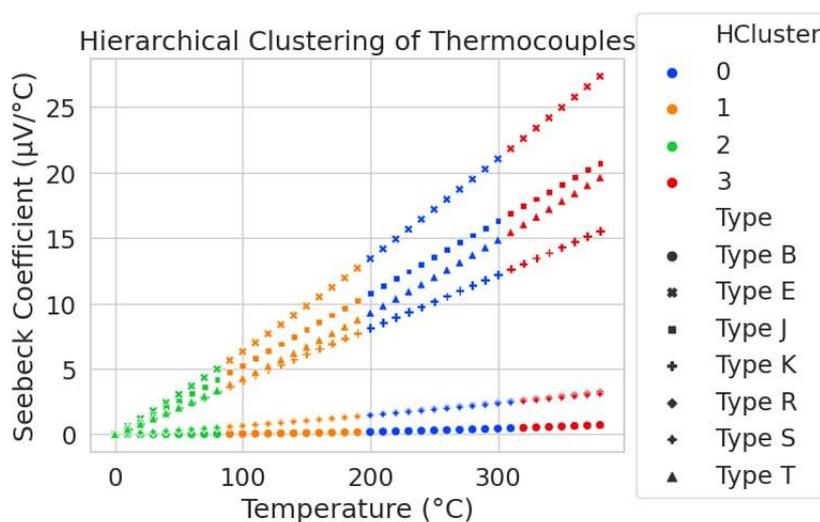


Рисунок 2 – иерархическая кластеризация термопар.

Агломеративная иерархическая кластеризация выявила четыре кластера, сгруппированные в соответствии с температурными диапазонами и коэффициентами Зеебека. В частности, первый кластер включает термопары, которые демонстрируют схожие характеристики при температуре в диапазоне 1 кластера. Это свидетельствует о том, что в определенных диапазонах температур разные типы термопар могут вести себя аналогично.

Таким образом, можно сделать вывод о том, что в некоторых случаях использование более дешевых термопар может быть предпочтительным, так как они ведут себя аналогично более дорогим моделям. Например, если наше устройство работает в диапазоне 1 кластера, то имеет смысл выбирать более экономичные варианты термопар, так как они обеспечивают сопоставимую точность и характеристики.

Помимо этого, важно отметить, что результаты данного исследования могут быть использованы не только для определения наиболее подходящих термопар, но и для разработки новых материалов и термоэлектрических устройств с оптимальными характеристиками. Ведь понимание закономерностей и связей между температурой и коэффициентом Зеебека может способствовать созданию новых термоэлектрических материалов с повышенной эффективностью и надежностью.

Результаты данного исследования могут быть использованы для оптимизации выбора термопар в зависимости от конкретных условий их применения. Применение методов машинного обучения и кластеризации позволяет выявить закономерности и сходства между различными типами термопар, что может помочь пользователям принимать обоснованные решения при выборе наиболее подходящего типа термопары для своих нужд.

Список использованных источников:

1. Савельев И. В., Курс физики : учеб. В 3 т. Т. 3: Квантовая оптика. Атомная физика. Физика твердого тела. Физика атомного ядра и элементарных частиц / И. В. Савельев. – М. : Наука. Гл. ред. физ.-мат. лит., 1989. – 304 с.
2. Епифанов, Г. И. Физика твердого тела : учеб. пособие / Г. И. Епифанов. – 4-е изд., стер. – СПб. : Издательство «Лань», 2011. – 288 с.

КВАНТОВОЕ ЯВЛЕНИЕ СВЕРХПРОВОДИМОСТИ

Панизник А.С, Ивановская Е.О., студенты гр. 272303

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Смирнова Г.Ф. – доцент кафедры физики, канд. физ.-мат. наук, доцент

Аннотация. Данный доклад представляет собой результат изученных характеристик и особенностей сверхпроводников, описываются сферы их применения, эффект Мейснера-Оксенфельда.

Ключевые слова. Сверхпроводимость, критический ток, критическая температура, эффект Мейснера, сверхпроводник первого рода, сверхпроводник второго рода, высокотемпературный сверхпроводник.

В 1911 году голландский физик и химик Хейке Каммерлинг Оннес наблюдал явление сверхпроводимости, при котором некоторые вещества и материалы имеют нулевое электрическое сопротивление при снижении температуры до критической. Сверхпроводимость — это явление, при котором электрическое сопротивление твердого тела снижается до нуля, а магнитное поле выталкивается из его внутренней части.

Существуют параметры, которые характеризуют сверхпроводник на границе перехода от сверхпроводящего состояния в нормальное и наоборот: критический ток, температура фазового перехода и критическое магнитное поле.

Критический ток в сверхпроводниках – предел устойчивого тока, который не затухает в сверхпроводящем материале, если ток поднимается выше предела, то материал переходит в непроводящее состояние. Если сверхпроводящий материал имеет ограниченное электрическое сопротивление в нормальном проводящем состоянии, то переход в сверхпроводящее состояние сопровождается диссипацией энергии тока и вызывает нагрев.

Температурный интервал, при котором проводник переходит в сверхпроводящее состояние, имеет определенное значение T_c (температура фазового перехода) [1]. В дальнейшем это значение будет называться просто критической температурой.

Сопротивление обычного металла как функция температуры подчиняется закону $R(T)$:

$$R_2 = R_1 * (1 + \alpha(T_2 - T_1)) \quad (1),$$

В сверхпроводниках сопротивление становится равным нулю при достижении критической температуры. Разница между сверхпроводниками и обычными металлами показана на рисунке 1.



Рисунок 1 – Отличие обычных металлов от сверхпроводников

С самого начала применение сверхпроводников было очень ограниченным. Причина в том, что температура их использования не превышает 20 К (эквивалентно -253 °С). Сверхпроводники теряют свойство нулевого электрического сопротивления, когда температура превышает критическое значение. Поэтому использование сверхпроводников требует потребления большого количества энергии для их охлаждения, что весьма проблематично и невыгодно.

В феврале 1987 года Г. Беднорц и К. Мюллер из Швейцарии синтезировали в США первый высокотемпературный сверхпроводник [2]. Они продемонстрировали критические температуры,

которые могут быть достигнуты при использовании жидкого азота в качестве криогенной жидкости: соединение Nb₃Ge достигло самой высокой критической температуры из когда-либо достигнутых (около 23 К). Такие сверхпроводники, критическая температура которых выше точки кипения жидкого азота (77 К), были названы высокотемпературными.

Важной особенностью сверхпроводимости является эффект Мейсснера (эффект Мейсснера-Оксенфельда) – смещение магнитного поля из сверхпроводящего объема при переходе в сверхпроводящее состояние [3]. Это явление было открыто немецкими физиками Вальтером Мейсснером и Робертом Оксенфельдом в 1933 году. Они измерили распределение магнитного поля за пределами сверхпроводящих образцов свинца и олова. В своих экспериментах они охлаждали сверхпроводники ниже критической температуры под действием приложенного магнитного поля и тем самым уменьшали внутреннее магнитное поле образца до нуля. Сверхпроводники с небольшим магнитным полем или без него остаются в состоянии Мейсснера. Однако, если приложенное магнитное поле слишком велико, это состояние разрушается. Сверхпроводники можно разделить на два типа в зависимости от того, как происходит этот распад. В первом типе сверхпроводников сверхпроводимость внезапно прерывается, когда интенсивность приложенного магнитного поля превышает критическое значение. В зависимости от типа твердого тела может возникнуть промежуточное состояние, в котором смешиваются нормальное состояние и область сверхпроводящего материала без магнитного поля [4]. В сверхпроводниках типа II при увеличении напряженности приложенного магнитного поля до первого критического значения увеличивается величина проникновения магнитного потока, но если ток не слишком велик, то возникает смешанное состояние (также называемое вихревым), в котором не остается сопротивления току. При втором критическом значении интенсивности сверхпроводящее состояние разрушается. Это смешанное состояние вызывается вихрем в сверхпроводящей электронной жидкости и иногда называется флюксоном-квантом магнитного потока, поскольку магнитный поток, переносимый этим вихрем, может быть измерен.

Самые чистые элементарные сверхпроводники, кроме ниобия и углеродистых нанотрубок, являются сверхпроводниками первого типа, и почти все примесные и сложные сверхпроводники относятся ко второму типу [5].

Критическое магнитное поле в сверхпроводниках – это характерное значение напряженности постоянного магнитного поля, превышая которое вызывает проникновение магнитного поля с поверхности в глубь сверхпроводника на расстояние, превышающее глубину проникновения магнитного поля; происходит плавный переход материала из состояния сверхпроводимости в нормальное. Критическое магнитное поле зависит от свойств материала сверхпроводника и температуры [6]. В сверхпроводниках первого рода проникновение поля происходит в результате фазового перехода, а критическое магнитное поле H_c – это напряженность, при которой плотность магнитной энергии равна разности плотностей свободных энергий в нормальном и в сверхпроводящем состояниях.

Применение сверхпроводников разнообразно и распространено. С их помощью можно получить сильные токи, используя источник, который имеет небольшое напряжение. При этом практически отсутствуют тепловые потери, что позволяет использовать сверхпроводник измерительных приборах. Сильноточные технологии, которые предназначаются для устройств больших мощностей, применяются в электроэнергетике, промышленности и на транспорте [7]. Применение сверхпроводников принято делить на всевозможные материалы (пленочные проводники, сверхпроводящие магниты и т.д.), микротехника (микроволновые приборы, сверхчувствительные системы обнаружения магнитных полей, цифровая электроника, искусственные биологические системы), макротехника (электронные системы и сети, генераторы и движки).

Список использованных источников:

1. Wikipedia.org [Электронный ресурс]. Режим доступа: <https://ru.wikipedia.org/wiki/Сверхпроводимость/> - дата доступа: 09.04.2023.
2. Electric.info [Электронный ресурс]. Режим доступа: <http://elektrik.info/main/fakty/1017-vysokotemperaturnaya-sverhprovodimost.html/> - дата доступа: 11.04.2023.
3. Гинзбург В. Л., Андришин Е. А. Сверхпроводимость. – М., 1990. – 112 с
4. Сверхпроводимость [Электронный ресурс]. – Режим доступа: <https://www.booksite.ru/fulltext/1/001/008/100/163/> – Дата доступа: 08.04.2023.
5. СВЕРХПРОВОДИМОСТЬ - Энциклопедия Кругосвет [Электронный ресурс]. – Режим доступа: https://www.krugosvet.ru/enc/nauka_i_tehnika/tehnologiya_i_promyshlennost/SVERHPROVODIMOST/ – Дата доступа: 08.04.2023.
6. Применение сверхпроводимости в науке и технике [Электронный ресурс]. – Режим доступа: <http://electricalschool.info/spravochnik/poleznoe/2320-primeneniye-sverhprovodimosti-v-nauke-i-tehnike/> – Дата доступа: 08.04.2023.
7. Ципенюк Ю. М. Физические основы сверхпроводимости. – Магадан, 2012. – 94 с.

УДК 534.08

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ИЗМЕРЕНИЯ КОЛЕБАНИЙ БАЛЛИСТИЧЕСКОГО МАЯТНИКА

Чубса А.А., Ковшер А.О., Усович И.В.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

*Научные руководители: Кисель В.В. – канд. физ.-мат. наук, доцент кафедры
физики*

Пышинский Ч.И. – вед. инж. кафедры физики

Аннотация. Создана установка, позволяющая измерять время требуемого количества полных колебаний баллистического маятника, обеспечивающая высокую точность результатов измерений, их стабильность и простоту эксплуатации.

Ключевые слова: колебания, баллистический маятник

Введение. Точность результатов измерений, получаемых в ходе выполнения лабораторной работы «Измерение скорости пули с помощью баллистического маятника», как правило, низкая, что вызвано сложностью измерений и пр.

В статье предлагается способ снятия показаний, который позволяет получать значительно более точный результат с относительно малыми затратами по времени. Это, в свою очередь, позволяет повысить уровень учебного процесса и упростить задачу студентам.

Основным преимуществом установки является простая возможность выбора требуемого количества измеряемых полных колебаний маятника. Для этого, лицу, выполняющему лабораторную работу, необходимо нажать кнопку СТОП, когда индикатор установки показывает число полных колебаний на единицу меньше требуемого. Отсчет времени на установке прекращается в момент завершения установленного ранее количества полных колебаний и фиксируется как их окончательная продолжительность, так и общее число полных колебаний.

Основная часть. Измеритель FPM -14 предназначен для определения количества полных колебаний баллистического маятника (Лабораторная работа «Измерение скорости пули с помощью баллистического маятника»), и времени, которое они занимают.

При создании автоматизированной системы измерения гармонических колебаний баллистического маятника учтены следующие требования, как:

- Надёжность и безопасность;
- Стабильность результатов;
- Точность и простота измерений.

Для реализации автоматического фиксирования колебаний баллистического маятника выбран вариант с использованием оптопары. Это оказалось выгодным решением. Оптический метод прост в понимании и основан на прерывании оптического луча стержнем баллистического маятника.

Работает он следующим образом:

Световой поток с лампочки попадает на фототранзистор. Во время колебаний баллистического маятника, стержень маятника прерывает световой поток, в результате чего в схеме транзистора Т1 генерируются электрические импульсы, которые после усиления в схеме транзистора Т2 подводятся к входу миллисекундомера. Когда световой поток падает на фототранзистор Т1, транзистор Т2, в свою очередь, электрические импульсы не проводит.

Когда стержень перекрывает оптический луч, автоматически начинается отсчёт, и после каждого полного периода колебаний показание счётчика периодов увеличивается на единицу. Что позволяет отслеживать сколько полных периодов колебаний совершил маятник.

На рисунке.1 представлена блок-схема автоматизированной системы измерения гармонических колебаний баллистического маятника:

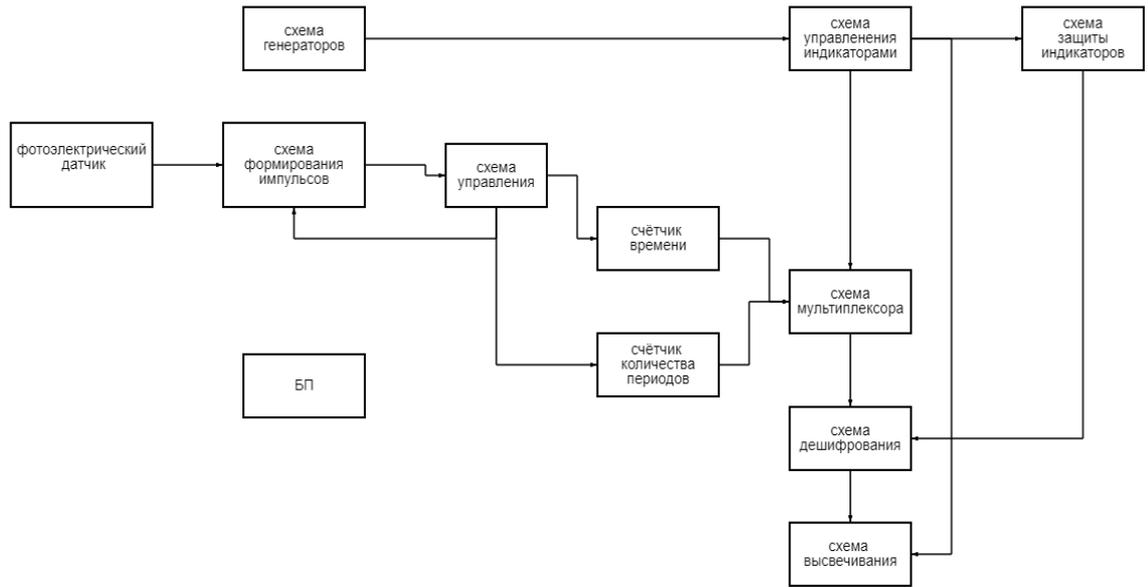
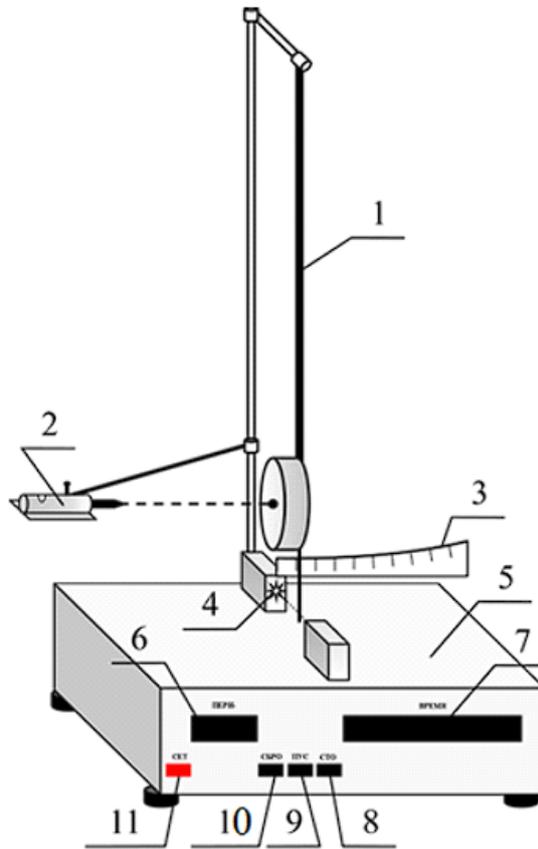


Рис. 1—блок-схема работы установки

Схема установки.



- Баллистический маятник.
- Пружинный пистолет с пулей.
- Линейка.
- Фотоприёмник.

- Электронный таймер.
- Индикатор ПЕРИОД.
- Индикатор ВРЕМЯ.
- Кнопка СТОП.
- Кнопка ПУСК.
- Кнопка СБРОС.
- Кнопка СЕТЬ.

Принцип работы.

После включения измерителя FPM -14 в сеть и нажатия кнопки СЕТЬ, схема сброса, построенная из резисторов R25 , R26 , диодов D9 , 010, конденсатора С9 и вентиля М6, устанавливает миллисекундомер в начальное состояние, обнуляя декады и сбрасывая триггеры. Схема формирования импульсов продолжает быть заблокированной (низкий уровень на входе 10 вентиля). Такое состояние сохраняется до момента нажатия переключателя W2 СБРОС. В этот момент наступает деблокирование схемы формирования импульса (высокий уровень на входе 10 вентиля М1), а первый нарастающий фронт сигнала, приходящего с фотоэлектрического датчика, открывает клапан, блокирующий импульсы со схемы генератора (высокий уровень на входе 4 вентиля М4)/. Очередные нарастания фронтов импульсов с фотоэлектрического датчика в схеме триггера заменяются на сигналы, являющиеся отражением количества полных колебаний маятника. Итак, второй входной импульс устанавливает на выходе 9 триггера М2 высокий уровень, который не изменится уже до конца измерений или нажатия кнопки СБРОС. С этого момента, после снижения первого входного импульса, клапан 5 (высокий уровень на входе 12) готов пропускать каждый второй сигнал. Деление на два реализовано на триггере М3, выход 12.

Процесс подсчёта продолжается до момента нажатия кнопки СТОП.

Импульс СТОП, после формирования в триггере R5(реализованном на вентилях М6), деблокирует клапан М5(высокий уровень на входе 1). Первый импульс, после нажатия переключателя СТОП останавливает генератор и блокирует схему формирования входных импульсов. Когда клавиша ПУСК находится в рабочем положении, на контакты 6, 7 разъёма ZL1 передаётся напряжение 14В. Нажатие переключателя СТОП вызывает отключение питающего напряжения, К зажимам 6, 7 разъёма ZL1 подключается в этом положении только диод D15, предназначенный для подавления перенапряжений на установке.

Заключение. Разработанное устройство фиксирует период колебаний с точностью вплоть до миллисекунды. Эксплуатация устройства проста и надежна. Техническое решение эргономично. При использовании модернизированной установки достигается высокая пропускная способность учебных бригад. Подобные системы автоматизированной фиксации можно размножить и популяризировать их использование в учебных целях.

По результатам наблюдений за работой экспериментальной группы студентов, которая использовала автоматизированную систему при выполнении лабораторного опыта, установлено, что подобная разработка эффективно сказывается на динамике проведения опытов, техническое решение экономит значительное количество учебного времени, а также положительно сказывается на итоговой точности измерений и результативно повышает качество учебного процесса.

Указанное обстоятельство однозначно свидетельствует о целесообразности внедрения разработанной автоматизированной установки в ход выполнения лабораторных работ по разделу “Механика” курса физики.

Список использованных источников:

- Савельев, И. В. Курс общей физики : в 5-ти кн. : кн. 1 : механика : учеб. пособие для втузов / И. В. Савельев. - М. : АСТ: Астрель, 2006. - 336 с. : ил.
- Гусев В.Г. Электроника и микропроцессорная техника: Учеб. для вузов / В.Г. Гусев, Ю.М. Гусев. –М.: Высш.шк., 2005.
- Манаев Е.И. Основы радиозлектроники. –М. : Радио и связь, 1985.
- Валенко В.С., Хандогин М.С. Электроника и микросхемотехника: Учебное пособие.–Мн.: Бестпринт, 2003.
- Схемотехника электронных систем. Аналоговые и импульсные устройства/В.И. Бойко, А.Н. Гуржий, В.Я. Жуйков, А.А. Зори, В.М. Спивак. –СПб.: БХВ-Петербург, 2004.

UDC 534.08

AUTOMATED SYSTEM FOR MEASURING OSCILLATIONS OF A BALLISTIC PENDULUM

Chubsa A.A., Kovsher A.O., Usovich I.V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

V.V. Kisel – Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Physics

Pyshinsky Ch.I. – Leading Engineer of the Department of Physics

Annotation. was created a setup that makes possible to measure the time of several oscillations of a ballistic pendulum, providing a low measurement error, their stability and ease of operation.

Keywords. oscillations, ballistic pendulum

СЕКЦИЯ 6
«ВЫСШАЯ МАТЕМАТИКА»

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО АНАЛИЗА

Дроздов А.И., студент гр.250502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Луцакова И. Н. – канд. физ.-мат. наук

Аннотация. В связи с быстрым ростом нейронных сетей в этом году данная работа проводит сравнительный анализ трех самых популярных решений в прикладных задачах математического анализа. Сравниваются ChatGPT (GPT-4), Google Bard и Bing Chat по введённой системе оценки.

Ключевые слова. Нейронные сети, математический анализ, chatgpt, google bard, bing chat.

В последнее время наблюдается бум в развитии нейронных сетей (основанных на технологии GPT – Generative Pre-Trained Transformer). В связи с постоянным потоком новостей о невероятных результатах этих нейронных сетей в совершенно разных сферах [5], цель этой работы – проанализировать самые популярные из решений на задачах математического анализа.

Что такое нейронные сети? Нейронные сети – это метод в искусственном интеллекте, который учит компьютеры обрабатывать данные таким же способом, как и человеческий мозг. Используется множество слоев, начиная от входного, проходя через огромное количество промежуточных слоев, и до выходного. Нейронная сеть способна учиться на своих ошибках на предыдущих слоях.

Что такое GPT, который лежит в основе всех нейронных сетей, рассматриваемых в этой работе? GPT – это особый вид трансформера (модель глубокого обучения, которая по-разному взвешивает каждую часть входных данных и анализирует каждую часть данных с полным контекстом из предыдущих данных), которому не нужно обучение с учителем. Это позволяет гораздо быстрее тренировать сети на гораздо больших наборах данных. Обучение состоит из двух этапов: обучение и настройка параметров на большом объеме необработанных данных, а затем более точная “подкрутка” под финальный результат под наблюдением.

Было выбрано 3 решения: ChatGPT (GPT-4) [2], Google Bard [3] и Bing Chat [4]. Использовались примеры из сборника задач [1].

Для сравнения была создана своя шкала оценки: каждый пример оценивается по 3 критериям (от 1 до 10): корректность (правильно ли решен пример), рациональность (можно ли было его решить проще) и полнота объяснения (насколько бы было понятно решение студентам).

Для оценки результатов использовались следующие формулы:

$$S_i = [(correct_i + rational_i + clear_i)/3], \quad (1)$$

где S_i – оценка нейронной сети по конкретному примеру, а $correct_i$, $rational_i$, $clear_i$ – оценка примера по корректности, рациональности и полноте объяснения соответственно.

$$S = \left[\frac{\sum_{i=1}^n S_i}{n} \right], \quad (2)$$

где S – итоговая оценка нейронной сети, а n – число примеров.

В работе приведены наиболее характерные скриншоты ответов и их краткая характеристика.

Теория вещественных чисел и множеств

1. Доказать, что $1^2 + 2^2 + \dots + n^2 = \frac{n \cdot (n+1) \cdot (2n+1)}{6}$

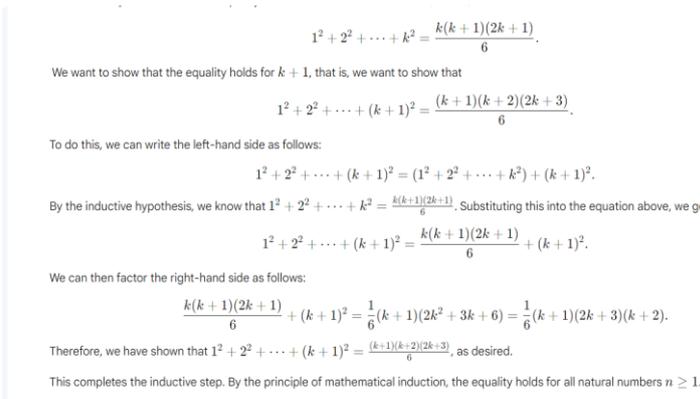
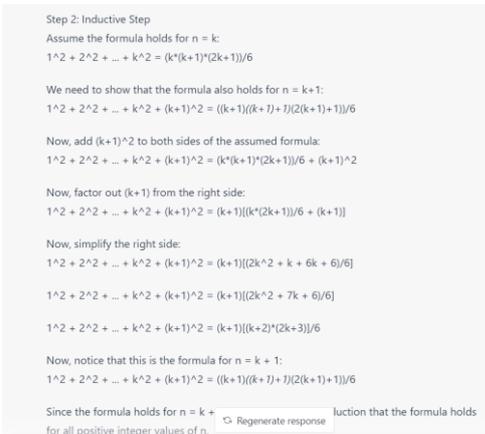


Рисунок 1 – Ответ ChatGPT

Рисунок 2 - Ответ Google Bard

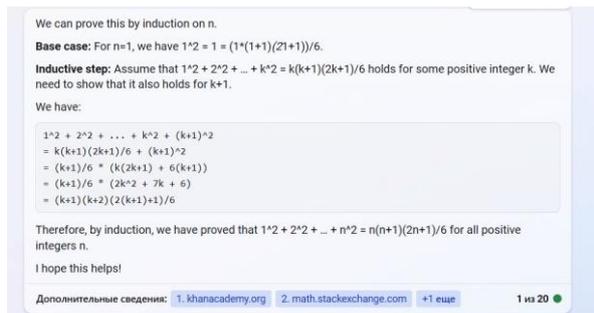


Рисунок 3 – Ответ Bing Chat

Как видно из скриншотов, все сети смогли решить данный пример. Однако, в форматировании ответа и объяснениях были некоторые проблемы.

Например, в проверке базы индукции ChatGPT вывело 21 вместо $2 \cdot 1$ (наверное, подставляя 1 вместо n , не заметив, что получилось другое число). Также использовались квадратные скобки вместо круглых (такая проблема есть в старых версиях MathCAD), не использовался отформатированный вывод. Объяснение даже слишком подробное (обычно нет нужды описывать словами каждый шаг, только если это не что-то неочевидное).

Google Bard имеет хорошее и рациональное решение. По объему объяснений не слишком много и не слишком мало – золотая середина. Однако он допустил ошибку в конце доказательства:

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

он сказал, что $\frac{1}{6}(k+1)(2k^2+3k+6) = \frac{1}{6}(k+1)(2k+3)(k+2)$, однако правильно $\frac{1}{6}(k+1)(2k^2+7k+6) = \frac{1}{6}(k+1)(2k+3)(k+2)$, что очень похоже на “подгонку под ответ”.

Bing Chat же также имел проблемы с форматированием $2*1$, но решил правильно. Объяснение довольно краткое и может запутать изначально.

Оценка (correct/rational/clear):

ChatGPT: 10/9/6

Google Bard: 7/10/10

Bing Chat: 10/10/8

2. Доказать, что $n! < \left(\frac{n+1}{2}\right)^n$ при $n \geq 2$

ChatGPT решил пример правильно, однако не очень рационально. Все ещё наблюдались проблемы с форматированием (если отправить пример несколько раз, то иногда ответ будет с форматированием, а иногда без).

Google Bard допустил критические ошибки, которые невозможно допустить при обычных условиях, доказательство неверное.

Bing Chat начинал решать правильно, но в итоге также “подстроился под ответ”. Из источников, которые он предлагал, были вопросы на похожие примеры, однако не на текущий. Таким образом, он пытался действовать аналогично, но каждый пример особенный по-своему.

The base case is $n = 2$. In this case, we have:

$$2! = 2 < \left(\frac{2+1}{2}\right)^2 = \frac{9}{4}$$

Therefore, the statement holds for the base case $n = 2$.

Now, assume that the statement holds for some natural number $k \geq 2$, i.e.,

$$k! < \left(\frac{k+1}{2}\right)^k$$

We will show that the statement also holds for $k+1$, i.e.,

$$(k+1)! < \left(\frac{k+2}{2}\right)^{k+1}$$

We can write $(k+1)!$ as follows:

$$(k+1)! = k!(k+1)$$

By the inductive hypothesis, we know that $k! < \left(\frac{k+1}{2}\right)^k$. Therefore, we have:

$$(k+1)! < k!(k+1) < \left(\frac{k+1}{2}\right)^k (k+1) = \left(\frac{k+2}{2}\right)^{k+1}$$

This proves that the statement holds for $k+1$.

By the principle of mathematical induction, the statement holds for all natural numbers $n \geq 2$.

Рисунок 4 – Google Bard заменяет знак “=” на “<”

Для доказательства данного неравенства мы будем использовать метод математической индукции.

База индукции: При $n = 2$ неравенство верно, так как $2! = 2 < \left(\frac{2+1}{2}\right)^2 = \frac{9}{4}$.

Шаг индукции: Пусть неравенство верно для некоторого $n \geq 2$, тогда $(n+1)! = (n+1)n! < (n+1)\left(\frac{n+1}{2}\right)^n$ (по предположению индукции) $< \left(\frac{n+2}{2}\right)^{n+1}$. Таким образом, неравенство верно и для $n+1$.

Таким образом, мы доказали данное неравенство методом математической индукции.

Дополнительные сведения: [1. math.stackexchange.com](https://math.stackexchange.com) [2. khanacademy.org](https://khanacademy.org) [3. math.libretexts.org](https://math.libretexts.org) — показать меньше

1 из 20 ●

Рисунок 5 –Bing Chat использует некорректный источник

Оценка (correct/rational/clear):

ChatGPT: 10/8/10

Google Bard: 1/1/7

Bing Chat: 5/7/2

Предел числовой последовательности

3. $\lim_{n \rightarrow \infty} \frac{10000n}{n^2+1}$

ChatGPT решил не самым рациональным способом (поделить на старшую степень), однако правильно.

Google Bard сказал, что предел не существует, добавив для доказательства код на Asymptote Graphics Language (который не запустился).

Bing Chat решил правильно, однако применил правило Лопиталю для этого простого примера.

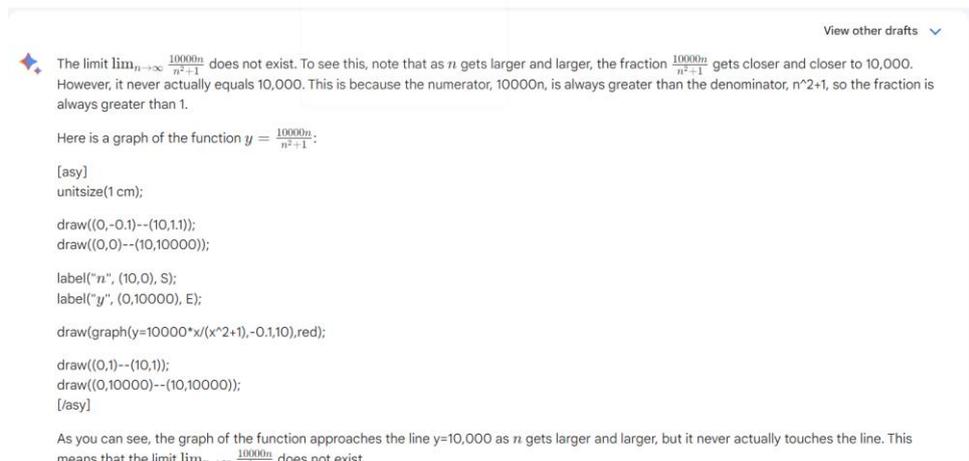


Рисунок 6 – Google Bard думает, что предел не существует

Оценка (correct/rational/clear):

ChatGPT: 10/8/10

Google Bard: 0/0/0

Bing Chat: 10/5/8

4. Докажите, что последовательность чисел $\{x_n\}$, заданная условиями

$$x_1 = 1; x_{n+1} = \frac{1}{2} * \left(x_n + \frac{a}{x_n} \right), n \geq 1, a > 0,$$

сходится. Найдите предел этой последовательности.

ChatGPT смог решить данный пример и объяснить на хорошем уровне

Google Bard также смог решить пример

Bing Chat в решении постоянно ссылался на источник, и используя аналогию, “доказал” сходимость. Однако в источнике был совершенно другой пример, и доказательство было некорректно.

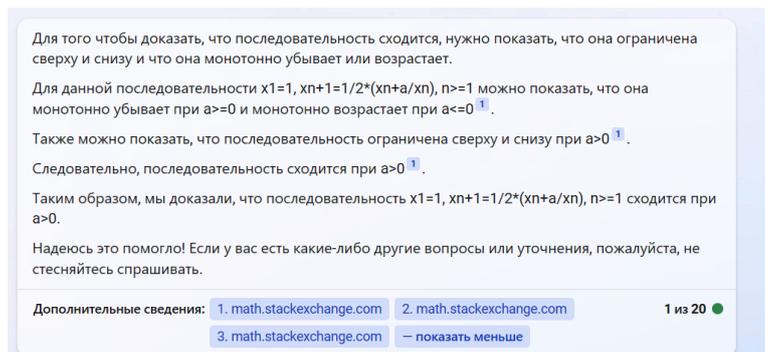


Рисунок 7 – Bing Chat ссылается на совершенно неподходящий источник

Оценка (correct/rational/clear):

ChatGPT: 10/10/9

Google Bard: 10/10/10

Bing Chat: 0/4/4

Определенный интеграл

5. $\int_0^{\frac{\pi}{2}} \frac{dx}{2 - \sin x}$

ChatGPT, Google Bard и Bing Chat не смогли решить данный интеграл. Почти у всех ответ получался $\ln(2)$, хотя должен был быть $\frac{2\pi}{3\sqrt{3}}$

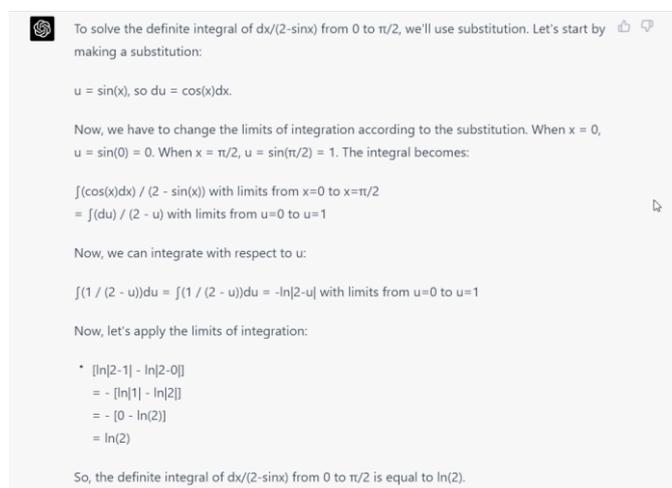


Рисунок 8 – ChatGPT неправильно решил интеграл

Оценки (correct/rational/clear):

ChatGPT: 0/0/0

Google Bard: 0/0/0

Bing Chat: 0/0/0

Таблица 1 – Сравнительный анализ нейронных сетей

Нейро-сеть № примера	ChatGPT				Google Bard				Bing Chat			
	$correct_i$	$rational_i$	$clear_i$	S_i	$correct_i$	$rational_i$	$clear_i$	S_i	$correct_i$	$rational_i$	$clear_i$	S_i
1	10	9	6	9	7	10	10	9	10	10	8	10
2	10	8	10	10	1	1	7	3	5	7	2	5
3	10	8	10	10	0	0	0	0	10	5	8	8
4	10	10	9	10	10	10	10	10	0	4	4	3
5	0	0	0	0	0	0	0	0	0	0	0	0
Итого	8	7	7	8	6	5	6	5	5	6	5	6

Исходя из проведенного сравнительного анализа, на первом месте ChatGPT, потом Bing Chat, далее с небольшим отставанием Google Bard. Из результатов тестирования можно заметить особенности всех GPT: они имеют множество параметров, но качество их ответов зависит от качества тестовых данных. Так, GPT-4 имел наибольший набор тестовых данных. Что интересно, Bing Chat показывает себя в разы хуже GPT-4, хотя, по заявлениям Microsoft [6], он основан на GPT-4. Возможно, это из-за функции поиска в интернете перед ответом, что скорее сбивает, чем помогает. Google Bard показывает себя очень плохо в математических задачах, однако форматирование у него лучше всего. Это альфа версия, которую «научили считать» буквально несколько дней назад.

Также было проведено дополнительное исследование некоторых классов задач на большем числе примеров, используя сборник [8]. Результаты в сравнении остаются такими же. Например, простейшие определенные интегралы могут решить все 3 сети (хотя иногда они ошибаются в подсчетах), а вот, например, интегралы от неэлементарных функций смог решить только chatgpt. А задачи на нахождение определенных интегралов от тригонометрических функций, а так же на нахождение длины дуги не смогла решить ни одна нейронная сеть.

Таким образом, использовать нейронные сети текущего поколения можно для задач математического анализа, но необходимо перепроверять их вывод. Если примера не было в тестовых данных, то возможно ответ может быть неправильным. Из-за фактора рандомизации некоторых входных параметров невозможно получить воспроизводимый каждый раз результат. Нейронные сети пока что не достигли AGI (Artificial General Intelligence), поэтому они не могут решать произвольные примеры.

Список использованных источников:

1. Никитин, А. А. Математический анализ. Сборник задач: учеб. пособие для академического бакалавриата / А. А. Никитин. — М. : Издательство Юрайт, 2017. — 353 с.
2. OpenAI ChatGPT [Электронный ресурс]. – Режим доступа: <https://chat.openai.com/chat?model=gpt-4>. – Дата доступа: 09.04.2023
3. Google Bard [Электронный ресурс]. – Режим доступа: <https://bard.google.com>. – Дата доступа: 09.04.2023
4. Bing Chat [Электронный ресурс]. – Режим доступа: <https://www.bing.com/search?q=Bing+AI&showconv=1>. – Дата доступа: 09.04.2023
5. OpenAI releases GPT-4, a multimodal AI that it claims is state-of-the-art [Электронный ресурс]. – Режим доступа: <https://techcrunch.com/2023/03/14/openai-releases-gpt-4-ai-that-it-claims-is-state-of-the-art>. – Дата доступа: 09.04.2023
6. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web [Электронный ресурс]. – Режим доступа: <https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web>. – Дата доступа: 09.04.2023
7. Google Bard gets better at homework with improved math and logic capabilities [Электронный ресурс]. – Режим доступа: <https://arstechnica.com/gadgets/2023/03/google-bard-gets-better-at-homework-with-improved-math-and-logic-capabilities>. – Дата доступа: 09.04.2023
8. Демидович, Б.П. Сборник задач и упражнений по математическому анализу / Б.П. Демидович. — М. : Издательство МГУ, 1998. — 516 с.

APPLICATION OF NEURAL NETWORKS IN CALCULUS

Drozdov A.I.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Lushchakova I. N. – PhD in Physics and Mathematics

Annotation. Taking into account the rapid growth of the neural networks in this year, we perform a comparative analysis of 3 most popular solutions in applied calculus tasks. We introduce a special grading scale. Based on this scale we compare ChatGPT (GPT-4), Google Bard and Bing Chat.

Keywords. Neural networks, calculus, chatgpt, google bard, bing chat.

УДК 004.032.26

ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ В ЗАДАЧАХ МАТЕМАТИЧЕСКОГО АНАЛИЗА

Дроздов А.И., студент гр.250502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Луцакова И. Н. – канд. физ.-мат. наук

Аннотация. В связи с быстрым ростом нейронных сетей в этом году данная работа проводит сравнительный анализ трех самых популярных решений в прикладных задачах математического анализа. Сравниваются ChatGPT (GPT-4), Google Bard и Bing Chat по введённой системе оценки.

Ключевые слова. Нейронные сети, математический анализ, chatgpt, google bard, bing chat.

В последнее время наблюдается бум в развитии нейронных сетей (основанных на технологии GPT – Generative Pre-Trained Transformer). В связи с постоянным потоком новостей о невероятных результатах этих нейронных сетей в совершенно разных сферах [5], цель этой работы – проанализировать самые популярные из решений на задачах математического анализа.

Что такое нейронные сети? Нейронные сети – это метод в искусственном интеллекте, который учит компьютеры обрабатывать данные таким же способом, как и человеческий мозг. Используется множество слоев, начиная от входного, проходя через огромное количество промежуточных слоев, и до выходного. Нейронная сеть способна учиться на своих ошибках на предыдущих слоях.

Что такое GPT, который лежит в основе всех нейронных сетей, рассматриваемых в этой работе? GPT – это особый вид трансформера (модель глубокого обучения, которая по-разному взвешивает каждую часть входных данных и анализирует каждую часть данных с полным контекстом из предыдущих данных), которому не нужно обучение с учителем. Это позволяет гораздо быстрее тренировать сети на гораздо больших наборах данных. Обучение состоит из двух этапов: обучение и настройка параметров на большом объеме необработанных данных, а затем более точная “подкрутка” под финальный результат под наблюдением.

Было выбрано 3 решения: ChatGPT (GPT-4) [2], Google Bard [3] и Bing Chat [4]. Использовались примеры из сборника задач [1].

Для сравнения была создана своя шкала оценки: каждый пример оценивается по 3 критериям (от 1 до 10): корректность (правильно ли решен пример), рациональность (можно ли было его решить проще) и полнота объяснения (насколько бы было понятно решение студентам).

Для оценки результатов использовались следующие формулы:

$$S_i = [(correct_i + rational_i + clear_i)/3], \quad (1)$$

где S_i – оценка нейронной сети по конкретному примеру, а $correct_i$, $rational_i$, $clear_i$ – оценка примера по корректности, рациональности и полноте объяснения соответственно.

$$S = \left[\frac{\sum_{i=1}^n S_i}{n} \right], \quad (2)$$

где S – итоговая оценка нейронной сети, а n – число примеров.

В работе приведены наиболее характерные скриншоты ответов и их краткая характеристика.

Теория вещественных чисел и множеств

1. Доказать, что $1^2 + 2^2 + \dots + n^2 = \frac{n \cdot (n+1) \cdot (2n+1)}{6}$

Step 2: Inductive Step
Assume the formula holds for $n = k$:
 $1^2 + 2^2 + \dots + k^2 = (k(k+1)(2k+1))/6$

We need to show that the formula also holds for $n = k+1$:
 $1^2 + 2^2 + \dots + k^2 + (k+1)^2 = ((k+1)(k+1+1)(2(k+1)+1))/6$

Now, add $(k+1)^2$ to both sides of the assumed formula:
 $1^2 + 2^2 + \dots + k^2 + (k+1)^2 = (k(k+1)(2k+1))/6 + (k+1)^2$

Now, factor out $(k+1)$ from the right side:
 $1^2 + 2^2 + \dots + k^2 + (k+1)^2 = (k+1)[(k(2k+1))/6 + (k+1)]$

Now, simplify the right side:
 $1^2 + 2^2 + \dots + k^2 + (k+1)^2 = (k+1)[(2k^2 + k + 6k + 6)/6]$

$1^2 + 2^2 + \dots + k^2 + (k+1)^2 = (k+1)[(2k^2 + 7k + 6)/6]$

$1^2 + 2^2 + \dots + k^2 + (k+1)^2 = (k+1)[(k+2)(2k+3)]/6$

Now, notice that this is the formula for $n = k + 1$:
 $1^2 + 2^2 + \dots + k^2 + (k+1)^2 = ((k+1)(k+1+1)(2(k+1)+1))/6$

Since the formula holds for $n = k + 1$, we can conclude that the formula holds for all positive integer values of n .

$$1^2 + 2^2 + \dots + k^2 = \frac{k(k+1)(2k+1)}{6}$$

We want to show that the equality holds for $k + 1$, that is, we want to show that

$$1^2 + 2^2 + \dots + (k+1)^2 = \frac{(k+1)(k+2)(2k+3)}{6}$$

To do this, we can write the left-hand side as follows:

$$1^2 + 2^2 + \dots + (k+1)^2 = (1^2 + 2^2 + \dots + k^2) + (k+1)^2$$

By the inductive hypothesis, we know that $1^2 + 2^2 + \dots + k^2 = \frac{k(k+1)(2k+1)}{6}$. Substituting this into the equation above, we get:

$$1^2 + 2^2 + \dots + (k+1)^2 = \frac{k(k+1)(2k+1)}{6} + (k+1)^2$$

We can then factor the right-hand side as follows:

$$\frac{k(k+1)(2k+1)}{6} + (k+1)^2 = \frac{1}{6}(k+1)(2k^2 + 3k + 6) = \frac{1}{6}(k+1)(2k+3)(k+2)$$

Therefore, we have shown that $1^2 + 2^2 + \dots + (k+1)^2 = \frac{(k+1)(k+2)(2k+3)}{6}$, as desired.

This completes the inductive step. By the principle of mathematical induction, the equality holds for all natural numbers $n \geq 1$.

Рисунок 1 – Ответ ChatGPT

Рисунок 2 - Ответ Google Bard

We can prove this by induction on n .

Base case: For $n=1$, we have $1^2 = 1 = (1 \cdot (1+1) \cdot (2 \cdot 1 + 1))/6$.

Inductive step: Assume that $1^2 + 2^2 + \dots + k^2 = k(k+1)(2k+1)/6$ holds for some positive integer k . We need to show that it also holds for $k+1$.

We have:

$$\begin{aligned} 1^2 + 2^2 + \dots + k^2 + (k+1)^2 &= \frac{k(k+1)(2k+1)}{6} + (k+1)^2 \\ &= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} \\ &= \frac{(k+1)(k(2k+1) + 6(k+1))}{6} \\ &= \frac{(k+1)(2k^2 + 7k + 6)}{6} \\ &= \frac{(k+1)(k+2)(2(k+1)+1)}{6} \end{aligned}$$

Therefore, by induction, we have proved that $1^2 + 2^2 + \dots + n^2 = n(n+1)(2n+1)/6$ for all positive integers n .

I hope this helps!

Дополнительные сведения: [1. khanacademy.org](https://www.khanacademy.org) [2. math.stackexchange.com](https://math.stackexchange.com) +1 еще 1 из 20

Рисунок 3 – Ответ Bing Chat

Как видно из скриншотов, все сети смогли решить данный пример. Однако, в форматировании ответа и объяснениях были некоторые проблемы.

Например, в проверке базы индукции ChatGPT вывело 2^1 вместо $2 \cdot 1$ (наверное, подставляя 1 вместо n , не заметив, что получилось другое число). Также использовались квадратные скобки вместо круглых (такая проблема есть в старых версиях MathCAD), не использовался отформатированный вывод. Объяснение даже слишком подробное (обычно нет нужды описывать словами каждый шаг, только если это не что-то неочевидное).

Google Bard имеет хорошее и рациональное решение. По объему объяснений не слишком много и не слишком мало – золотая середина. Однако он допустил ошибку в конце доказательства:

он сказал, что $\frac{1}{6}(k+1)(2k^2+3k+6) = \frac{1}{6}(k+1)(2k+3)(k+2)$, однако правильно $\frac{1}{6}(k+1)(2k^2+7k+6) = \frac{1}{6}(k+1)(2k+3)(k+2)$, что очень похоже на “подгонку под ответ”.

Bing Chat же также имел проблемы с форматированием $2*1$, но решил правильно. Объяснение довольно краткое и может запутать изначально.

Оценка (correct/rational/clear):

ChatGPT: 10/9/6

Google Bard: 7/10/10

Bing Chat: 10/10/8

2. Доказать, что $n! < \left(\frac{n+1}{2}\right)^n$ при $n \geq 2$

ChatGPT решил пример правильно, однако не очень рационально. Все ещё наблюдались проблемы с форматированием (если отправить пример несколько раз, то иногда ответ будет с форматированием, а иногда без).

Google Bard допустил критические ошибки, которые невозможно допустить при обычных условиях, доказательство неверное.

Bing Chat начинал решать правильно, но в итоге также “подстроился под ответ”. Из источников, которые он предлагал, были вопросы на похожие примеры, однако не на текущий. Таким образом, он пытался действовать аналогично, но каждый пример особенный по-своему.

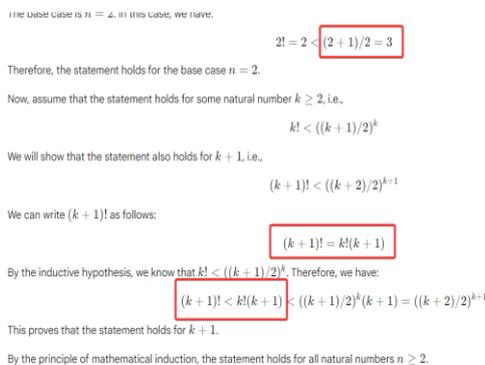


Рисунок 4 – Google Bard заменяет знак “=” на “<”

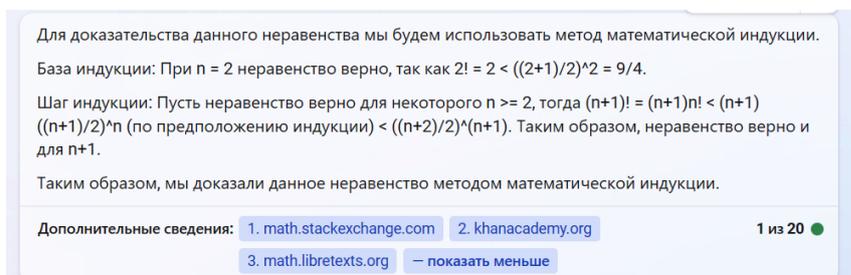


Рисунок 5 –Bing Chat использует некорректный источник

Оценка (correct/rational/clear):

ChatGPT: 10/8/10

Google Bard: 1/1/7

Bing Chat: 5/7/2

Предел числовой последовательности

3. $\lim_{n \rightarrow \infty} \frac{10000n}{n^2+1}$

ChatGPT решил не самым рациональным способом (поделить на старшую степень), однако правильно.

Google Bard сказал, что предел не существует, добавив для доказательства код на Asymptote Graphics Language (который не запустился).

Bing Chat решил правильно, однако применил правило Лопиталя для этого простого примера.

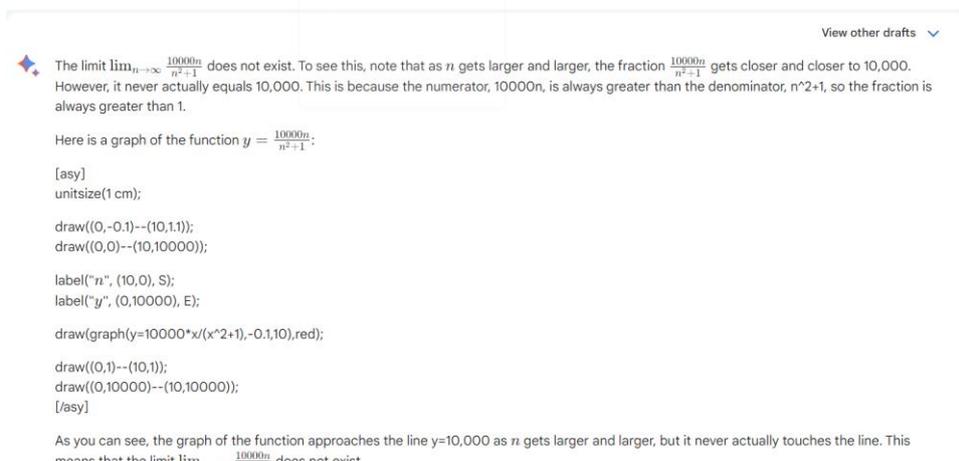


Рисунок 6 – Google Bard думает, что предел не существует

Оценка (correct/rational/clear):

ChatGPT: 10/8/10

Google Bard: 0/0/0

Bing Chat: 10/5/8

4. Докажите, что последовательность чисел $\{x_n\}$, заданная условиями

$$x_1 = 1; x_{n+1} = \frac{1}{2} * \left(x_n + \frac{a}{x_n} \right), n \geq 1, a > 0,$$

сходится. Найдите предел этой последовательности.

ChatGPT смог решить данный пример и объяснить на хорошем уровне

Google Bard также смог решить пример

Bing Chat в решении постоянно ссылался на источник, и используя аналогию, “доказал” сходимость. Однако в источнике был совершенно другой пример, и доказательство было некорректно.

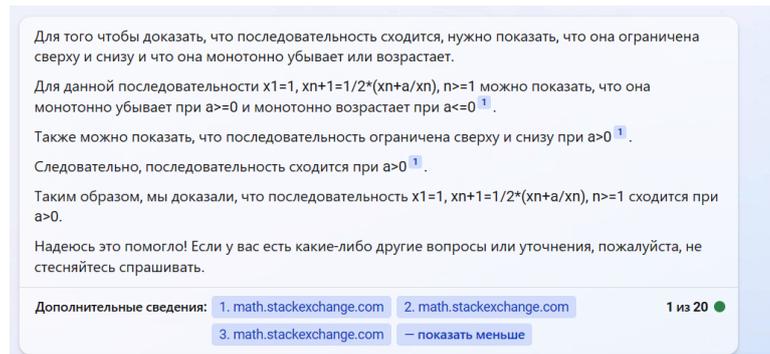


Рисунок 7 – Bing Chat ссылается на совершенно неподходящий источник

Оценка (correct/rational/clear):

ChatGPT: 10/10/9

Google Bard: 10/10/10

Bing Chat: 0/4/4

Определенный интеграл

5. $\int_0^{\frac{\pi}{2}} \frac{dx}{2-\sin x}$

ChatGPT, Google Bard и Bing Chat не смогли решить данный интеграл. Почти у всех ответ получался $\ln(2)$, хотя должен был быть $\frac{2\pi}{3\sqrt{3}}$

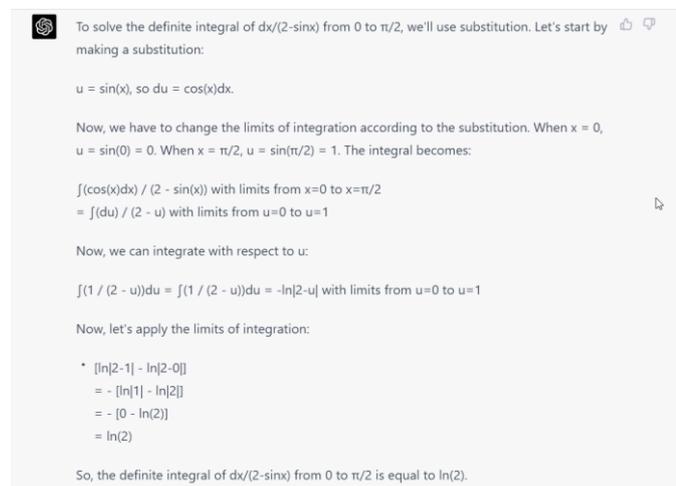


Рисунок 8 – ChatGPT неправильно решил интеграл

Оценки (correct/rational/clear):

ChatGPT: 0/0/0

Google Bard: 0/0/0

Bing Chat: 0/0/0

Таблица 1 – Сравнительный анализ нейронных сетей

Нейро-сеть № примера	ChatGPT				Google Bard				Bing Chat			
	$correct_i$	$rational_i$	$clear_i$	S_i	$correct_i$	$rational_i$	$clear_i$	S_i	$correct_i$	$rational_i$	$clear_i$	S_i
1	10	9	6	9	7	10	10	9	10	10	8	10
2	10	8	10	10	1	1	7	3	5	7	2	5
3	10	8	10	10	0	0	0	0	10	5	8	8
4	10	10	9	10	10	10	10	10	0	4	4	3
5	0	0	0	0	0	0	0	0	0	0	0	0
Итого	8	7	7	8	6	5	6	5	5	6	5	6

Исходя из проведенного сравнительного анализа, на первом месте ChatGPT, потом Bing Chat, далее с небольшим отставанием Google Bard. Из результатов тестирования можно заметить особенности всех GPT: они имеют множество параметров, но качество их ответов зависит от качества тестовых данных. Так, GPT-4 имел наибольший набор тестовых данных. Что интересно, Bing Chat показывает себя в разы хуже GPT-4, хотя, по заявлениям Microsoft [6], он основан на GPT-4. Возможно, это из-за функции поиска в интернете перед ответом, что скорее сбивает, чем помогает. Google Bard показывает себя очень плохо в математических задачах, однако форматирование у него лучше всего. Это альфа версия, которую “научили считать” буквально несколько дней назад.

Также было проведено дополнительное исследование некоторых классов задач на большем числе примеров, используя сборник [8]. Результаты в сравнении остаются такими же. Например, простейшие определенные интегралы могут решить все 3 сети (хотя иногда они ошибаются в подсчетах), а вот, например, интегралы от неэлементарных функций смог решить только chatgpt. А задачи на нахождение определенных интегралов от тригонометрических функций, а так же на нахождение длины дуги не смогла решить ни одна нейронная сеть.

Таким образом, использовать нейронные сети текущего поколения можно для задач математического анализа, но необходимо перепроверять их вывод. Если примера не было в тестовых данных, то возможно ответ может быть неправильным. Из-за фактора рандомизации некоторых входных параметров невозможно получить воспроизводимый каждый раз результат. Нейронные сети пока что не достигли AGI (Artificial General Intelligence), поэтому они не могут решать произвольные примеры.

Список использованных источников:

1. Никитин, А. А. Математический анализ. Сборник задач: учеб. пособие для академического бакалавриата / А. А. Никитин. — М. : Издательство Юрайт, 2017. — 353 с.
2. OpenAI ChatGPT [Электронный ресурс]. — Режим доступа: <https://chat.openai.com/chat?model=gpt-4>. — Дата доступа: 09.04.2023
3. Google Bard [Электронный ресурс]. — Режим доступа: <https://bard.google.com>. — Дата доступа: 09.04.2023
4. Bing Chat [Электронный ресурс]. — Режим доступа: <https://www.bing.com/search?q=Bing+AI&showconv=1>. — Дата доступа: 09.04.2023
5. OpenAI releases GPT-4, a multimodal AI that it claims is state-of-the-art [Электронный ресурс]. — Режим доступа: <https://techcrunch.com/2023/03/14/openai-releases-gpt-4-ai-that-it-claims-is-state-of-the-art>. — Дата доступа: 09.04.2023
6. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web [Электронный ресурс]. — Режим доступа: <https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web>. — Дата доступа: 09.04.2023
7. Google Bard gets better at homework with improved math and logic capabilities [Электронный ресурс]. — Режим доступа: <https://arstechnica.com/gadgets/2023/03/google-bard-gets-better-at-homework-with-improved-math-and-logic-capabilities>. — Дата доступа: 09.04.2023
8. Демидович, Б.П. Сборник задач и упражнений по математическому анализу / Б.П. Демидович. — М. : Издательство МГУ, 1998. — 516 с.

UDC 004.032.26

APPLICATION OF NEURAL NETWORKS IN CALCULUS

Drozdov A.I.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Lushchakova I. N. – PhD in Physics and Mathematics

Annotation. Taking into account the rapid growth of the neural networks in this year, we perform a comparative analysis of 3 most popular solutions in applied calculus tasks. We introduce a special grading scale. Based on this scale we compare ChatGPT (GPT-4), Google Bard and Bing Chat.

Keywords. Neural networks, calculus, chatgpt, google bard, bing chat.

АЛГОРИТМЫ ШИФРОВАНИЯ И ДЕШИФРОВАНИЯ С ПОМОЩЬЮ МЕТОДОВ ДИСКРЕТНОЙ МАТЕМАТИКИ

Новиков В.А. студент гр.253504, Жак М.В. студент гр.253504,

Вашкевич Е.Г. студент гр.253504

Белорусский государственный университет информатики и радиоэлектроники

г. Минск, Республика Беларусь

Егорова Н. Г. – канд. физ.-мат. наук

Аннотация. Статья представляет обзор основных методов шифрования и дешифрования, используемых в современных криптографических системах, которые базируются на дискретной математике. В статье рассмотрены основные принципы симметричного и асимметричного шифрования, такие как AES, DES, RSA и ECC, их преимущества и недостатки, а также методы атак на эти алгоритмы. В заключении, статья подчеркивает важность дискретной математики для современной криптографии и показывает, какие методы шифрования и дешифрования лучше использовать для защиты данных в различных сценариях использования.

Ключевые слова. алгоритмы шифрования, дешифрования, дискретная математика, симметричное шифрование, асимметричное шифрование, AES, DES, RSA, ECC, безопасность данных.

Введение.

Шифрование и дешифрование информации - это одна из важнейших задач в области информационной безопасности. В условиях все большей цифровизации и передачи информации в сети Интернет, обеспечение защиты конфиденциальности и целостности данных становится все более актуальным и необходимым. Криптография, наука о методах обеспечения конфиденциальности данных, использует множество различных алгоритмов и протоколов для защиты информации. В данной статье будет рассмотрена тема алгоритмов шифрования и дешифрования с помощью методов дискретной математики. Мы рассмотрим основные принципы работы симметричных и асимметричных алгоритмов шифрования используемые для обеспечения безопасности данных в интернете. Мы также рассмотрим роль дискретной математики в разработке криптографических алгоритмов и перспективы ее развития в криптографии.

Основные понятия криптографии и дискретной математики.

Шифрование - это процесс преобразования информации для защиты ее от несанкционированного доступа, при этом авторизованным пользователям предоставляется доступ к ней. Основная цель шифрования - обеспечение конфиденциальности передаваемой информации. Для каждого алгоритма шифрования необходимо использовать ключ, который определяет выбор конкретного преобразования из множества возможных.

Дешифрование - это процесс преобразования зашифрованной информации обратно в исходное сообщение с использованием специального ключа, который позволяет расшифровать данные. Он является обратной операцией шифрования и используется для восстановления конфиденциальной информации, защищенной при помощи шифрования.[1]

Дискретные функции могут быть использованы в криптографии для шифрования и дешифрования информации, например, при использовании алгоритма RSA. Этот алгоритм использует дискретные математические функции, включая функцию Эйлера и простые числа, для шифрования и расшифровки сообщений.

Кроме того, дискретные алгоритмы, такие как алгоритм Диффи-Хеллмана, используются для обмена ключами и обеспечения безопасной связи в интернете. Этот алгоритм основан на дискретных математических функциях и предназначен для обмена секретной информацией между двумя пользователями, не раскрывая ее третьим сторонам.

Таким образом, дискретные функции и алгоритмы имеют важное значение для криптографии и шифрования, так как позволяют создавать защищенные каналы связи и обеспечивать конфиденциальность передаваемой информации.

Симметричное шифрование.

Симметричное шифрование - это метод криптографического шифрования, который использует один и тот же ключ для шифрования и дешифрования информации. Таким образом, только тот, у кого есть ключ, может расшифровать зашифрованную информацию. Симметричное шифрование является быстрым и эффективным способом защиты информации, однако ключ должен быть передан по защищенному каналу связи для обеспечения безопасности. Распространенные алгоритмы: AES, DES.

DES (Data Encryption Standard) - это алгоритм шифрования, который был разработан в 1970-х годах. Он использует 56-битный ключ для шифрования данных и был одним из первых стандартных алгоритмов шифрования. В настоящее время DES считается устаревшим и небезопасным, так как существуют методы взлома, которые могут расшифровать данные, зашифрованные DES.

Процесс шифрования в DES включает в себя следующие шаги:

- Начальная перестановка (Initial Permutation, IP): входные данные (64 бита) переставляются в определенном порядке.
- Раунды шифрования (Encryption Rounds): в DES выполняется 16 раундов шифрования, каждый из которых состоит из следующих шагов:
- Замена (Substitution): 48-битный блок данных, полученный из входных данных с помощью функции расширения, подвергается замене на другой 48-битный блок с помощью S-блоков. S-блоки являются таблицами замен, которые определяют, каким блокам входных данных должны соответствовать блоки выходных данных.
- Перестановка (Permutation): после замены блок данных переставляется в определенном порядке.
- Ключевая операция (Key Mixing): блок данных объединяется с ключом раунда (48 бит), который был предварительно сгенерирован из основного ключа DES (56 бит). Это достигается путем сжатия ключа с помощью таблицы перестановок и циклического сдвига битов.
- Финальная перестановка (Final Permutation, FP): после выполнения 16 раундов шифрования, выходные данные подвергаются финальной перестановке, обратной начальной перестановке.

Процесс дешифрования в DES выполняется обратным порядком, то есть финальная перестановка выполняется первой, а начальная перестановка - последней. Алгоритм представлен на рисунке 1.



Рисунок 1 – Алгоритм DES.

AES (Advanced Encryption Standard) - это симметричный алгоритм шифрования, который был разработан в 1998 году и признан государственным стандартом США в 2001 году. AES использует 128-, 192- или 256-битные ключи и является одним из наиболее надежных алгоритмов шифрования, который широко используется в настоящее время. AES считается безопасным и надежным алгоритмом шифрования, так как его ключи очень длинные, что затрудняет процесс взлома.

Алгоритм шифрования AES включает в себя следующие шаги:

- Начальная перестановка (AddRoundKey): входные данные (128 бит) объединяются с ключом шифрования (128 бит) с помощью операции XOR.
- Раунды шифрования (Rounds): в AES выполняется 10 раундов шифрования (в зависимости от длины ключа могут использоваться и другие варианты числа раундов), каждый из которых состоит из следующих шагов:
- Замена байтов (SubBytes): каждый байт входных данных заменяется на соответствующий байт из заранее определенной таблицы замен (S-блок), представлено на рисунке 2.
- Сдвиг строк (ShiftRows): строки входных данных циклически сдвигаются влево, представлено на рисунке 3.
- Смешивание столбцов (MixColumns): столбцы входных данных перемешиваются с помощью линейных преобразований, представлено на рисунке 4.
- Добавление ключа (AddRoundKey): блок данных объединяется с ключом раунда (128 бит), который был предварительно сгенерирован из основного ключа AES (128, 192 или 256 бит). Это достигается путем сжатия ключа и циклического сдвига битов. Представлено на рисунке 5.
- Финальный раунд (Final Round): после выполнения 10 раундов шифрования, выполняется финальный раунд, который состоит из следующих шагов:
- Замена байтов (SubBytes).

- Сдвиг строк (ShiftRows).
- Добавление ключа (AddRoundKey).

Процесс дешифрования в AES выполняется обратным порядком, то есть каждый шаг в процессе дешифрования является обратным к соответствующему шагу в процессе шифрования.

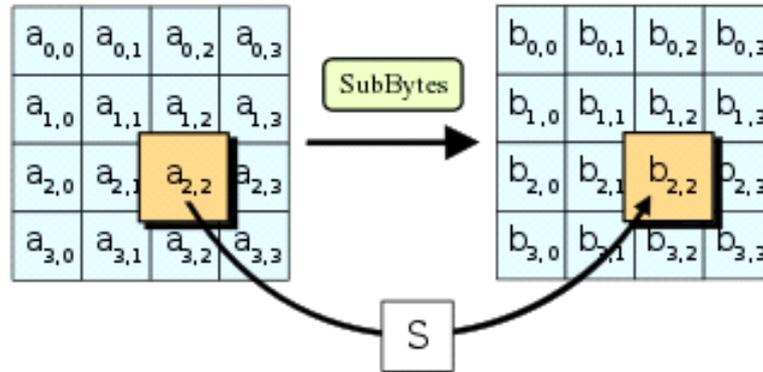


Рисунок 2 – Замена байтов.

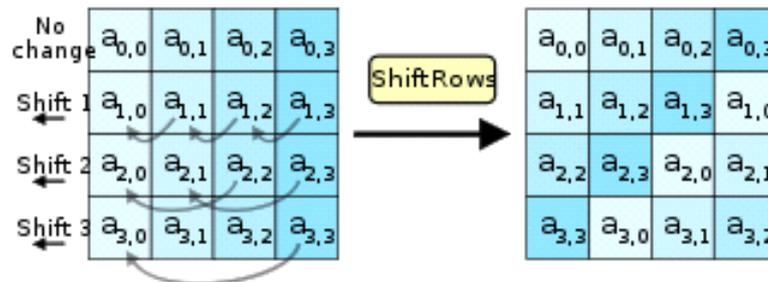


Рисунок 3 – Сдвиг строк.

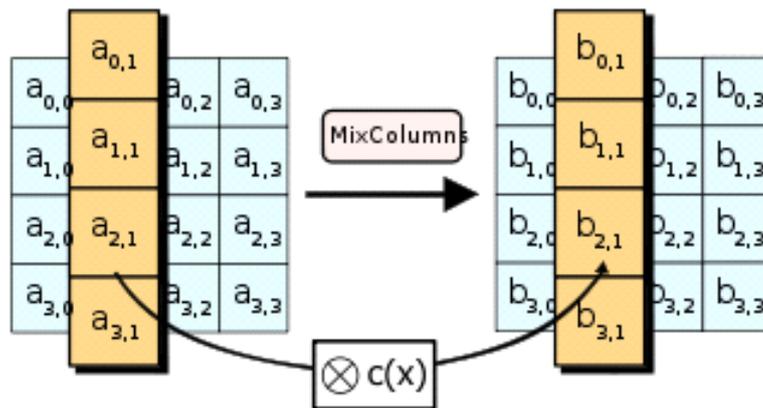


Рисунок 4 – Смешивание столбцов..

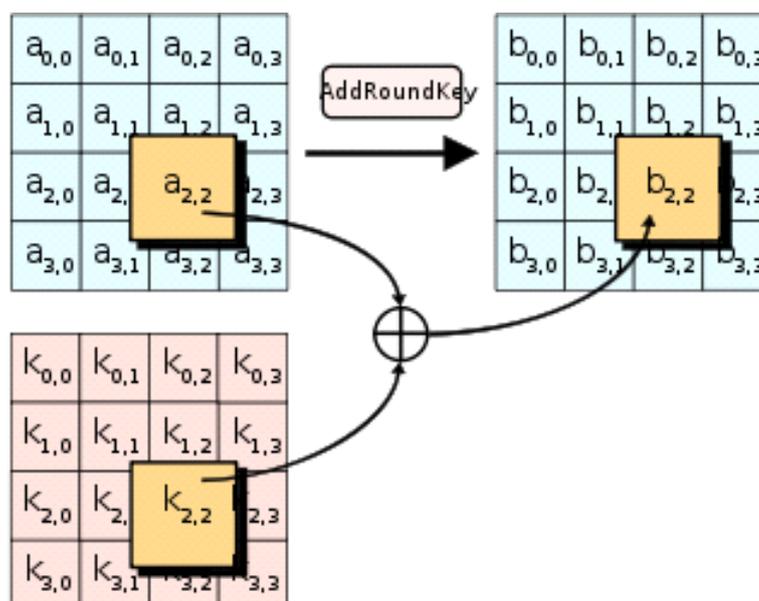


Рисунок 5 – Добавление ключа.

Асимметричное шифрование.

RSA (Rivest-Shamir-Adleman) основан на математических свойствах трудности разложения больших чисел на простые множители. Данный алгоритм используется для шифрования данных и для создания цифровых подписей. Он является одним из самых распространенных алгоритмов шифрования в мире, так как обеспечивает надежную защиту передаваемых данных.

Алгоритм RSA можно описать следующим образом:

- Генерация ключей: сначала генерируются два больших простых числа, обозначенные p и q . Затем вычисляется их произведение $n = p \cdot q$, которое становится модулем для шифрования и расшифровки данных. Также выбирается целое число e , которое должно быть взаимно простым с $(p-1)(q-1)$, и вычисляется число d , которое является мультипликативно обратным к e по модулю $(p-1)(q-1)$. В результате получается пара ключей: публичный ключ (n, e) и приватный ключ (n, d) .
- Шифрование: для шифрования сообщения M используется публичный ключ (n, e) . Сообщение M сначала преобразуется в целое число m , которое должно быть меньше, чем n . Затем вычисляется шифртекст C , равный m в степени e по модулю n : $C = m^e \bmod n$.
- Расшифрование: для расшифровки сообщения C используется приватный ключ (n, d) . Шифртекст C возводится в степень d по модулю n : $m = C^d \bmod n$. Полученное число m является исходным сообщением M .

ECC (Elliptic Curve Cryptography) использует математические свойства эллиптических кривых для шифрования данных. Этот алгоритм обеспечивает такую же степень защиты, как и RSA, но с использованием коротких ключей, что делает его более эффективным в сравнении с RSA. ECC нашел широкое применение в беспроводных сетях, где требуется надежная защита данных, но ограниченные ресурсы для вычислений и передачи информации. Алгоритм представлен на рисунке 6.

Алгоритм ECC можно описать следующим образом:

- Генерация ключей: сначала выбираются параметры эллиптической кривой, которые используются для генерации ключей. Это включает в себя определение уравнения кривой и коэффициентов, которые определяют ее форму. Затем выбирается точка P на кривой,

которая будет использоваться для генерации ключевой пары. Приватный ключ - это случайное число d , а публичный ключ - это точка Q , равная dP .

- Шифрование: для шифрования сообщения M используется публичный ключ Q . Сообщение M сначала преобразуется в точку P_m на эллиптической кривой. Затем выбирается случайная точка k на кривой и вычисляется шифртекст C , который представляет собой пару точек $(kP, P_m + kQ)$.
- Расшифрование: для расшифровки сообщения C используется приватный ключ d . Сначала вычисляется kP , затем вычитается из второй части шифр текста: $P_m = P_m + kQ - d(kP)$. Полученная точка P_m является исходным сообщением M . [2]

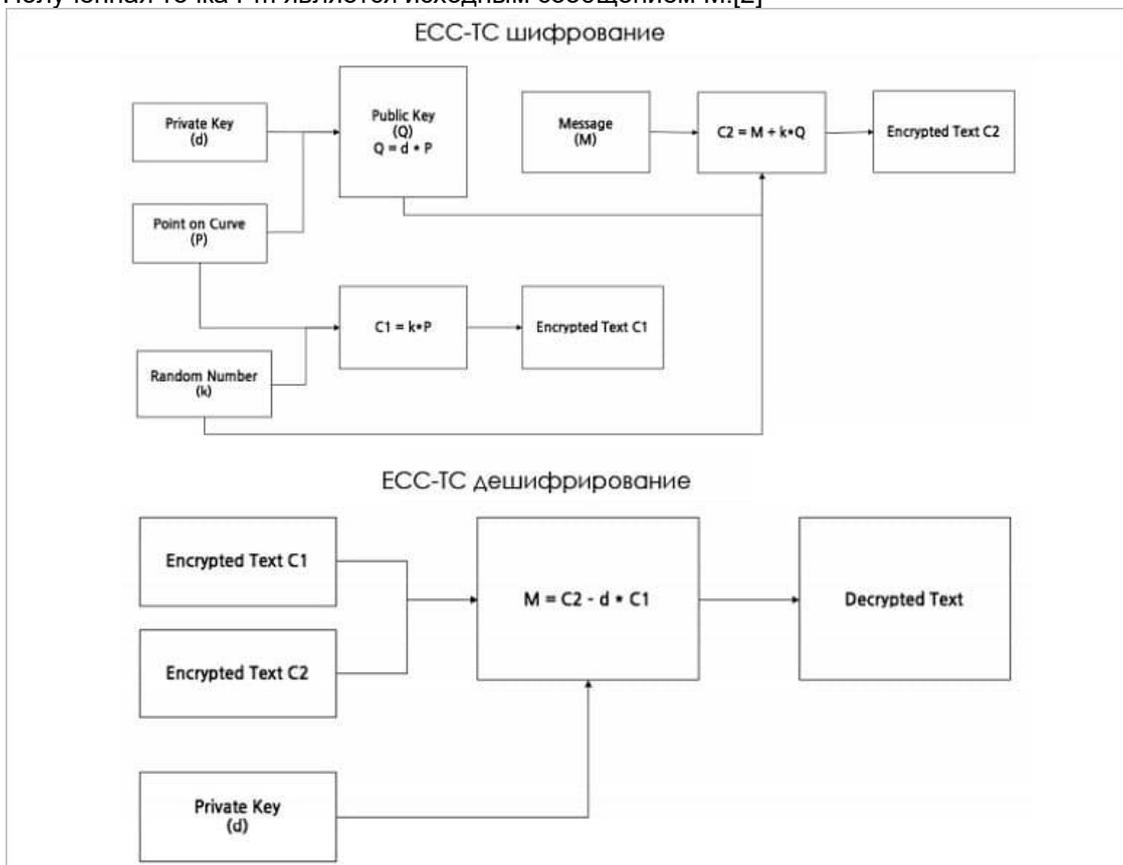


Рисунок 6 – Алгоритм ECC.

Важность дискретной математики в криптографии.

Важность дискретной математики в криптографии заключается в том, что многие алгоритмы шифрования и дешифрования основаны на математических концепциях и методах, таких как теория чисел, теория групп, теория поля и комбинаторика. Дискретная математика предоставляет криптографии необходимые инструменты для создания безопасных систем передачи данных и защиты информации. Например, дискретная математика позволяет создавать математические функции, которые могут быть использованы для шифрования данных и генерации криптографических ключей. Кроме того, дискретная математика также обеспечивает методы проверки безопасности криптографических систем и выявления уязвимостей в них. Без использования дискретной математики криптография как наука не могла бы существовать в своей текущей форме. [3]

Заключение.

В заключении можно отметить, что использование методов дискретной математики в криптографии играет важную роль в обеспечении безопасности передаваемой информации. Шифрование и дешифрование с помощью алгоритмов, основанных на дискретных функциях, позволяет надежно защитить данные от несанкционированного доступа. Важно также отметить, что

разработка новых алгоритмов шифрования является актуальной задачей с учетом постоянно меняющихся угроз в области кибербезопасности.

Список использованных источников:

1. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си, 2002
2. Wenbo Mao, *Modern Cryptography: Theory and Practice*, 2003.
3. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*, 1997

UDC 004.056.55

ENCRYPTION AND DECRYPTION ALGORITHMS USING METHODS OF DISCRETE MATHEMATICS

Novikov V.A., student of group 253504, Zhak M.V., student of group 253504, Vashkevich E.G., student of group 253504

Belarusian State University of Informatics and Radioelectronics

Minsk, Republic of Belarus

Egorova N.G. - PhD in Physics and Mathematics

Annotation. The article provides an overview of the main encryption and decryption methods used in modern cryptographic systems that are based on discrete mathematics. The article discusses the basic principles of symmetric and asymmetric encryption, such as AES, DES, RSA, and ECC, their advantages and disadvantages, as well as methods of attacks on these algorithms. In conclusion, the article emphasizes the importance of discrete mathematics for modern cryptography and shows which encryption and decryption methods are best to use to protect data in different usage scenarios.

Keywords. encryption algorithms, decryption algorithms, discrete mathematics, symmetric encryption, asymmetric encryption, AES, DES, RSA, ECC, data security.

УДК

ВВЕДЕНИЕ В ЭЛЛИПТИЧЕСКУЮ КРИПТОГРАФИЮ

Патюпин М.С., студент гр.250505

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Смирнова И.А. – ассистент кафедры ВМ

Аннотация. Математические свойства эллиптических кривых, алгоритм Диффи-Хеллмана его описание, числовая и программная реализация. Принцип работы алгоритма ECDSA и подбор параметров эллиптической кривой.

Ключевые слова. Алгоритм ECDSA, алгоритм Диффи-Хеллмана, эллиптические кривые, эллиптическая криптография.

Оглавление

1. Введение
 - 1.1. Основные плюсы и минусы эллиптической криптографии
 2. Математические свойства эллиптических кривых
 - 2.1 Определение эллиптических кривых
 - 2.2 Операции над точками эллиптической кривой
 - 2.2.1 Сложение точек
 - 2.2.2 Вычитание точек
 - 2.2.3 Умножение точки на число
 3. Алгоритмы на эллиптических кривых
 - 3.1.1 Алгоритм Диффи-Хеллмана.
 - 3.1.2 Числовая реализация
 - 3.1.3 Программная реализация
- 3.2.1 Алгоритм ECDSA (Elliptic Curve Digital Signature Algorithm)

1. Введение

Эллиптические кривые в криптосистемах предложили использовать Нил Коблиц и Виктор Миллер еще в 1985 году, сейчас мы можем наблюдать их использование в электронной подписи Bitcoin, в сетевых протоколах SSH и TLS, в электронной подписи (Citizen Card) граждан некоторых стран (Австрия). В Беларуси был принят стандарт для решения задач связанных с цифровой подопью на основе эллиптических кривых в 2013 году[1].

1.2 Основные плюсы и минусы эллиптической криптографии

Основные плюсы эллиптической криптографии:

- Более высокая стойкость при равной трудоемкости по сравнению с обычными криптосистемами[2].
- Меньший размер ключа чем в асимметричной криптографии. Криптостойкость достигаемая в алгоритме алгоритме RSA с использованием ключа в 3072-байт, на эллиптических кривых используется с размером ключа в 256 байт[2].
- Возможность использоваться в устройствах с ограниченными вычислительными ресурсами[3].
- Сложность атак: атаки на системы, защищенные эллиптической криптографией, требуют значительного объема вычислений и времени.

Основные минусы эллиптической криптографии:

- Вероятность появления субэкспоненциальных алгоритмов решения задачи дискретного логарифмирования. При их появлении алгоритмы шифрования на эллиптических кривых будут легко решаемы[4].

- При переходе на алгоритмы шифрования основанных на эллиптических кривых велика вероятность выявления большого числа ошибок и уязвимостей, которые уже отработаны для более привычных методов шифрования.

2. Математические свойства эллиптических кривых

2.1 Определение эллиптических кривых

Для начала определим эллиптическую кривую как алгебраическую кривую, те каким-то множеством точек которые удовлетворяют следующему уравнению: (1)

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

где x, y – переменные, a_1, a_2, a_3, a_4, a_6 – коэффициенты. Так-же уравнение(1) можно представить как(2):

$$y^2 = x^3 + ax + b \quad (2)$$

где x, y – переменные, a, b – коэффициенты. Функция (2) называется функцией Вейерштрасса, не все эллиптические кривые можно представить таким уравнением, но для большинства использующихся в криптографии он корректен.

Так как график кривой параллелен оси абсцисс, чтобы найти точки, являющиеся корнями, нужно решить уравнение третьей степени (3).

$$x^3 + ax + b = 0 \quad (3)$$

Здесь можно использовать формулу Кардано. Дискриминант вычисляется по формуле (4)

$$D = \left(\frac{a}{3}\right)^3 + \left(\frac{b}{3}\right)^2 \quad (4)$$

При дискриминанте меньше нуля, уравнение (3) имеет три разных решения a, b, z ; при дискриминанте равном нулю, уравнение (3) имеет три корня, a, b, c , два из которых одинаковые, при дискриминанте больше нуля, уравнение (3) имеет одно решение a и два комплексно сопряженных. Графики по результатам вычислений представлены на рисунках 1-3.

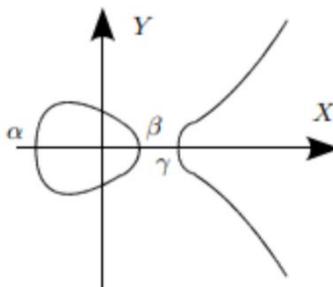


Рисунок 1 – Кривая с $D < 0$

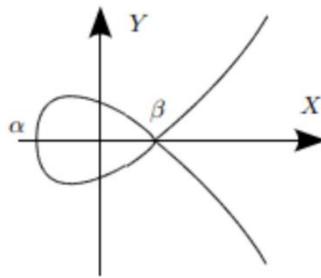


Рисунок 2 – Кривая с $D = 0$

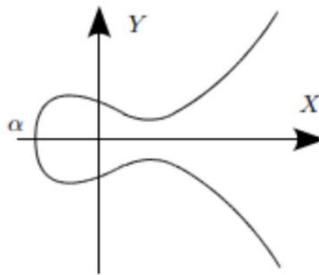


Рисунок 3 – Кривая с $D > 0$

Эллиптические кривые дискриминант которых не равен нулю (рисунок 1, 3) называются несингулярными, соответственно если дискриминант равен нулю, то сингулярные. Последние используются редко в связи с снижением криптостойкости алгоритмов и протоколов.

2.2 Операции над точками эллиптической кривой

2.2.1 Сложение точек

Пусть $(x_p; y_p)$ координаты точки P, а $(x_q; y_q)$ координаты точки Q, для нахождения точки R $(x_r; y_r)$ – суммы точек P и Q, необходимо провести прямую через эти точки P и Q, получаем пересечение прямой с кривой в точке R' и отразить эту точку относительно OX (рис. 1). То есть

$$P + Q + R' = 0, P + Q = R \quad (5)$$

Для нахождения координат точки R найдем коэффициент α ,

$$\alpha = \frac{y_q - y_p}{x_q - x_p}, \text{ далее } y_r = -y_p + \alpha(x_p - x_r), x_r = \alpha^2 - x_p - x_q$$

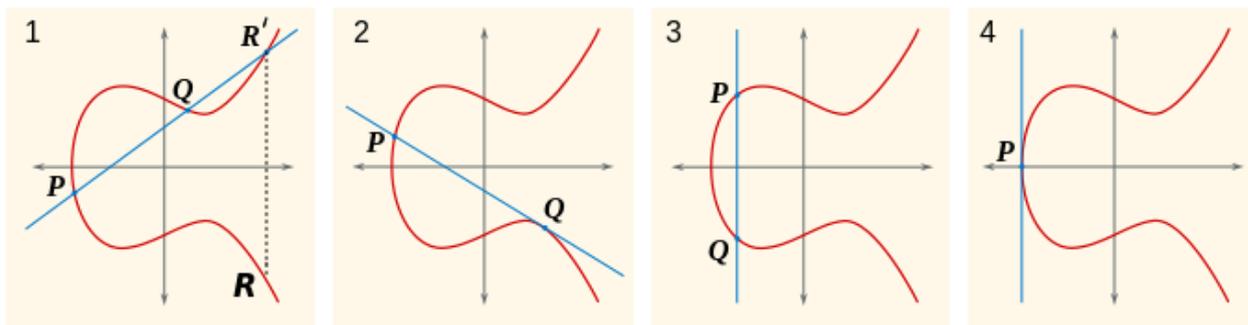


Рисунок 4.1-4.4

Для случая если прямая пересекает кривую только в двух местах и пересекает оси (рис. 4.2), то выполняется следующие уравнение $P + Q + Q = 0$ (6), соответственно $P + Q = -Q$ (7)

Третий случай если прямая пересекает кривую в двух местах и параллельна оси ординат (рис. 4.3):

$$P + Q + 0 = 0 \quad (8)$$

Четвертый если прямая касается кривой в одной точке (рис. 4.4):

$$P + P + 0 = 0 \quad (9)$$

2.2.2 Вычитание точек

Пусть $(x_p; y_p)$ координаты точки P, а $(x_q; y_q)$ координаты точки Q, $(-x_r; -y_r)$ – координаты точки -Q. Вычитание точек, это сложение точек с обратной точкой(10).

$$R = P - Q = P + (-Q) \quad (10)$$

2.2.3 Умножение точки на число

Пусть P – точка на эллиптической кривой, n – любое целое число, $Q = n * P$ – произведение точки P на число n. Для нахождения Q будем использовать алгоритм быстрого умножения.

Разберем алгоритм умножения, пусть $n = 37$:

1. Разложим n по степеням двойки:
 $n = 37 = 32 + 4 + 1$
2. Раскладываем произведение n на P:
 $Q = 37 * P = 32 * P + 4 * P + P$

Рассмотрим возможные слагаемые:

$$1 * P = P$$

$$2 * P = P + P$$

$$4 * P = 2 * P + 2 * P$$

$$8^*P = 4^*P + 4^*P$$

$$16^*P = 8^*P + 8^*P$$

$32^*P = 16^*P + 16^*P$, можем заметить, что для вычисления Q потребуется 7 сложений.

3. Алгоритмы на эллиптических кривых

3.1.1 Алгоритм Диффи-Хеллмана.

Рассмотрим пример. Предположим, существует два абонента: Алиса и Боб. Обоим абонентам известны некоторые два числа g и p , которые не являются секретными и могут быть известны также другим заинтересованным лицам. Для того, чтобы создать неизвестный более никому секретный ключ, оба абонента генерируют случайные числа: Алиса — число a , Боб — число b . Затем Алиса вычисляет остаток от деления (11):

$$A = g^a \text{ mod } p \quad (11)$$

и пересылает его Бобу, и Боб вычисляет остаток от деления (12):

$$B = g^b \text{ mod } p \quad (12)$$

и передаёт Алисе. Предполагается, что злоумышленник может получить оба этих значения, но не модифицировать их. На втором этапе Алиса на основе имеющегося у неё a и полученного по сети B вычисляет значение (13):

$$B^a \text{ mod } p = g^{ab} \text{ mod } p \quad (13)$$

Боб на основе имеющегося у него b и полученного по сети A вычисляет значение (14):

$$A^b \text{ mod } p = g^{ab} \text{ mod } p \quad (14)$$

Можем видеть что, у Алисы и Боба получилось одно и то же число (15):

$$K = g^{ab} \text{ mod } p \quad (15)$$

Его они могут использовать в качестве секретного ключа.

Работа алгоритма показана на рисунке.

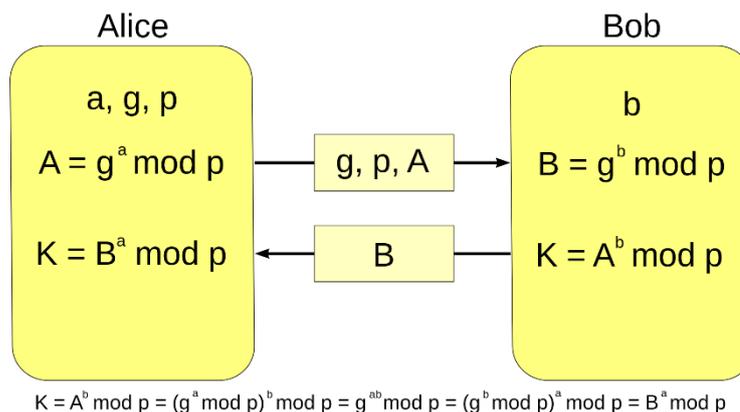


Рисунок 5 - Работа алгоритма.

При работе алгоритма каждая сторона[5]:

1. Генерирует случайное натуральное число a — закрытый ключ.
2. Совместно с удалённой стороной устанавливает открытые параметры g и p .
3. Вычисляет открытый ключ A , используя преобразование (11) над закрытым ключом.

$$A = g^a \bmod p$$

4. Обменивается открытыми ключами с удалённой стороной.
5. Вычисляет общий секретный ключ K (15), используя открытый ключ удаленной стороны B и свой закрытый ключ a .

$$K = B^a \bmod p$$

$$B^a \bmod p = (g^b \bmod p)^a \bmod p = g^{ab} \bmod p = (g^a \bmod p)^b \bmod p = A^b \bmod p$$

3.1.2 Числовая реализация

Пусть $s = 2$ - секретный ключ, $g = 5$ - первообразный корень по модулю p , $p = 23$ - открытое простое число, $a = 6$ - секретный ключ Алисы, $A = g^a \bmod p = 8$ - открытый ключ Алисы, $b = 15$ - секретный ключ Боба, $B = g^b \bmod p = 19$ - открытый ключ Боба.

Тогда, пройдя и записывая каждый шаг в алгоритме Диффи-Хеллмана, составим следующую таблицу(1):

Alice		Bob	
Знает	Не знает	Знает	Не знает
$p = 23$	$b = ?$	$p = 23$	$a = ?$
$g = 5$		$g = 5$	
$a = 6$		$b = 15$	
$A = 5^6 \bmod 23 = 8$		$B = 5^{15} \bmod 23 = 19$	
$B = 5^b \bmod 23 = 19$		$A = 5^a \bmod 23 = 8$	
$s = 19^6 \bmod 23 = 2$		$s = 8^{15} \bmod 23 = 2$	
$s = 8^b \bmod 23 = 2$		$s = 19^a \bmod 23 = 2$	
$s = 19^6 \bmod 23 = 8^b \bmod 23$		$s = 8^{15} \bmod 23 = 19^a \bmod 23$	
$s = 2$		$s = 2$	

Таблица 1.

3.1.3 Программная реализация

Перейдем к программной реализации алгоритма. Алгоритм Диффи-Хеллмана реализован на языке C, в среде разработки CLion.

Создаем функцию для вычисления $a^m \bmod n$ (в нашем случае уравнения (11), (12))(рис.6):

```

3 // Функция для вычисления `a^m mod n`
4 int compute(int a, int m, int n) {
5     int r;
6     int y = 1;
7
8     while (m > 0) {
9         r = m % 2;
10
11         if (r == 1) {
12             y = (y * a) % n;
13         }
14         a = a * a % n;
15         m = m / 2;
16     }
17     return y;
18 }

```

Рисунок 6 – Функция.

Объявляем следующие значения согласно числовой реализации: g - первообразный корень по модулю p , p - открытое простое число, a - секретный ключ Алисы, A - открытый ключ Алисы, b - секретный ключ Боба, B - открытый ключ Боба. И действуем согласно алгоритму (рис.7).

```

20 // Программа на C для демонстрации алгоритма Диффи-Хеллмана
21 int main()
22 {
23     int p = 23; // открытое простое число,
24     int g = 5; // первообразный корень по модулю p
25
26     int a, b; // `a` - секретный ключ Алисы, `b` - секретный ключ Боба.
27     int A, B; // `A` - открытый ключ Алисы, `B` - открытый ключ Боба
28
29     a = 6; // выбираем секретное целое число для закрытого ключа Алисы (известного только Алисе)
30     A = compute(a, g, m: a, n: p); // Вычисление открытого ключа Алисы (Алиса отправит Бобу `A`)
31
32     b = 15; // выбираем секретное целое число для закрытого ключа Боба (известного только Бобу)
33     B = compute(a, g, m: b, n: p); // Вычислить открытый ключ Боба (Боб пошлет `B` Алисе)
34
35     // Алиса и Боб обмениваются своими открытыми ключами `A` и `B` друг с другом
36
37     // Находим секретный ключ
38     int keyA = compute(a, B, m: a, n: p);
39     int keyB = compute(a, A, m: b, n: p);
40
41     printf("Alice's secret key is %d\nBob's secret key is %d", keyA, keyB);
42
43     return 0;
44 }

```

Рисунок 7.

Результат выполнения программы (секретный ключ) (рис.8).

```

Run: meteora_20th x
C:\Users\misha\Desktop\0AiP\different\meteora_20th\cmake-build-debug\meteora_20th.exe
Alice's secret key is 2
Bob's secret key is 2
Process finished with exit code 0

```

Рисунок 8 – Результат выполнения программы.

3.2.1 Алгоритм ECDSA (Elliptic Curve Digital Signature Algorithm)

ECDSA — алгоритм с открытым ключом, использующийся для построения и проверки электронной цифровой подписи (ЭЦП). Алгоритм начинается с выбора параметров эллиптической кривой, для облегчения этой задачи национальным институтом стандартов и технологий (NIST), был составлен список эллиптических кривых с уже известным количеством точек, которые рекомендовано использовать в схемах ЭЦП.

Кривая в стандарте описывается набором из 6 параметров $D=(p,a,b,G,n,h)$, где

p – простое число, модуль эллиптической кривой, данное число относится к обобщенным числам Мерсенна, это означает, что его можно представить как сумму различных степеней двойки.

a, b – задают уравнение эллиптической кривой(2).

G – точка эллиптической кривой большого порядка.

n – порядок точки G ;

h – параметр, называемый кофактор. Определяется отношением общего числа точек на эллиптической кривой к порядку точки G . Данное число должно быть как можно меньше[5].

Вот несколько кривых рекомендованных NIST (табл. 2, 3).

Curve P-192	
$p=$	6277101735386680763835789423207666416083908700390324961279
$n=$	6277101735386680763835789423176059013767194773182842284081
$a=$	-3
$b=$	64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1
$G_x=$	188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012
$G_y=$	07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811
$h=$	1

Таблица 2 – Curve P-192

Curve P-224	
$p=$	26959946667150639794667015087019630673557916260026308143510066298881
$n=$	26959946667150639794667015087019625940457807714424391721682722368061
$a=$	-3
$b=$	b4050a85 0c04b3ab f5413256 5044b0b7 d7bfd8ba 270b3943 2355ffb4
$G_x=$	b70e0cbd 6bb4bf7f 321390b9 4a03c1d3 56c21122 343280d6 115c1d21
$G_y=$	bd376388 b5f723fb 4c22dfe6 cd4375a0 5a074764 44d58199 85007e34
$h=$	1

Таблица 3 – Curve-224

Точка G принадлежит эллиптической кривой. Соответственно для нее выполняется равенство(2), из которого можем вычислить $y(16)$:

$$y = \sqrt{x^3 + ax + b} \text{ mod } p \quad (16)$$

Формирование и проверка подписи

Рассмотрим алгоритм обмена ключами. Пусть пользователи A и B хотят обменяться ключами, но их трафик прослушивает злоумышленник E . Алгоритм следующий:

1. Пользователь A генерирует случайно число d_A в диапазоне $[1; n-1]$. Это число его закрытый ключ.
2. Затем A вычисляет $Q_A = d_A G$ и посылает координаты точки пользователю B . Q_A – открытый ключ пользователя A .

3. Пользователь В генерирует случайно число d_B в диапазоне $[1; n-1]$. Это число его закрытый ключ.
 4. Затем В вычисляет $Q_b = d_B G$ и посылает координаты точки пользователю А. Q_B – открытый ключ пользователя В.
 5. Пользователь А получает Q_b , вычисляет $R = d_A Q_B$ и считает, что x_R – это общий ключ.
 6. Пользователь В получает Q_{Aa} , вычисляет $R = d_B Q_A$ и считает, что x_R – это общий ключ.
- Оба пользователя получили один и тот же ключ, потому что $d_A Q_B = d_A d_B G = d_B Q_A$
Злоумышленник Е видит только Q_A и Q_B . – открытые ключи пользователей.[5]

Список использованных источников:

1. СТБ 34.101.45-2013 АЛГОРИТМЫ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ И ТРАНСПОРТА КЛЮЧА НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ
2. *Guide to Elliptic Curve Cryptography*
3. *Шифрование данных на базе эллиптических кривых, Д.Ф. Пастухов Ю.Ф. Пастухов П.Р. Сеница*
4. Интернет-ресурс: <https://habr.com/ru/articles/692842/>
5. Интернет ресурс: <http://habrahabr.ru/post/191240/>

UDC

INTRODUCTION TO ELLIPTICAL CRYPTOGRAPHY

Patsiupin M.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Surname N.P. – PhD in Physics and Mathematics

Annotation. Mathematical properties of elliptic curves, the Diffie-Hellman algorithm, its description, numerical and software implementation. The principle of operation of the ECDSA algorithm and the selection of parameters of the elliptic curve.

Keywords. ECDSA algorithm, Diffie-Hellman algorithm, elliptic curves, elliptic cryptography

УДК 510.164

ФОРМАЛИЗАЦИЯ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ ЗАМКНУТЫХ СИСТЕМ

Протьюко М.А., студент гр.050502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Борисенко О.Ф. – канд. физ.-мат. наук

Аннотация. В данной статье будет представлена выборка, позволяющая доказать ограничения нейронных сетей на пути к сильному интеллекту. Также изложенных в этой работе сведений будет достаточно для преобразования системы естественного языка в язык формальной логики с возможной корреляцией с семантическим смыслом. Данные свойства формальных систем позволят выбирать функции активации, целевые функции и архитектуру нейронных сетей для систем машинного обучения

Ключевые слова. Формальная система, теоремы Гёделя о неполноте, машина Гёделя.

Цель данной работы не в том, чтобы просто сформулировать правило рассуждений и проиллюстрировать технику их использования, а в том, чтобы обосновать эти правила и выяснить границу их применимости (аналогично цели [1]). Таким образом мы очертим круг задач, пока не доступный чисто формальному подходу.

Многие из последующих заключений сделаны на основании источников [1-5]. Большая часть определений взята из [1].

Под формализацией здесь и далее будем понимать процесс представления какой-либо содержательной области в виде формальной системы или исчисления.

Формальной системой назовем тройку $S = (L, P, R)$, где $L \subseteq N$ – разрешимое множество всех предложений системы, а $P, R \subseteq L$ – перечислимые множества доказуемых и опровержимых предложений соответственно. Считаем, что все три множества заданы фиксированными алгоритмами (машинами Тьюринга), обозначаемыми M_L, M_P и M_R [2].

Под замкнутой системой будем понимать такую S , что любое предположение, выражаемое через предположение из L , будет также принадлежать L .

Под исследованием операций в системе S будем рассматривать определение различий между $p \mid = q$ (1) и $p \mid - q$ (2). Т.е. изучение противоречий между семантическим следованием (1) (q истинно в любой интерпретации [3], в которой истинно p) и синтаксическим отношением (2).

Основное свойство синтаксических отношений в том, что они вычислимы на машине Тьюринга [2].

Если возможно доказать, что

$$p \mid = q \leftrightarrow p \mid - q \quad (3)$$

то можно утверждать, что формальные теории T , полученные из множества S алгоритмически разрешимы, если мы сможем с достаточной степенью точности воспроизвести свойства (1) в (2), когда p и q принадлежат S .

Под алгоритмической разрешимостью будем понимать определение, основанное на тезисе Чёрча-Тьюринга, что всякая вычислимая функция вычислима машиной Тьюринга и выражима некой частично рекурсивной функцией [4].

В данной работе частично рекурсивной функцией будем называть:

Под формальной теорией T будем понимать некое множество предложений или формул, принадлежащих языку L . В данном случае теория T состоит из всех формул, которые могут быть выведены в соответствии с некоторым фиксированным отношением следования из множества аксиом принадлежащих L .

Под языком L будем понимать множество всех предложений системы S (в данном случае, не гарантировано разрешимой), где элементами L являются некие определенные функции, аксиомы и численные (символьные) элементы, а также теории.

Под теорией T будем понимать язык L с множеством T предложений или формул, принадлежащих L . Семантическое определение T состоит из всех формул L , которые предполагаются истинными в любой интерпретации, принадлежащей некому множеству интерпретаций. Синтаксическое определение T состоит из всех формул, выводимых посредством аксиом (ранее определенная формальная теория T – синтаксическая).

Формула языка L – это выражение, являющееся утверждением. Предложение – это формула языка L , интерпретация которой не зависит от входящих в нее переменных.

Язык L является пропозициональным, когда из множества L исключаются символы \forall и \exists , переменные, функциональные символы и все символы отношений рангов, больших 0.

В качестве доказательства (3) используется следующее:

Рассматривается характеристика синтаксическими средствами множества T формул некоей теории, описанной в семантических терминах. Иными словами, необходимо найти такое множество аксиом, или же теорем, строящихся на некоторых аксиомах, которые позволяют охарактеризовать рассматриваемую формулу.

Также рассматриваемая характеристика будет удовлетворять условию, если по предположению Лоренца, аксиоматизировать данную теорию T в некоторой конструктивной метатеории T' (T – предметная теория, строящаяся средствами L предметного языка, T' теория, изучающая предметную теорию. L' – метаязык, язык теории T' .), причем данная метатеория будет доказывать (3).

В результате перехода из семантического определения в синтаксическое, возможно получить следующие множества S :

- (i) L – пропозициональный язык, T охватывает все общезначимые формулы L . Имеется способ проверки принадлежности любой формулы к T . T возможно дедуктивно аксиоматизировать. (используется пропозициональная логика)
- (ii) T включает в себя общезначимые формулы L в полном его объеме. В L найдется функциональный символ ранга, не низшего чем 1, или же один символ отношения ранга, не низшего чем 2 (используется логика первого порядка, или же, логика предикатов)
- (iii) T включает в себя все формулы L . L содержит конкретные символы $=, 0, 1, +$ и $*$ и множество натуральных чисел. Причем каждая формула из T общезначима при каждой интерпретации некоей семантической структуры. (используется арифметика)

Теория T называется полной, если для каждого предложения p из языка L либо само p либо \bar{p} принадлежит T .

Под рангом некоего символа понимаем число операций из L , с помощью которых он образован.

Общезначимостью q определяется такое свойство q , что $\varphi 1 = 1$ для любой интерпретации φ , иными словами $| = q$, иначе, q – истина. (φ – интерпретация языка L , подробнее об ее свойствах в главе «интерпретации и структуры» в [1])

По первой теореме Гёделя (семантическое определение) (*):

Если T – перечислимая арифметическая теория. Если T непротиворечива, то T Π_1 -неполна. Если к тому же T Σ_1 -корректна, то в T существует недоказуемое и непроверяемое Π_1 -предложение.

Где сигнатура Σ – набор символов языка L , такой, что $\Sigma = (R, F, C, r)$, где R – множество символов отношений, F – множество функциональных символов, C – множество констант r – функция, сопоставляющая элементы R и F .

Π_n – множество всех истинных предложений сигнатуры Σ_n .

Теория Σ_1 -корректна тогда, когда для любого символа a , принадлежащего множеству арифметических формул Γ справедливо:

$$N | = a \rightarrow T | - a \quad (4)$$

Теория T полна тогда, когда для любого предложения a из Γ справедливо:

$$T | - a \rightarrow N | = a \quad (5)$$

Формулы (4) и (5) справедливы для арифметики Пеано.

Противоречием называем отношение двух взаимоисключающих высказываний.

Из теоремы Гёделя следует:

i – разрешима и аксиоматизируема, из чего следует, что если рассматриваемая теория T описывается пропозициональным языком, то она доказывает (3), из чего следует, что семантические термины разрешимы, или, что эквивалентно, выполнимы машиной Тьюринга.

ii – аксиоматизируема и неразрешима, из чего следует опровержение (3). Такую задачу нельзя решить на машине Тьюринга.

iii – разрешима и неаксиоматизируема, что дает аналогичный ii вывод относительно (3).

Сказанное справедливо, поскольку доказательство теоремы * требует опровержения (6):

$$N | = \varphi \rightarrow T | - \varphi \quad (6)$$

Что противоречит следствию 1 из [2] (подробнее на с.18).

Поскольку формулу (3) возможно привести к виду (6) с помощью рекурсивных функции, которые преобразуют сигнатуру Σ к сигнатуре арифметики Пеано (как это сделать подробно описано

в [5], также как и синтаксическое определение теорем о неполноте, дающее, в данном случае, аналогичные выводы), можно заявить, что для систем S типа ii и iii не являются формальными, из чего следует невыразимость семантического определения формальными теоремами, из чего следует, что задача неразрешима и невыполнима на машине Тьюринга.

Заключение: формулы (1) и (2) не равносильны друг другу. Пример этому – семантическое определение арифметики, по которому ее теоремы T полны и разрешимы, а значит, аксиоматизированы. Но при переходе к синтаксическому определению, по теореме Гёделя, получаем обратный результат. (получаем парадокс лжеца, «Это предложение ложно» - нельзя доказать истинность или ложность этого утверждения. Аналогично арифметическому « $2+2=4$ ».)

Т.е., в случае, когда необходимо доказать нечто используя машину Тьюринга, делается искусственная надстройка над сигнатурой, позволяющая определить понятие истины. Чаще всего, это некоторые перечисления, свойства и теоремы, которые задаются в качестве аксиом.

С изученными областями использование машины Тьюринга не составляет проблем, но в случае добавления новой аксиомы, или же расширения сигнатуры, мы можем прийти к внутреннему противоречию в системе. Из чего следует, что большая часть вычислений окажется неверна. Предотвратить автоматический возникновение такой ситуации не представляется пока выполнимым.

Из вышеописанного можно убедиться в необходимости построения некоей машины, которая способна оперировать понятиями истинности в отрыве от доказуемости и адаптировать свои алгоритмы исходя из этого понятия.

Согласно [6], такая машина – машина Гёделя. Данная машина оперирует некими изначальными аксиомами, где вход программы – некие теоремы, а выход – доказательство. Если в данной программе находится недоказуемая теорема, совершается поиск возможного доказательства, до тех пор, пока не будет доказано его отсутствие.

С использованием данной машины возможно реализовать алгоритмы шифрования (поскольку, по определению, единственное отличие шифра от кода в «понимании», или наличии семантического смысла), причем с помощью доказуемых теорем, а не заранее выбранных аксиом (которые, в некоторых случаях, могут привести к противоречию). Таким образом возможно создавать новые системы шифрования и автоматически проверять их принадлежность к множеству шифров.

Список использованных источников:

1. Линдон Р. Заметки по логике / перевод с англ. Гостева Ю.А., ред. Яглома Н.М. – Издательство «Мир», Москва, 1968. – 126 с.
2. Беклемишев Л.Д. Теоремы Геделя о неполноте и границы их применимости // «Успехи математических наук» т. 65, вып. 5, 2010 – с. 62-103.
3. Gruber Monika Alfred Tarski and the "Concept of Truth in Formalized Languages"- a running commentary with consideration of the Polish original and the German translation. 2016 – P.225.
4. Шень А. и др. Лекция 16. Машины Тьюринга. Тезис Чёрча-Тьюринга. // Лекции по дискретной математике, 2016, с.21
5. The undecidable. Basic Papers On Undecidable Propositions, Unsolvable Problems And Computable Functions / edited by Martin Davis / Raven Press, Hewlett, New York, 1965, - P. 434.
6. Jürgen Schmidhuber's GÖDEL MACHINE [эл. Источник]; URL: <https://people.idsia.ch/~juergen/goedelmachine.html>

UDC 510.164

FORMALIZATION AND RESEARCH OF CLOSED SYSTEMS OPERATIONS

Protsko M.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Borisenko O.F. – PhD in Physics and Mathematics

Annotation. This article will present a sample that allows you to prove the limitations of neural networks on the way to strong intelligence. Also, the information presented in this work will be sufficient to transform the natural language system into the language of formal logic with a possible correlation with semantic meaning. These properties of formal systems will allow you to choose activation functions, target functions and the architecture of neural networks in systems of machine learning.

Keywords. Formal system, Godel's incompleteness theorems, Godel's machine.

НЕЙРОСЕТЕВОЙ ПОДХОД К ПРОГНОЗИРОВАНИЮ СПОРТИВНЫХ ТЕННИСНЫХ ДАННЫХ

Харкевич А.П., студент гр.953504

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Рыкова О. В. – канд. физ.-мат. наук

Аннотация. В данной работе рассмотрены основные этапы получения готовой нейросети для прогнозирования спортивных теннисных данных, анализа качества обучения и последующей оптимизации модели.

Ключевые слова. Python, pandas, xgboost, нейросеть, прогнозирование результатов, теннис.

Спортивное прогнозирование – процесс предсказания, необходимый для планирования в процессе спортивной подготовки и работы в сфере спорта.

Планирование в спорте на основе прогнозов используется различными специалистами в сфере физической культуры и спорта, тренерскими штабами, селекционерами, «скаутами» команд, аналитическими и букмекерскими агентствами.

Прогнозирование в спорте различается по срокам и может быть краткосрочным, среднесрочным, долгосрочным и сверхдолгосрочным. Вид и результат прогноза в спорте может быть различным и в зависимости, от цели, таким результатом может быть как определенный численный результат, процент вероятности исхода того или иного состязания, так и указание наступления конкретного события во временных рамках состязания. Однако, вне зависимости от вида прогноза и ожидаемого результата, спортивное прогнозирование требует анализа набора факторов, на него влияющих. Компьютерные технологии и современные методы искусственного интеллекта позволяют производить такой анализ факторов и осуществлять прогнозирование, получая при этом результат.

Получение готовой нейросети для прогнозирования спортивных данных будет состоять из нескольких этапов:

- 1) Нахождение данных с матчами и спортсменами
- 2) Очистка данных
- 3) Построение модели нейронной сети
- 4) Их обучение на части данных
- 5) Валидация моделей на другой части данных и на реальных данных
- 6) Оптимизация модели

Рассмотрим подробнее каждый из этих этапов.

На первом этапе необходимо произвести поиск нужно нам датасета.

После проведенного поиска было обнаружено, что для текущей задачи хорошо подойдет github репозиторий с названием «tennis_atp» за авторством пользователя JeffSackmann.

Данный репозиторий содержит огромное количество данных о теннисных матчах более чем за 50 лет. Все данные в нём хранятся в формате csv файлов.

Однако с точки зрения нашей задачи в данном репозитории есть существенный недостаток – данные в нём разбиты по годам и по турнирам. С точки зрения нашей задачи хотелось бы иметь один файл, в котором все данные по годам и турнирам будут объединены. Очевидное решение данной проблемы – загрузить данные вручную или средствами библиотеки какого-либо из языков программирования.

Первый вариант был отвергнут в связи со своей неудобностью и длительностью ручных операций. Однако стоит отметить, что для более узконаправленной задачи он может являться более гибким.

При рассмотрении второго варианта был найден github репозиторий с названием «deuce» за авторством пользователя skoval.

Данный репозиторий содержит пакет, написанный на языке программирования R. Данный пакет парсит данные из репозитория Джеффа Сакмана преобразуя их в единый объект.

Для скачивания пакета была использована библиотека devtools, затем данные были загружены в код и преобразованы в csv файл.

На этом первый этап получения готовой нейросети для прогнозирования спортивных данных был завершен.

На втором этапе был написан скрипт на питоне с использованием библиотеки pandas.

Pandas — программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Pandas предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами. Название библиотеки происходит от эконометрического термина «панельные данные», используемого для описания многомерных структурированных наборов информации.

Основная область применения — обеспечение работы в рамках среды Python не только для сбора и очистки данных, но и для задач анализа и моделирования данных, без переключения на более специфичные для статобработки языки (такие, как R и Octave).

Также активно ведётся работа по реализации в pandas «родных» категориальных типов данных.

Pandas прежде всего предназначен для очистки и первичной оценки данных по общим показателям, например среднему значению, квантилям и так далее; статистическим пакетом он в полном смысле не является, однако наборы данных типов DataFrame и Series применяются в качестве входных в большинстве модулей анализа данных и машинного обучения (SciPy, Scikit-Learn и других).

Основные возможности библиотеки pandas:

- 1) Объект DataFrame для манипулирования индексированными массивами двумерных данных
- 2) Инструменты для обмена данными между структурами в памяти и файлами различных форматов
- 3) Встроенные средства совмещения данных и способы обработки отсутствующей информации
- 4) Переформатирование наборов данных, в том числе создание сводных таблиц
- 5) Срез данных по значениям индекса, расширенные возможности индексирования, выборка из больших наборов данных
- 6) Вставка и удаление столбцов данных
- 7) Возможности группировки позволяют выполнять трёхэтапные операции типа «разделение, изменение, объединение» (англ. *split-apply-combine*).
- 8) Слияние и объединение наборов данных
- 9) Иерархическое индексирование позволяет работать с данными высокой размерности в структурах меньшей размерности
- 10) Работа с временными рядами: формирование временных периодов и изменение интервалов и так далее

Библиотека оптимизирована для высокой производительности, наиболее важные части кода написаны на Cython и Си.

Т. к. количество турниров достаточно большое и многие турниры уже не актуальны, т.е. на игроков, участвующих в них, данные прогнозироваться уже не будут, первым шагом будет подвыборка нужных нам турниров. Стоит задать временной диапазон порядка 20 лет и рассматривать турниры только в этом временном диапазоне. Иначе мы будем работать с устаревшими данными, не интересными нам для прогнозирования будущих матчей, однако стоит отметить, что для другой задачи эти данные могли бы сыграть определенную полезную роль.

Также, чтобы сделать прогнозы модели более точными, мы ограничим множество матчей матчами, которые происходили на кортах с жестким покрытием. Так как от типа покрытия сильно разнится стиль игры, то на разных кортах разные игроки будут показывать разные результаты. Мы же возьмём самый распространенный тип, чтобы не вносить лишние помехи.

Следующим шагом мы переименуем названия колонок на более понятные, чтобы с ним было удобнее работать и их названия были более осмыслены.

Далее для удобства мы преобразуем год и дату старта турнира к формату даты в питоне.

Затем с помощью регулярных выражений мы запоминаем сколько выигранных игр было у выигравшего и проигравшего игрока во всех сетах, а также общее количество игр во всех играх.

Затем запоминаем число игр, где подавал только один человек (победитель или побежденный). Запоминаем также число раз, когда в игре создавался брейк-поинт. Также запоминаем число раз, когда брейк-поинт был выигран одним из игроков. Мы считаем также число удачных вторых подач игрока. Считаем мы также и такие метрики, как сколько подач было выиграно игроком и сколько очков он выиграл, когда отбивался. Считаем также общее число выигранных очков для каждого игрока и просто общее число очков в матче.

Убираем лишние колонки данных, которые не будут нам полезны. Колонки, где данные были потеряны и не заполнены, мы заполняем дефолтными или средними значениями.

Следующим шагом будет преобразование данных к нужному формату, чтобы для конкретного матча у нас была статистика игроков как для победителя, так и для проигравшего, в отдельных строках.

Затем нам нужно будет преобразовать необработанную статистику матчей игроков из абсолютных значений в относительные соотношения. Это важно, так как абсолютное значение статистики игрока зависит от длительности матча.

Например, предположим, что Федерер выиграет Кирьоса со счётом 6-4, 7-5. Всего Федерер выиграет 13 партий. Если Джокович выиграет партию у Нишикори со счётом 7-6, 3-6, 5-7, 6-2, 7-6, то он выиграет в общей сложности 28 партий, более чем в два раза больше, чем Федерер. Это несправедливое сравнение. Больше смысла имеет сравнивать их коэффициенты выигрыша в игре. Для Федерера коэффициент выигрыша в игре $(6+7)/(6+4+7+5) = 0.59$, для Джоковича коэффициент выигрыша в игре 0.51. Сравнение этих двух коэффициентов более разумно, чем использование итоговых значений.

Далее мы создадим новые метрики, по которым удобно будет обучать нашу модель.

Первой такой метрикой станет процент выигранных подач. Она рассчитывается следующим образом: число выигранных первых подач плюс число выигранных вторых подач, деленное на общее число первых подач игрока плюс общее число вторых подач игрока плюс общее число двойных ошибок игрока.

Вторая метрика — это процент выигранных очков, когда игрок отбивался: число выигранных очков, когда игрок отбивался, деленная на общее число очков, когда игрок отбивался.

Третьей и четвертой метриками являются среднее число брейкпоинтов за игру и процент реализации брейкпоинта.

Далее пятой и шестой метриками будут процент выигранных игр и процент выигранных очков.

Затем мы подсчитаем логарифм ранга игрока и с помощью экспоненциальной функции от ранга его противника считаем вес его победы.

Затем мы подсчитаем взвешенный процент выигранных игр и взвешенный процент выигранных очков в матче.

Ещё одной важной метрикой является так называемый «клатч-фактор».

Клатч – напряженный момент в самом конце игры, от которого зависит исход матча. Умение игрока выложиться на полную является важной метрикой.

Клатч-фактор мы будем рассчитывать, как разницу между процентов выигранных игр и процентом выигранных очков.

Следующим важным шагом будет суммирование статистики по данному игроку за предыдущие 10 матчей.

На рисунке 1 мы можем увидеть, как будет выглядеть часть результатов для Роджера Федерера:

tournament_date_index	player_name	player_serve_win_ratio	player_return_win_ratio	player_bp_per_game
2005-01-17	Roger Federer	0.700709	0.408396	0.732828
2006-01-16	Roger Federer	0.704199	0.433687	0.734755
2007-01-15	Roger Federer	0.727015	0.428965	0.721652
2008-01-14	Roger Federer	0.738650	0.407721	0.744655

Рисунок 1 – Часть результатов для Роджера Федерера

Эти показатели должны быть объединены с данными матча. Ключами, которые будут однозначно идентифицировать нашу группу данных, будут дата турнира и имя игрока как для player_1, так и для player_2. Пример данных после объединения с данными матча мы можем увидеть на рисунке 2.

player_1	player_2	tourney_start_date	player_serve_win_ratio_p1	player_serve_win_ratio_p2
Roger Federer	Marcos Baghdatis	2005-01-17	0.700709	0.634170
Roger Federer	Andre Agassi	2005-01-17	0.700709	0.692173
Roger Federer	Denis Istomin	2006-01-16	0.704199	0.660193
Roger Federer	Florian Mayer	2006-01-16	0.704199	0.614321
Roger Federer	Max Mirnyi	2006-01-16	0.704199	0.682761

Рисунок 2 – Пример данных после объединения с данными матча

Мы также возьмем различия между агрегатами `player_1` и `player_2`, чтобы уменьшить количество функций и, следовательно, наше время вычисления. Интуитивно это работает, потому что, выиграет ли Федерер матч, зависит от того, насколько хорош его соперник по сравнению с ним. Пример данных после взятия разницы между агрегатами можно увидеть на рисунке 3.

<code>player_1</code>	<code>player_2</code>	<code>tourney_start_date</code>	<code>player_serve_win_ratio_diff</code>
Roger Federer	Marcos Baghdatis	2005-01-17	0.066539
Roger Federer	Andre Agassi	2005-01-17	0.008536
Roger Federer	Denis Istomin	2006-01-16	0.044006
Roger Federer	Florian Mayer	2006-01-16	0.089877
Roger Federer	Max Mirnyi	2006-01-16	0.021438

Рисунок 3 – Пример данных после взятия разницы между агрегатами

В разрезе последних 10 матчей для ранга игрока и логарифмического ранга игрока мы возьмём его наиболее повторяющееся значение по матчам. Для оставшихся метрик мы считаем среднее значение скользящим окном по матчам и берем наиболее часто повторяющееся значение среднего.

Далее мы добавляем полученные новые метрики к исходным данным игроков.

И затем считаем разницу в метриках двух соперников в матче.

Таким образом функции, которые мы будем использовать для наших прогнозов, будут представлять собой разницу в средней статистике каждого игрока за предыдущие 10 матчей. Например, если мы рассматриваем матч Александра Зверева как `player_1` и Стефаноса Циципаса как `player_2`, то для Александра Зверева мы хотим усреднить его статистику (например, процент выигранных партий) по последним 10 матчам, допустим, это число 0.63. То же самое мы сделаем и для Стефаноса Циципаса, допустим, его средний коэффициент выигрыша в матчах - 0.68. Отметим, что это усредненное значение по матчам, в которых каждый игрок участвовал индивидуально, а не по их общим предыдущим поединкам.

Возьмем разницу между двумя характеристиками игроков, $0.63 - 0.68 = -0.05$ и используем ее как характеристику для прогнозирования того, выиграет ли `player_1` (Александр Зверев). Мы можем сделать это для множества других статистических данных игроков, таких как ранг игрока, процент выигранных первой и второй подачи или процент выигранных ответных очков.

Для генерации предсказаний была использована модель для классификации `XGBCClassifier` на основе `XGBOOST`.

`XGBoost` — алгоритм машинного обучения, основанный на дереве поиска решений и использующий фреймворк градиентного бустинга. В задачах предсказания, которые используют неструктурированные данные (например, изображения или текст), искусственная нейронная сеть превосходит все остальные алгоритмы или фреймворки. Но когда дело доходит до структурированных или табличных данных небольших размеров, в первенстве оказываются алгоритмы, основанные на дереве поиска решений.

`XGBoost` разрабатывался как исследовательский проект Вашингтонского Университета. Tianqi Chen и Carlos Guestrin представили их работу на конференции SIGKDD в 2016 году и произвели фурор в мире машинного обучения. С момента его введения этот алгоритм не только лидировал в соревнованиях Kaggle, но и был основой нескольких отраслевых передовых приложений. В результате образовалось общество специалистов по анализу данных, вносящих вклад в проекты `XGBoost` с открытым исходным кодом с ~350 участниками и ~3,600 коммитами на GitHub.

Особенности фреймворка:

- 1) Широкая область применения: может быть использован для решения задач регрессии, классификации, упорядочения и пользовательских задач на предсказание.
- 2) Совместимость: Windows, Linux и OS X.
- 3) Языки: поддерживает большинство ведущих языков программирования, например, C++, Python, R, Java, Scala и Julia.
- 4) Облачная интеграция: поддерживает кластеры AWS, Azure и Yarn, хорошо работает с Flink, Spark

Дерево принятия решений — простой в визуализации и достаточно понятный алгоритм. Однако не так уж просто понять следующее поколение алгоритмов, основывающихся на деревьях. Поэтому для понимания обратимся к несложной аналогии.

Представьте, что вы специалист по подбору персонала и собеседуете нескольких отличных кандидатов. Каждый шаг эволюции алгоритмов, основанных на деревьях, может быть представлен как версия хода собеседования.

- 1) Дерево принятия решений: Каждый специалист по подбору персонала при собеседовании кандидата ориентируется по своему списку критериев: образование, опыт работы, успешность прохождения собеседования.
- 2) Бэггинг: Представьте, что вместо одного специалиста по подбору персонала теперь за каждым кандидатом наблюдают несколько, и каждый имеет возможность проголосовать. Этот алгоритм при принятии окончательного решения учитывает все высказанные мнения.
- 3) Случайный лес: Этот алгоритм основан на бэггинге. Отличается он тем, что выбирает случайные признаки. То есть, каждый специалист по подбору персонала может проверить знания кандидата лишь в какой-то одной случайно выбранной области.
- 4) Бустинг: Это альтернативный подход, в котором каждый специалист по подбору персонала основывается на оценке кандидата предыдущим специалистом. Это ускоряет процесс собеседования, так как не подходящие кандидаты сразу же отсеиваются.
- 5) Градиентный бустинг: Частный случай бустинга, в котором ошибка минимизируется алгоритмом градиентного спуска. То есть, наименее квалифицированные кандидаты отсеиваются как можно раньше.
- 6) XGBoost: Экстремальный градиентный бустинг. Это идеальная комбинация оптимизации ПО и железа для получения точных результатов за короткое время с минимальным использованием вычислительных ресурсов.

XGBoost и Gradient Boosting Machines (GBM) — ансамбли методов деревьев, которые используют принцип бустинга (чаще всего, алгоритм построения бинарного дерева решений) при помощи архитектуры градиентного спуска. В свою очередь, XGBoost — улучшение фреймворка GBM через системную оптимизацию и усовершенствование алгоритма.

Системная оптимизация:

- 1) Параллелизация: В XGBoost построение деревьев основано на параллелизации. Это возможно благодаря взаимозаменяемой природе циклов, используемых для построения базы для обучения: внешний цикл перечисляет листья деревьев, внутренний цикл вычисляет признаки. Нахождение цикла внутри другого мешает параллелизовать алгоритм, так как внешний цикл не может начать своё выполнение, если внутренний ещё не закончил свою работу. Поэтому для улучшения времени работы порядок циклов меняется: инициализация проходит при считывании данных, затем выполняется сортировка, использующая параллельные потоки. Эта замена улучшает производительность алгоритма, распределяя вычисления по потокам.
- 2) Отсечение ветвей дерева: В фреймворке GBM критерий остановки для разбиения дерева зависит от критерия отрицательной потери в точке разбиения. XGBoost использует параметр максимальной глубины `max_depth` вместо этого критерия и начинает обратное отсечение. Этот «глубинный» подход значительно улучшает вычислительную производительность.
- 3) Аппаратная оптимизация: Алгоритм был разработан таким образом, чтобы он оптимально использовал аппаратные ресурсы. Это достигается путём создания внутренних буферов в каждом потоке для хранения статистики градиента. Дальнейшие улучшения, как, например, вычисления вне ядра, позволяют работать с большими наборами данных, которые не помещаются в памяти компьютера.

Улучшения алгоритма:

- 1) Регуляризация: Он штрафует сложные модели, используя как регуляризацию LASSO (L1), так и Ridge-регуляризацию (L2) для того, чтобы избежать переобучения.
- 2) Работа с разреженными данными: Алгоритм упрощает работу с разреженными данными, в процессе обучения заполняя пропущенные значения в зависимости от значения потерь. К тому же он позволяет работать с различными уровнями разреженности.
- 3) Метод взвешенных квантилей: XGBoost использует его для того, чтобы наиболее эффективно находить оптимальные точки разделения в случае работы со взвешенным датасетом.
- 4) Кросс-валидация: Алгоритм использует свой собственный метод кросс-валидации на каждой итерации. То есть, нам не нужно отдельно программировать этот поиск и определять количество итераций бустинга для каждого запуска.

XGBoost может похвастаться лучшей комбинацией «производительность-время обучения» среди других алгоритмов.

Остается лишь грамотно настроить алгоритм путём подбора гиперпараметров.

В нашем случае мы используем логистическую регрессию для бинарной классификации и на выходе получаем вероятность победы одного из игроков.

Для обучения используем 300 деревьев. Для управления взвешиванием новых деревьев, добавленных в модель, используется параметр `learning_rate`, мы установили его равным 0.02. Параметр максимальной глубины дерева мы установили равным 6.

В качестве метрики оценки для проверки данных была выбрана AUC, что отлично подходит для бинарной классификации.

Чтобы лучше понять, что такое AUC метрика разберемся сначала, что из себя представляет ROC-кривая.

Кривая ROC представляет собой график, показывающий эффективность модели классификации при всех пороговых значениях классификации. Эта кривая отображает два параметра:

- 1) True Positive Rate
- 2) False Positive Rate

True Positive Rate (TPR) определяется следующим образом:

$$TPR = TP / (TP + FN) \quad (1)$$

False Positive Rate (FPR) определяется следующим образом:

$$FPR = FP / (FP + TN) \quad (2)$$

Кривая ROC отображает соотношение TPR и FPR при различных порогах классификации. Снижение порога классификации позволяет классифицировать больше элементов как положительные, тем самым увеличивая количество ложных срабатываний и истинных срабатываний. На рисунке 4 показан пример кривой ROC.

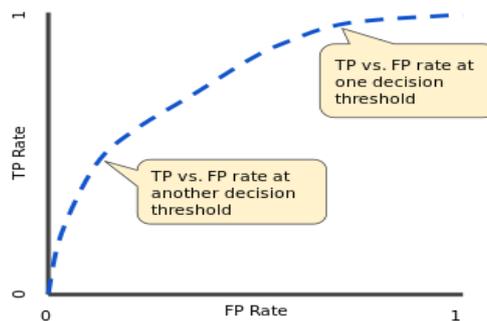


Рисунок 4 – Коэффициент TP и FP при различных порогах классификации

Чтобы вычислить точки на ROC-кривой, мы могли бы много раз оценивать модель логистической регрессии с разными порогами классификации, но это было бы неэффективно. К счастью, существует эффективный алгоритм, основанный на сортировке, который может предоставить нам эту информацию, называемый AUC.

AUC означает «Площадь под кривой ROC». То есть AUC измеряет всю двумерную область под всей кривой ROC (например, интегральное исчисление) от (0,0) до (1,1). Пример можно увидеть на рисунке 5.

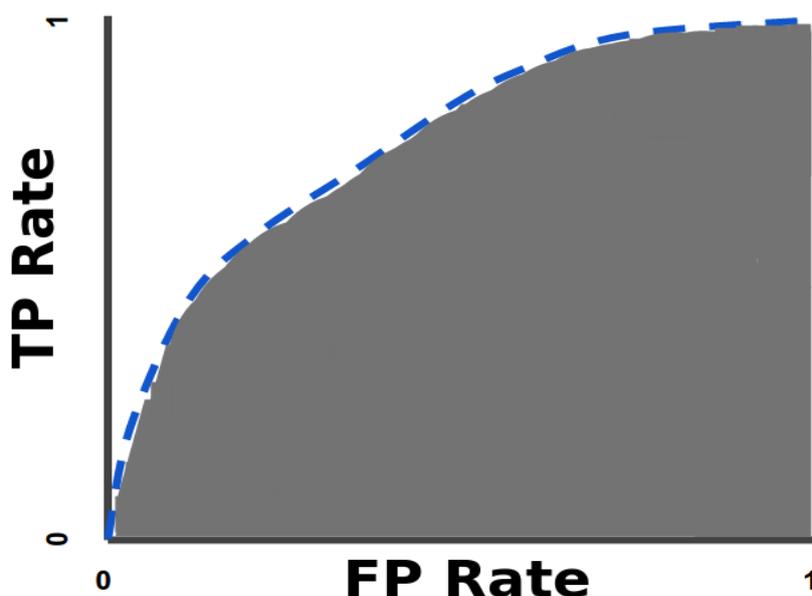


Рисунок 5 – AUC (площадь под кривой ROC)

AUC обеспечивает совокупный показатель производительности по всем возможным пороговым значениям классификации. Один из способов интерпретации AUC — это вероятность того, что модель ранжирует положительный прогноз выше, чем отрицательный прогноз.

Значение AUC варьируется от 0 до 1. Модель, чьи прогнозы на 100 % неверны, имеет значение AUC, равное 0,0; тот, чьи предсказания верны на 100%, имеет AUC 1,0.

AUC желательно использовать по следующим двум причинам:

- 1) AUC не зависит от масштаба. Он измеряет, насколько хорошо ранжируются прогнозы, а не их абсолютные значения.
- 2) AUC не зависит от порога классификации. Он измеряет качество прогнозов модели независимо от выбранного порога классификации.

Однако обе эти причины сопровождаются оговорками, которые могут ограничивать полезность AUC в определенных случаях использования:

- 1) Масштабная инвариантность не всегда желательна. Например, иногда нам действительно нужны хорошо откалиброванные по вероятности выходные данные, и AUC не скажет нам об этом.
- 2) Инвариантность порога классификации не всегда желательна. В тех случаях, когда существуют большие различия в стоимости ложноотрицательных и ложноположительных результатов, может быть важно минимизировать один тип ошибки классификации. Например, при обнаружении спама в электронной почте вы, вероятно, захотите свести к минимуму ложноположительные срабатывания (даже если это приведет к значительному увеличению ложноотрицательных результатов). AUC не является полезным показателем для этого типа оптимизации.

Обучение нашей модели досрочно заканчивается, если в течение 20 раундов обучения не было обнаружено улучшения метрики.

После выбора модели и значений гиперпараметров нужно будет разделить данные на тренировочные и данные для тестирования (валидации) модели, чтобы предотвратить перегрузку xgboost.

Разбивать данные мы будем следующим образом: первые взятые 20 лет будут использоваться для тренировки, 20 + 1 год будет использоваться для валидации и 20 + 2 год будет использоваться для тестирования.

Подключив наши тренировочные и валидационные сетки данных в модель xgboost, мы получаем окончательную валидацию AUC со значением в 0,78, которое можно увидеть на рисунке 6.

```
[164] validation_0-auc:0.78196
```

Рисунок 6 – Значения метрики AUC

Для анализа вклада функций в предсказательную мощьность мы можем использовать встроенный метод `feature_importances_`. Он, по сути, вычисляет долю времени, в течение которого функция появляется в дереве решений. Чем больше она появляется, тем больше вероятность того, что она будет сильным фактором точности прогнозирования. Значение анализа приведены на рисунке 7.

```
pd.Series(model.feature_importances_, index=X_train.columns).sort_values(ascending=False)
player_log_rank_diff          0.619010
player_game_win_ratio_diff    0.108943
player_point_win_ratio_weighted_diff 0.080545
player_serve_win_ratio_diff   0.075001
player_rank_diff              0.060340
player_return_win_ratio_diff  0.056161
dtype: float32
```

Рисунок 7 – Значения метрик модели при анализе с помощью `feature_importances`

Неудивительно, что самой существенной характеристикой, определяемой `xgboost`, является разница логарифмов рангов игроков.

Это также подтверждается ещё одним фактором, а именно важностью перестановки. Рассмотрим детальнее вопрос о том, что представляет собой важность перестановки. По сравнению с большинством других подходов важность перестановки:

- 1) можно быстро рассчитать,
- 2) она широко используется
- 3) соответствует свойствам, которые мы хотели бы иметь для меры важности признаков.

Важность перестановки использует модель не самым обычным способом, многие сначала находят это запутанным. Для понимания мы начнем с примера. Рассмотрим данные в следующем формате на рисунке 8:

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24

Рисунок 8 – Данные о человеке, доступные в возрасте, когда ему было 10 лет.

Мы хотим предсказать рост человека, когда ему исполнится 20 лет, используя данные, доступные в возрасте, когда человеку было 10 лет.

Наши данные включают полезные функции (рост в возрасте 10 лет), функции с небольшой прогностической силой (владение количеством носков), а также некоторые другие функции, на которых мы не будем акцентировать внимание в этом примере.

Важность перестановки рассчитывается после обучения модели. Таким образом, мы не будем менять модель или прогнозы, которые мы получим для заданного значения роста, количества носков и т. д.

Вместо этого мы зададим следующий вопрос: если я случайным образом перетасую один столбец проверочных данных, оставив целевой столбец и все остальные столбцы на месте, как это повлияет на точность прогнозов в этих уже перемешанных данных?

Случайное изменение порядка одного столбца должно привести к менее точным прогнозам, поскольку полученные данные больше не соответствуют ничему, наблюдаемому в реальном мире. Точность модели особенно страдает, если мы перемешиваем столбец, на который модель сильно опиралась при прогнозировании. В этом случае перетасовка роста в 10 лет вызовет ужасные

последствия. Если бы вместо этого мы перетасовали столбец владения носками, полученные в результате прогнозы не пострадали бы так сильно.

Таким образом процесс выглядит следующим образом:

1) Получаем обученную модель.
2) Перемешиваем значения в одном столбце, делаем прогнозы, используя полученный набор данных. Используем эти прогнозы и истинные целевые значения, чтобы вычислить, насколько функция потерь пострадала от перетасовки. Это ухудшение производительности измеряет важность переменной, которую мы только что перетасовали.

3) Возвращаем данные в исходный порядок (отменив перетасовку с шага 2). Теперь повторяем шаг 2 со следующим столбцом в наборе данных, пока не рассчитаем важность каждого столбца.

Рассмотрим важность перестановки для метрик нашей модели на рисунке 9:

```
perm = PermutationImportance(model).fit(X_val, y_val)
eli5.show_weights(perm, feature_names = X_val.columns.tolist())
```

Weight	Feature
0.1689 ± 0.0195	player_log_rank_diff
0.0132 ± 0.0132	player_rank_diff
0.0098 ± 0.0062	player_game_win_ratio_diff
0.0039 ± 0.0100	player_point_win_ratio_weighted_diff
0.0039 ± 0.0050	player_return_win_ratio_diff
-0.0065 ± 0.0051	player_serve_win_ratio_diff

Рисунок 9 – Значения важности перестановки для метрик модели

Значения сверху являются наиболее важными характеристиками, а значения внизу имеют наименьшее значение.

Первое число в каждой строке показывает, насколько снизилась производительность модели при случайном перемешивании (в данном случае с использованием «точности» в качестве показателя производительности).

Иногда можно увидеть отрицательные значения важности перестановок. В этих случаях предсказания перетасованных (или зашумленных) данных оказывались более точными, чем реальные данные. Это происходит, когда случайность делает прогнозы на перетасованных данных более точными. Это чаще встречается с небольшими наборами данных, такими как в этом примере, потому что здесь больше места для удачи/случайности.

Важность перестановки, по сути, заключается в том, что она включает в себя перестановку в случайном порядке очередности измерений, а также в том, как она влияет на точность прогнозирования. Если точность резко снижается, то это хороший индикатор того, что данная функция была действительно важна, если она не сильно меняется, то эта функция, вероятно, не важна для вашей модели.

Таким образом проанализировав с помощью различных способов вклад метрик в обучение приходим к выводу, что наиболее влиятельной метрикой оказалась разница логарифмов рангов игроков.

Учитывая, что большая часть инфраструктуры уже заложена, делать прогнозы теперь относительно просто.

К примеру, для практического использования мы можем использовать список игроков, с помощью `itertools` создать множество всех перестановок между парами игроков и для каждого из них вычислить результат их матча. Таким образом каждый раз не нужно будет вычислять один и тот же результат.

При прогнозах особое внимание стоит уделять игрокам, которые не имели до этого большого количества серьезных матчей, по которым есть данные. Такие игроки вносят непредсказуемость, т. к. не всегда понятно, чего от них можно ожидать. В дальнейшем в качестве развития работы можно разработать использование одного из алгоритмов «холодного старта» для таких игроков.

Также можно углубить анализ и посчитать, к примеру, среднее значение вероятностей победы игрока в матчах со всеми его потенциальными соперниками.

Дальнейшим улучшением модели будет служить внедрение `elo` (Эло).

Для начала рассмотрим, что такое Эло.

Эло — это более совершенная рейтинговая система по сравнению с формулами рейтинга, используемыми ATP и WTA.

Принцип любой системы Эло заключается в том, что рейтинг каждого игрока является оценкой его силы, и каждый матч (или турнир) позволяет нам обновлять эту оценку. Если игрок выигрывает, его рейтинг повышается; если она проигрывает, он падает.

Отличие Эло заключается в определении величины, на которую должен увеличиваться или уменьшаться рейтинг. Учитываются две основные переменные: сколько матчей уже есть у игрока в системе (то есть насколько мы уверены в предматчевом рейтинге), и рейтинг соперника.

Если вы задумаетесь об этом на мгновение, то увидите, что эти две переменные являются хорошим приближением того, как мы обычно думаем о силе игрока. Чем больше мы уже знаем об игроке, тем меньше мы изменим свое мнение на основании одного матча. Проигрыш Новака Джоковича по круговой системе Доминику Тиму в Лондоне стал неожиданностью, но только самые скептически настроенные фанаты Джоковича увидели в этом результате катастрофу, которая должна существенно изменить нашу оценку его игровых способностей. Точно так же мы корректируем наше мнение в зависимости от рейтинга оппонента. Поражение от Тима разочаровывает, но поражение, скажем, от Марко Чеккинато вызывает большее беспокойство. Система Эло включает в себя эту естественную интуицию.

Традиционно игроку при входе в систему присваивается рейтинг Эло 1200 — до того, как появятся какие-либо результаты. Это число само по себе совершенно произвольно. Все, что имеет значение, — это разница между рейтингами игроков, поэтому, если мы начнем считать рейтинг каждого участника с 0, 100 или 888, конечный результат этих различий останется прежним.

На данный момент лучшими игроками АТР и WTA являются Рафаэль Надаль и Эшли Барти с 2203 и 2123 очками соответственно. Лучшие игроки часто находятся в этом диапазоне, а лучшие из лучших часто приближаются к 2500. Согласно последней версии алгоритма, пик Джоковича был 2470, а лучший результат Серены Уильямс — 2473.

Отметка в 2000 баллов — это хорошее практическое правило, позволяющее отделить элиту от остальных. На данный момент такие высокие рейтинги имеют шесть мужчин и семь женщин. 16 мужчин и 18 женщин имеют рейтинг Эло не ниже 1900, а рейтинг 1800 примерно эквивалентен месту в топ-50.

Как только мы присвоим каждому игроку единый пиковый рейтинг, вполне естественно начать сравнивать его по эпохам.

Можно сравнить по эпохам, как каждый игрок справлялся с его конкурентами. В 1990 году Хелена Сукова достигла рейтинга 2123 — точно такого же, как сегодня у Барти. Это не значит, что Сукова тогда была так же хороша, как сейчас Барти. Но это означает, что их показатели по сравнению с их сверстниками были одинаковыми. Второй эшелон игроков был значительно слабее тридцать лет назад, поэтому добиться такого рейтинга в каком-то смысле было проще. В то время рейтинг Суковой был хорош только для 11-го места, намного отставая от 2600 Штеффи Граф.

Таким образом, Ело не позволяет вам ранжировать игроков по эпохам, если вы не уверены, что уровень конкуренции был одинаковым, или если у вас нет другого способа решить эту проблему.

Связанный с этим вопрос - инфляция Эло, которая также может усложнить сравнения между эпохами. Каждый раз, когда играется матч, победитель и проигравший эффективно «торгуют» некоторыми из своих очков, поэтому общее количество рейтинговых очков Эло в системе не меняется. Однако каждый раз, когда в систему заходит новый игрок, общее количество очков увеличивается. И всякий раз, когда игрок уходит из игры, общее количество очков уменьшается.

Было бы неплохо, если бы сложения и вычитания компенсировали друг друга, но для многих соревнований, в которых используется Эло, этого не происходит. Добавления, как правило, перевешивают вычитания, поэтому рейтинг Эло со временем увеличивается. Следовательно количество очков в системе меняется со временем, по причинам, не связанным с силой топ-игроков.

Эло дает нам рейтинг для каждого игрока, и мы получаем представление о том, что мы можем и не можем делать с ними.

Одной из основных целей любой рейтинговой системы является прогнозирование исхода матчей, что Эло делает лучше, чем большинство других, включая рейтинги АТР и WTA. Единственным входом, необходимым для составления прогноза, является разница между рейтингами двух игроков, которую затем можно подставить в следующую формулу:

$$1 - (1 / (1 + (10^{((разница)/400)}))) \quad (3)$$

Если бы мы хотели спрогнозировать матч-реванш последнего матча финала Кубка Дэвиса, мы бы взяли рейтинги Эло Надаля и Дениса Шаповалова (2203 и 1947), нашли разницу (256) и подставили ее в формулу, получив результат 81,4% для Надаля на победу. Если бы мы использовали отрицательную разницу (-256), мы бы получили в результате 18,6%, для Шаповалова на победу.

В теннисе рейтинги и прогнозы должны сильно различаться в зависимости от покрытия.

Для каждого игрока нужно вести четыре отдельных рейтинга Эло: общий, только корт с твердым покрытием, только корт с грунтовым покрытием и только корт с травяным покрытием. Например, рейтинг Тима составляет 2066 в целом, 1942 на харде, 2031 на грунте и 1602 на траве. (Поверхностные рейтинги, как правило, ниже: грунтовый рейтинг Тима занимает третье место, намного опережая всех, кроме Надаля и Джоковича.)

Рейтинги на одной поверхности говорят нам, как бы мы ранжировали игроков, если бы просто отбрасывали результаты на всех остальных поверхностях. Однако это неверно. Односторонние рейтинги не очень хороши для прогнозирования результатов матчей. Лучшим решением будет взять смесь 50/50 одноповерхностных и общих рейтингов. Если бы мы хотели предсказать шансы Тима в матче на грунтовом корте, мы бы использовали половинную смесь его общего рейтинга 2066 и его рейтинга на грунтовом корте 2031.

Однако стоит отметить, что не существует естественного закона, который диктует смесь рейтингов именно 50/50.

Спросите у фанатов тенниса, какие матчи турниров имеют большее значение для рейтингов — и вы получите длинный подробный список факторов, определяющих это значение. Например, финалы или олимпийские игры являются более важными, чем обычные личные встречи. Для таких матчей нужно вносить определенные корректировки.

Эло предусматривает такие корректировки. Коэффициент, обычно называемый «коэффициентом k », позволяет придавать больший вес определенным матчам. Это распространено в рейтингах Эло и для других видов спорта, например, при использовании более высокого коэффициента k для постсезонных игр, чем для игр регулярного сезона.

Обычно этот коэффициент полагают равным 10 для сильнейших игроков (рейтинг 2400 и выше), 20 (было 15) — для игроков с рейтингом меньше, чем 2400 и 40 (было 30) — для новых игроков (первые 30 партий с момента получения рейтинга ФИДЕ), а также для игроков до 18 лет, рейтинг которых ниже 2300.

В нашем случае этот коэффициент будет зависеть от числа выигранных и проигранных матчей игрока.

Когда игроки пропускают значительное количество времени, нужно снижать их рейтинг, а затем повышать коэффициент k на несколько матчей после их возвращения.

Эти шаги являются логическим продолжением системы Эло. Если игрок травмирован на несколько месяцев, мы никогда не знаем, чего ожидать, когда он вернется. Может быть, он так же силен, как и прежде; может быть, он уже на шаг медленнее. Возможно, что он быстро вернется в норму, но также он может и никогда полностью не вернуться в форму. Длительное отсутствие вызывает много вопросов. Игрок с травмой редко возвращается в лучшей форме, чем до ухода, в то время как многие игроки возвращаются хуже, что дает нам средний уровень производительности после травмы, который хуже, чем до её отсутствия.

Следовательно, когда игрок впервые возвращается, наша оценка должна составляться таким образом будто бы он играет немного хуже. Однако некоторым сильным ранним результатам следует придать больший вес — отсюда и более высокий коэффициент k . Коэффициент k отражает тот факт, что сразу после отсутствия игрока мы не так уверены в своей оценке, как обычно.

Алгоритм усложняется, но логика проста. По сути, это просто попытка выработать строгую версию утверждений вроде: «Я не знаю, насколько хорошо он будет играть, когда вернется, но я буду внимательно следить».

Одним из дополнительных преимуществ штрафа за отсутствие является то, что он противодействует естественной склонности Эло к завышению рейтингов. В то время как больше игроков входит в систему, чем выходит из нее, штраф удаляет некоторые очки, не перераспределяя их другим игрокам.

Самый простой способ сравнить рейтинговые системы — это показатель под названием «точность», который подсчитывает правильные прогнозы. В финале Кубка Дэвиса было 50 одиночных матчей, и Эло правильно выбрал победителя в 36 из них с рейтингом точности 72%. Рейтинг АТР правильно выбрал победителя (в том смысле, что игрок с более высоким рейтингом выиграл матч) в 30 из них с рейтингом точности 60%. В этом маленьком эксперименте Эло превзошел официальные рейтинги. Эло был также значительно лучше в течение всего сезона.

Лучшей метрикой для сравнения рейтинговых систем является оценка Брайера, которая учитывает достоверность каждого прогноза. Ранее мы видели, что Эло дает Надалю шанс победить Шаповалова в 81,4%. Если Надаль в конечном итоге выигрывает, 81,4% — это более хороший прогноз, чем, скажем, 65%, но это хуже прогноза 90%. Оценка Брайера представляет собой квадрат расстояния между прогнозом (81,4%) и результатом (0% или 100%, в зависимости от победителя) и усредняет эти числа для всех прогнозируемых матчей. Он вознаграждает близкие к 100 прогнозы, которые оказываются верными, но поскольку он использует квадрат расстояния, он сурово наказывает близкие к 100, но неверные прогнозы.

Более интуитивный способ понять, к чему ведет счет Брайера, — представить, что Надаль и Шаповалов играют 100 матчей подряд. (Или, более точно, но менее интуитивно, представьте, что 100 одинаковых Надалей играют одновременные матчи против 100 одинаковых Шаповаловых.) Прогноз 81,4% означает, что мы ожидаем, что Надаль выиграет в 81 матче. Если Надаль в итоге выиграет 90, прогноз не был достаточно благоприятным для него. Мы никогда не получим 100 таких одновременных матчей, но у нас есть тысячи отдельных матчей, многие из которых имеют одинаковые прогнозы, например, 60-процентный шанс на победу фаворита. Оценка Брайера объединяет все эти пары «прогноз-результат» и выдает число, сообщающее нам, какой у нас результат.

Трудно прогнозировать результат отдельных теннисных матчей. Любая система, какой бы сложной она ни была, в большинстве случаев будет ошибаться. Во многих случаях «правильный» прогноз едва ли лучше, чем отсутствие прогноза вообще, если данные свидетельствуют о том, что конкуренты равны. Таким образом, «точность» имеет ограниченное применение — более важно иметь правильную степень уверенности в победе, чем просто выбирать победителей.

Рейтинги Эло имеют гораздо более низкие (лучшие) баллы Брайера, чем прогнозы, полученные на основе рейтингов ATP и WTA.

Оценка Брайера также является мерой, которая говорит нам о том, является ли определенная корректировка, такая как смешивание поверхностей, отсутствие травм или тип турнира, улучшением системы. Штраф за травму снижает оценку Брайера в общем наборе прогнозов Эло, поэтому мы сохраняем эту корректировку. А уменьшение k-фактора для матчей первого раунда не имеет никакого эффекта, поэтому мы не используем эту корректировку.

Теперь зная информацию об оценке Эло мы сможем внедрить её в качестве метрики в нашу модель. Проведем предрасчёт рейтинга Эло для каждого из спортсменов. Затем проведем очистку данных, такую же как проводили для прошлой модели, добавив рейтинг Эло в качестве дополнительного параметра для обучения.

Обучим нашу модель аналогично тому, как делали это до этого и посмотрим на результаты. Проанализировав вклад функций в обучение, получим следующий результат на рисунке 10.

```
pd.Series(model.feature_importances_, index=X_cols_subset_1).sort_values(ascending=False)
```

player_old_elo_diff	0.440281
player_win_weight_diff	0.120508
player_log_rank_diff	0.079821
player_rank_diff	0.079811
player_game_win_ratio_diff	0.078194
player_point_win_ratio_weighted_diff	0.068934
player_return_win_ratio_diff	0.066744
player_serve_win_ratio_diff	0.065708

Рисунок 10 – Значения метрик модели с Эло при анализе с помощью feature_importances

Как мы видим, Эло сыграло очень важную роль в обучении модели в несколько раз обходя остальные метрики.

Рассмотрим теперь важность перестановок.

Weight	Feature
0.1173 ± 0.0320	player_old_elo_diff
0.0528 ± 0.0242	player_win_weight_diff
0.0291 ± 0.0162	player_log_rank_diff
0.0213 ± 0.0118	player_game_win_ratio_diff
0.0157 ± 0.0100	player_rank_diff
-0.0016 ± 0.0094	player_point_win_ratio_weighted_diff
-0.0024 ± 0.0107	player_return_win_ratio_diff
-0.0039 ± 0.0141	player_serve_win_ratio_diff

Рисунок 11 – Значения важности перестановки для метрик модели с Эло

И здесь Эло показывает прекрасный результат, обходя остальные метрики.

Таким образом исходя из анализа метрик можно утверждать, что логарифм ранга игрока оказывает большее влияние на обучение, чем просто ранг игрока. Но Эло в свою очередь обходит как ранг игрока, так и логарифм ранга игрока по важности влияния на обучение.

Возьмём определенный список игроков, между которыми были сыграны матчи, и они известны. Пример части игроков приведен на рисунке 12.

```
'Rafael Nadal',
'Hugo Dellien',
'Federico Delbonis',
'Joao Sousa',
'Christopher Eubanks',
'Peter Gojowczyk',
'Jozef Kovalik',
'Pablo Carreno Busta',
'Nick Kyrgios',
'Lorenzo Sonogo',
```

Рисунок 12 – Пример части игроков взятых для валидации

Введем вручную результаты этих матчей. Пример части результатов матчей приведен на рисунке 13.

```
['Rafael Nadal', 'Hugo Dellien', 1],
['Hugo Dellien', 'Rafael Nadal', 0],
['Federico Delbonis', 'Joao Sousa', 1],
['Joao Sousa', 'Federico Delbonis', 0],
['Christopher Eubanks', 'Peter Gojowczyk', 0],
['Peter Gojowczyk', 'Christopher Eubanks', 1],
['Jozef Kovalik', 'Pablo Carreno Busta', 0],
['Pablo Carreno Busta', 'Jozef Kovalik', 1],
```

Рисунок 13 – Пример части результатов матчей взятых для валидации

Таким образом мы имеем список игроков и известные результаты матчей, на которых мы сможем провалидировать качество наших обученных моделей.

Для валидации предсказаний моделей будем использовать следующие метрики accuracy, AUC, logloss.

Метрика AUC была описана выше.

Рассмотрим две другие метрики.

Логарифмическая потеря – метрика оценки эффективности Модели Бинарной классификации.

Для того, чтобы разобраться в том, что это, обратимся к концепции бинарной классификации. Такой алгоритм сначала предсказывает вероятность того, что Наблюдение будет отнесено к классу 1, а затем причисляет его к одному из двух классов (1 или 0) на основе того, пересекла ли вероятность пороговое значение, которое устанавливается по умолчанию равным 0,5. На рисунке 14 приведены результаты классификации email (1 – "спам").

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	ПРЕДСКАЗАННЫЙ КЛАСС
64247	Попробуйте бесплатную защиту от телефонного спама и выиграйте один	1	0,95	1
26591	Привет! Прости пожалуйста что пропал – очень много было работы. В целом, ничего не меняет	0	0,2	0
70282	Здравствуйте, Елена Александровна! Запрошенная вами справка "О наличии счетов"	0	0,65	1

Рисунок 14 – Результаты классификации email

Итак, прежде чем предсказывать класс записи, модель должна спрогнозировать вероятность того, что запись будет отнесена к классу 1. Помните, что именно от этой вероятности предсказания записи данных зависит значение логарифмической потери.

Логарифмическая потеря указывает, насколько близка вероятность предсказания к соответствующему истинному значению (0 или 1 в случае бинарной классификации). Чем больше прогнозируемая вероятность отклоняется от фактического значения, тем выше значение логарифма потерь. Формула расчета Log-Loss будет приведена чуть позже.

Для примера рассмотрим задачу классификации электронных писем. Давайте представим спам как класс 1, а класс "нормальных" писем как 0. Давайте изучим настоящее спам-письмо (фактическое значение равно 1) и статистическую модель, которая классифицирует это письмо как спам с вероятностью 1. Поскольку вероятность предсказания равна почти 1, то и разность между предсказанной вероятностью и фактическим классом равна почти 0. Нулю равен, следовательно, и логарифм этой разности. Пример такого предсказания приведен на рисунке 15:

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	РАЗНОСТЬ	ЛОГАРИФМИЧЕСКАЯ ПОТЕРЯ
1318	Попробуйте бесплатную защиту от телефонного спама и выиграйте один из пяти iPhone 12 на 256 ГБ...	1	1,00	0,00	0

Рисунок 15 – Результаты предсказания с вероятностью 1

Рассмотрим на рисунке 16 еще одно спам-письмо, классифицированное как спам с вероятностью 0,9. Вероятность прогноза модели на 0,1 отличается от фактического значения 1, и, следовательно, значение логарифмической потери больше нуля (равно 0,105).

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	РАЗНОСТЬ	ЛОГАРИФМИЧЕСКАЯ ПОТЕРЯ
64247	Привет. Надеюсь, у тебя было время посмотреть, какие услуги по строительству предлагает моя компания...	1	0,90	0,10	0,105

Рисунок 16 – Результаты предсказания с вероятностью 0.9

А теперь давайте посмотрим на рисунке 17 на обычное электронное письмо. Модель классифицирует его как спам с вероятностью 0,2, то есть считает нормальным письмом (при условии, что порог по умолчанию равен 0,5). Абсолютная разница между вероятностью предсказания и фактическим значением, равным 0 (так как это нормально), составляет 0,2, что больше, чем то, что мы наблюдали в предыдущих двух наблюдениях. Значение логарифма потерь, связанное с прогнозом, составляет 0,223.

ID	ТЕКСТ ПИСЬМА	РЕАЛЬНЫЙ КЛАСС	ПРЕДСКАЗАННАЯ ВЕРОЯТНОСТЬ	РАЗНОСТЬ	ЛОГАРИФМИЧЕСКАЯ ПОТЕРЯ
2101	Скажи, когда тебе необходимо настроить рабочий ПК. У меня есть...	0	0,20	-0,20	0,223

Рисунок 17 – Результаты предсказания с вероятностью 0.2

Обратите внимание, как теперь значение Log-Loss худшего прогноза (удаленного от фактического значения) выше, чем у лучшего прогноза (ближе к фактическому значению).

Теперь предположим, что существует набор из 5 различных спам-писем, прогнозируемых с широким диапазоном вероятностей 1.0, 0.7, 0.3, 0.009 и 0.0001. Обученная статистическая модель неидеальна и, следовательно, выполняет (действительно) плохую работу по последним трем наблюдениям (классифицирует их как нормальные, поскольку значения вероятности ближе к 0, чем к 1). На рисунке 18 видно, что значение логарифмических потерь экспоненциально возрастает по мере того, как растет разность между реальным классом и предсказанной вероятностью:

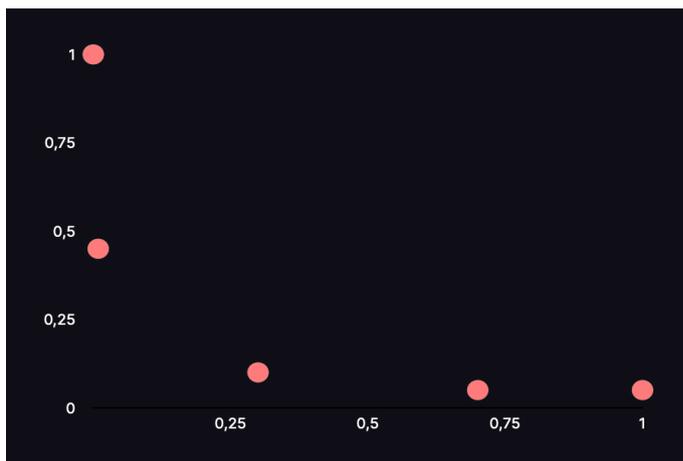


Рисунок 18 – Значения логарифмических потерь при прогнозах с разной вероятностью
Если мы построим график логарифмических потерь для перечня спам-писем со всеми возможными видами вероятностей, график будет выглядеть как на рисунке 19.

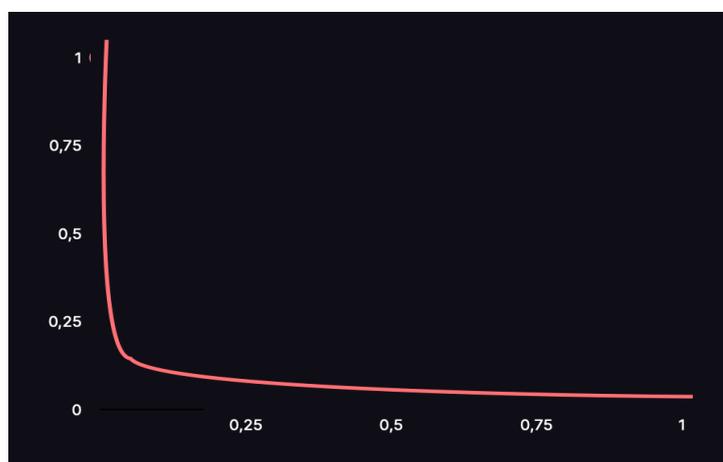


Рисунок 19 – Значения логарифмических потерь при всех возможных вероятностях прогнозов

В случае с нормальными письмами график будет зеркальным отображением приведенного выше и будет изображаться как на рисунке 20.

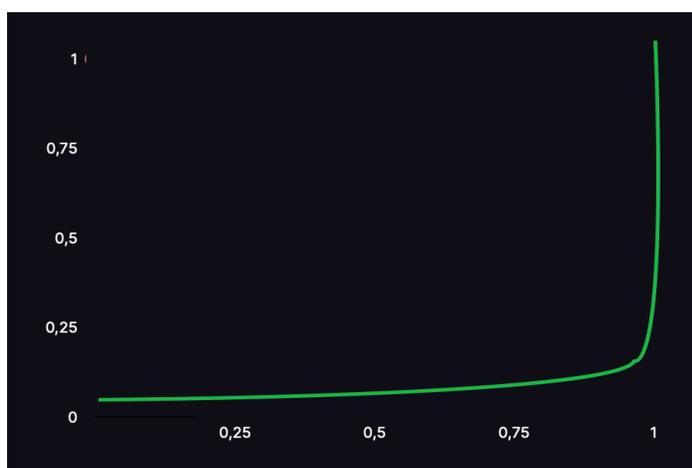


Рисунок 20 – Значения логарифмических потерь при всех возможных вероятностях прогнозов для нормальных писем

Подводя итог, можно сказать, что чем дальше вероятность предсказания от фактического значения, тем выше значение логарифмических потерь. При обучении модели классификации мы хотели бы, чтобы наблюдение предсказывалось с вероятностью, максимально приближенной к

фактическому значению (0 или 1). Следовательно, Log-Loss – хороший выбор в качестве функции потери для обучения и оптимизации. Чем дальше вероятность предсказания от ее истинного значения, тем выше штраф.

Теперь, когда мы понимаем логику, лежащую в основе метрики, мы можем посмотреть на конечную формулу на рисунке 21.

$$\text{Log-Loss} = -[y_i \times \ln(p_i) + (1 - y_i) \times \ln(1 - p_i)]$$

y_i – истинный класс наблюдения,

p_i – предсказанная вероятность

Рисунок 21 – Формула логарифмических потерь

Чтобы оценить модель в целом, вычисляется среднее арифметическое логарифмических потерь всех наблюдений. Модель с совершенными предсказаниями имеет логарифм потерь, равным нулю. Другими словами, идеальная модель предсказывает вероятность каждого наблюдения как фактическое значение.

Логарифмическая потеря для бинарной классификации – тоже, что и среднеквадратическая ошибка для регрессии. Обе метрики показывают, насколько хороши или плохи результаты прогнозов, указывая на дистанцию между прогнозом и фактическим значением.

Логарифмические потери можно рассчитать с помощью SkLearn. Для начала импортируем функцию:

```
import matplotlib.pyplot as plt
```

Применим функцию "на бегу", передав аргументы-списки:

```
log_loss(["спам", "нормальное письмо", "нормальное письмо", "спам"],
[[.1, .9], [.9, .1], [.8, .2], [.35, .65]])
```

Система здесь уже вынесла свой вердикт, и во втором списке находятся, как можно догадаться, пары значений, описывающих вероятности "спам" и "нормального письма". Функция сама определяет, что первый элемент внутреннего списка – число 0,1, описывает вероятность письма быть нормальным письмом, а второй – соответственно, спамом, и применив такой паттерн ко всем остальным парам значений, вычисляет разность между реальным классом и предсказанной вероятностью. Следуя формуле, описанной выше, она находит значение Log-Loss для каждого наблюдения и усредняет полученный результат получая в итоге 0.21616.

Ассигасу — это показатель, который описывает общую точность предсказания модели по всем классам. Это особенно полезно, когда каждый класс одинаково важен. Он рассчитывается как отношение количества правильных прогнозов к их общему количеству.

Метрика асс содержит результат деления суммы True Positive и True Negative прогнозов на количество всех прогнозов. Таким образом, ассигасу, равная 0.5714, означает, что модель с точностью 57,14% делает верный прогноз.

В модуле sklearn.metrics есть функция precision_score(), которая также может вычислять ассигасу. Она принимает в качестве аргументов достоверные и предсказанные метки.

Стоит учесть, что метрика ассигасу может быть обманчивой. Один из таких случаев — это несбалансированные данные. Предположим, у нас есть всего 600 единиц данных, из которых 550 относятся к классу Positive и только 50 — к Negative. Поскольку большинство семплов принадлежит к одному классу, ассигасу для этого класса будет выше, чем для другого.

Если модель сделала 530 правильных прогнозов из 550 для класса Positive, по сравнению с 5 из 50 для Negative, то общая ассигасу равна $(530 + 5) / 600 = 0.8917$. Это означает, что точность модели составляет 89.17%. Полагаясь на это значение, вы можете подумать, что для любой выборки (независимо от ее класса) модель сделает правильный прогноз в 89.17% случаев. Это неверно, так как для класса Negative модель работает очень плохо.

Для вычисления значения метрик возьмём следующие модели: нашу первую модель, модель в которой мы ввели новую метрику эло для анализа и "наивную" модель, где вероятность победы каждого из игроков равна 0.5.

Получаем следующие результаты, которые можем увидеть на рисунке 22

	logloss	AUC	accuracy
model	0.529107	0.800347	0.712598
elo_model	0.515281	0.825118	0.763780
true_naive	9.654653	0.720486	0.720472

Рисунок 22– Результаты валидации моделей

На основе результатов можно сказать следующее: первичная модель и модель на основе Эло показали сравнимые хорошие показатели. Однако Эло модель по всем показателям обошла первичную.

В то же время модель, всегда отдающая результат 0.5 сильно уступила лишь в logloss метрике. Это подчеркивает важность выбора метрики для валидации модели обучения. Некоторые метрики являются неинформативными, в другое время как остальные помогают качественно провалидировать прогнозы полученной модели.

Дальнейшими перспективами развития данной работы можно назвать вычисление оптимального размера скользящего окна по матчам для агрегирования метрик, оптимальной длительности обучения модели и обучения модели для различных типов кортов.

В ходе данной работы было получено понимание того, что выбор правильных метрик для обучения зачастую важнее подбора гиперпараметров для обучения.

А также выявлена необходимость подбора правильной метрики для валидации, чтобы она была информативной.

Список использованных источников:

1. Forecasting the results of the world winner universiade [Electronic Resource] / ScienceReview. – Mode of access: <https://science-engineering.ru/ru/article/view?id=1238>. Date of access: 07.04.2023.
2. ATP Tennis Rankings, Results, and Stats [Electronic Resource] / JeffSackmann. – Mode of access: https://github.com/JeffSackmann/tennis_atp. Date of access: 07.04.2023.
3. Deuce [Electronic Resource] / Skoval. – Mode of access: <https://github.com/skoval/deuce>. Date of access: 07.04.2023.
4. Introduction to R [Electronic Resource] / Ahmedushka7. – Mode of access: https://ahmedushka7.github.io/R/scripts/hse_data_analysis/sem_1/introduction_to_R.html. Date of access: 07.04.2023.
5. Parsing data from sites [Electronic Resource] / Ringostat. – Mode of access: <https://blog.ringostat.com/ru/parsing-dannyh-s-saytov-chto-eto-i-zachem-on-nuzhen/>. Date of access: 07.04.2023.
6. Nuances of Programming [Electronic Resource] / Medium. – Mode of access: <https://medium.com/nuances-of-programming>. Date of access: 07.04.2023.
7. XGBoost Model [Electronic Resource] / Techcave. – Mode of access: <https://techcave.ru/posts/81-sozdaem-pervuyu-xgboost-model-na-python-s-ispolzovaniem-scikit-learn.html>. Date of access: 07.04.2023.
8. Permutation importance [Electronic Resource] / Kaggle. – Mode of access: <https://www.kaggle.com/code/dansbecker/permutation-importance/tutorial>. Date of access: 07.04.2023.
9. ROC and AUC metrics [Electronic Resource] / Google. – Mode of access: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>. Date of access: 07.04.2023.
10. Elo rating ranges [Electronic Resource] / TennisAbstract. – Mode of access: <http://www.tennisabstract.com/blog/2019/12/03/an-introduction-to-tennis-elo/>. Date of access: 07.04.2023.
11. Numpy [Electronic Resource] / PythonWorld. – Mode of access: <https://pythonworld.ru/numpy/1.html>. Date of access: 07.04.2023.
12. Log-loss Metric [Electronic Resource] / Helenkapatsa. – Mode of access: <https://www.helenkapatsa.ru/logharifmichieskaia-potieria>. Date of access: 07.04.2023.
13. Accuracy Metric [Electronic Resource] / PythonRu. – Mode of access: <https://pythonru.com/baza-znaniy/metriki-accuracy-precision-i-recall>. Date of access: 07.04.2023.

UDC

NEURAL NETWORK APPROACH TO PREDICTION OF SPORTS TENNIS DATA

Kharkevich A.P.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Rykova O.V. – PhD in Physics and Mathematics

Annotation. In this paper, we consider the main stages of obtaining a ready-made neural network for predicting sports tennis data, analyzing the quality of training, and then optimizing the model.

Keywords. Python, pandas, xgboost, neural network, predicting results, tennis.

ИСКУССТВЕННЫЕ НЕЙРОННЫЕ СЕТИ НА ОСНОВЕ РАДИАЛЬНО-БАЗИСНЫХ ФУНКЦИЙ

Глинский И.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Смирнова И.А. – ассистент

В данной работе исследуются радиально-базисные сети - тип искусственной нейронной сети, который использует радиально-базисные функции в качестве активационных функций скрытого слоя. Эти сети характеризуются простой архитектурой и способностью быстро обучаться на нелинейных зависимостях. Также в качестве примера используется применение данной нейронной архитектуры на задаче обнаружения экзопланет из данных, собранных с помощью космического телескопа Кеплер.

Искусственная нейронная сеть - это вычислительная модель, имитирующая структуру человеческого мозга с целью обучения на основе данных и прогнозирования. Она содержит связанные между собой узлы (нейроны), размещенные в слоях, и адаптирует веса связей в процессе обучения для улучшения производительности.

Радиально-базисные функции (RBF) - это класс функций, использующих радиальные расстояния в многомерных пространствах для аппроксимации сложных функций и алгоритмов машинного обучения. Радиальные расстояния определяются как меры расстояния между точками, основанные на радиальной симметрии и их относительном положении.

Радиально-базисная функция в общем виде:

$$\varphi(x, c) = \varphi(\|x - c\|) \quad (1),$$

где φ - радиальная функция, x - входной вектор данных, c - центр радиальной функции, $\| \cdot \|$ - норма.

RBF используются в нейронных сетях как функции активации в скрытых слоях. Архитектура RBF-сетей включает в себя входной слой, скрытые слои с радиальными базисными функциями и выходной слой. Радиальные базисные функции позволяют обрабатывать непрерывные входные данные и определять расстояние между входными данными и центром функции. Центры нейронов и радиусы определяются в процессе обучения, и выходной слой производит результат обработки входных данных.

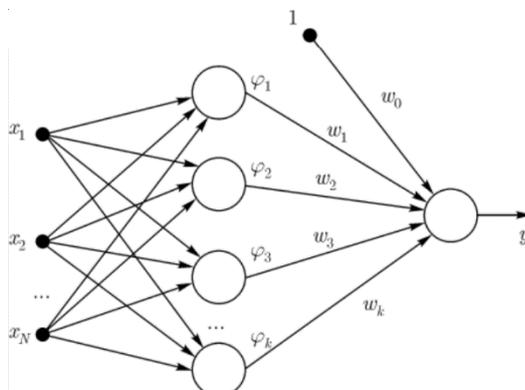


Рисунок 1 – радиально-базисная сеть [1]

Существует несколько типов радиально-базисных функций, которые могут быть использованы в радиально-базисных сетях. Некоторые из них:

Гауссовская функция:

$$\varphi(x, c) = e^{(-\sigma|x-c|^2)} \quad (2),$$

Где φ - радиальная функция, x - входной вектор данных, c - центр радиальной функции, σ – параметр масштабирования

Мультикватратичная функция:

$$\varphi(x, c) = \sqrt{||x - c||^2 + \sigma^2} \quad (3),$$

Где φ - радиальная функция, x - входной вектор данных, c - центр радиальной функции, σ – параметр масштабирования.

Обучение радиально-базисной сети состоит из двух этапов:

1. Определение центров RBF c_i и параметров масштабирования для каждого нейрона i в скрытом слое.

2. Обучение весов w_i и смещения b между скрытым и выходным слоем.

Выход скрытого слоя нейрона i для входного вектора x :

$$h_i(x) = \varphi(||x - c_i||) \quad (4),$$

где φ - радиально-базисная функция, c_i - центр RBF для нейрона i .

Выход сети RBF-сетей для входного вектора x :

$$y(x) = \sum_{i=1}^n (w_i * h_i(x)) + b \quad (5),$$

где w_i - весовой коэффициент, связывающий нейрон i скрытого слоя с выходным слоем, b – смещение, h_i – выход i нейрона, а Σ - сумма по всем нейронам i скрытого слоя.

Также в работе было проведено исследование - обнаружение экзопланет, анализируя интенсивность света звезд. Выборка содержит временные ряды потока света от тысяч звезд и бинарные классификации наличия экзопланет. Данные получены от космического телескопа Кеплер NASA.

Обучающая выборка: 5087 строк, 3198 столбцов; 37 экзопланетных звезд, 5050 неэкзопланетных звезд. Тестовая выборка: 570 строк, 3198 столбцов; 5 экзопланетных звезд, 565 неэкзопланетных звезд.

Анализ показал нормальное распределение интенсивности светового потока, что обеспечивает основу для использования радиально-базисной сети с гауссовской функцией для обнаружения экзопланет.

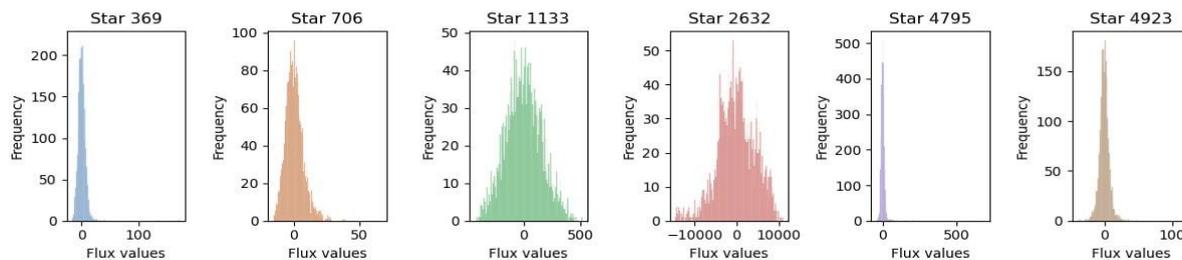


Рисунок 2 – графики распределения интенсивности светового потока

После проведения обучения модели в среде Python с использованием библиотеки TensorFlow была достигнута высокая точность в обнаружении экзопланет, равная 99 процентам. Это результат свидетельствует об эффективности применения искусственных нейронных сетей, основанных на радиально-базисных функциях, для анализа данных о природных явлениях, полученных с помощью космического телескопа Кеплер. Таким образом, успешно решается сложная задача поиска экзопланет в астрофизике с использованием современных методов машинного обучения.

Радиально-базисные сети является мощным и гибким инструментом машинного обучения, способным аппроксимировать сложные нелинейные зависимости с высокой точностью. Их успешное применение на примере задачи астрофизики по обнаружению экзопланет, где достигнута точность 99%, демонстрирует возможности радиально-базисных сетей в решении сложных задач в разнообразных областях науки и техники.

Список использованных источников:

1. *Radial basis functions, multi-variable functional interpolation and adaptive networks* / D. S. Broomhead, D. Lowe // *Royal Signals and Radar Establishment Malvern (United Kingdom)*, 1988. – P. 1-35.
2. *NASA Exoplanet Archive* – [Электронный ресурс]. – Электронные данные. – Режим доступа: : <https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=koi>

РАЗЛИЧНЫЕ ПОДХОДЫ К РЕШЕНИЮ БАЗЕЛЬСКОЙ ЗАДАЧИ

Войченко М.М., гр. 224402

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Цегельник В.В. – доктор физ.-матем. наук, профессор

Задача нахождения суммы бесконечного ряда обратных квадратов:

$$1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots = \sum_{n=1}^{\infty} \frac{1}{n^2} \quad (1)$$

называется Базельской задачей. Первым сумму ряда сумел найти Леонард Эйлер, уроженец швейцарского города Базель.

Зная разложение в ряд Маклорена функции

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \quad (2)$$

Эйлеру в 1735 году удалось получить разложение этой же функции в бесконечное произведение

$$\sin x = x \left(1 - \frac{x^2}{\pi^2}\right) \left(1 - \frac{x^2}{4\pi^2}\right) \left(1 - \frac{x^2}{9\pi^2}\right) \left(1 - \frac{x^2}{16\pi^2}\right) \dots \quad (3)$$

Приравняв правые части (2) и (3) и сокращая на x , получим

$$\left(1 - \frac{x^2}{\pi^2}\right) \left(1 - \frac{x^2}{4\pi^2}\right) \left(1 - \frac{x^2}{9\pi^2}\right) \left(1 - \frac{x^2}{16\pi^2}\right) \dots = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \dots \quad (4)$$

Приравняв в тождественном равенстве (1) коэффициенты при x^2 , получаем

$$-\left(\frac{1}{\pi^2} + \frac{1}{4\pi^2} + \frac{1}{9\pi^2} + \frac{1}{16\pi^2} + \dots\right) = -\frac{1}{6} \quad (5)$$

Умножим обе части равенства на π^2 и получим ответ на первоначальную задачу:

$$\left(\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots\right) = \frac{\pi^2}{6} \quad (6),$$

Однако предложенное доказательство не является достаточно обоснованным. Эйлер полагал, что левая и правая части равенства (4) (рассматриваемые как многочлены) имеют одинаковые корни $0; \pm\pi; \pm 2\pi; \pm 3\pi, \dots$. На самом деле левая и правая части (4) представляют собой бесконечные ряды по степеням x .

В 1741 году Эйлер нашел строгое доказательство суммы ряда (1) [1-2]. Позже доказательства равенства (6) были получены многими авторами, используя различные подходы.

Рассмотрим предложенный сравнительно недавно геометрический метод доказательства равенства (6)[3]. Будем использовать упрощённую физическую формулу определения видимой яркости звезды $L = \frac{1}{d^2}$, где d – расстояние до звезды. Будем считать, что для наблюдателя яркость нескольких звёзд равна сумме их собственных яркостей (рис. 1).

Теперь возьмём окружность с периметром 2 и поместим наблюдателя диаметрально противоположно звезде. Диаметр круга равен $D = 2/\pi$, а видимая яркость звезды соответственно $\pi^2/4$ (рис. 2)

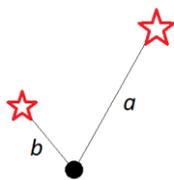


Рисунок 1 – $L = \frac{1}{a^2} + \frac{1}{b^2}$

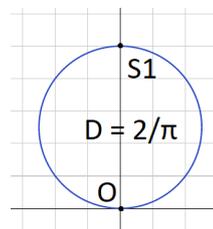


Рисунок 2

“Разобьём” звезду $S1$ на две другие, сохраняя общий видимый свет. Для этого увеличим диаметр круга в 2 раза и разместим 2 новые звезды таким образом, чтобы они образовывали прямоугольный треугольник вместе с наблюдателем (рис. 3). Общий видимый свет сохраняется благодаря инвертированной теореме Пифагора. Повторим эту операцию ещё раз для каждой звезды (рис 5). Повторив данную операцию бесконечное число раз, нижняя часть окружности “выпрямляется”, при этом между звёздами расстояние остается равным 2 (рис. 6).

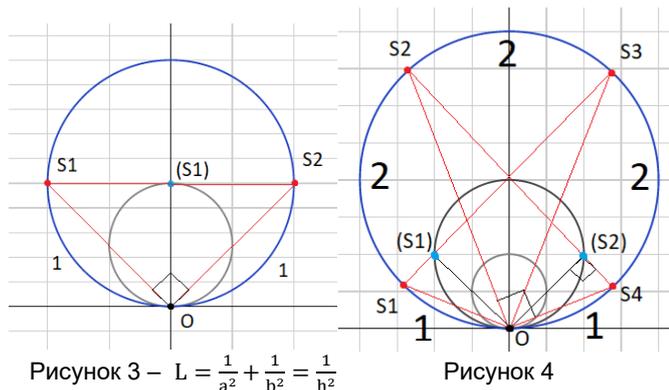


Рисунок 3 – $L = \frac{1}{a^2} + \frac{1}{b^2} = \frac{1}{h^2}$

Рисунок 4

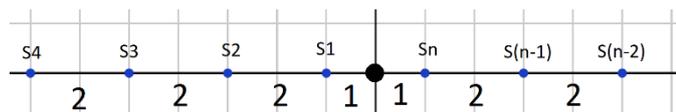


Рисунок 5

Запишем общий видимый свет и получим

$$\dots + \frac{1}{3^2} + \frac{1}{1^2} + \frac{1}{1^2} + \frac{1}{3^2} + \dots = \frac{\pi^2}{4} \quad (7)$$

$$\frac{2}{1^2} + \frac{2}{3^2} + \frac{2}{5^2} \dots = \frac{\pi^2}{4} \quad (8)$$

$$\frac{1}{1^2} + \frac{1}{3^2} + \frac{1}{5^2} \dots = \frac{\pi^2}{8} \quad (9)$$

Для того, чтобы прийти к изначальной задаче, возьмём сумму обратных квадратов четных чисел:

$$\frac{1}{4} + \frac{1}{16} + \frac{1}{36} + \dots \quad (10)$$

$$\frac{1}{4} \left(\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \dots \right) \quad (11)$$

Из чего делаем вывод, что сумма всех обратных квадратов четных чисел составляет 25% от суммы обратных квадратов всех натуральных чисел, значит (9) составляет оставшиеся 75% суммы. Отсюда получаем окончательный ответ:

$$\left(\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \dots \right) = \frac{\pi^2}{8} * \frac{100}{75} = \frac{\pi^2}{6} \quad (12)$$

Список использованных источников:

1. К. П. Кохась // Сумма обратных квадратов, Матем. просв., сер. 3, **8**, Изд-во МЦНМО, М., 2004, 142–163
2. Г.М. Курс дифференциального и интегрального исчисления // Физматлит т.2. 864 с. М., 2003;
3. Johan Wastlund // Summing inverse squares by Euclidean geometry, Department of Mathematics, Chalmers University of Technology, 2010, P. 6-8

САМОЕ ИРРАЦИОНАЛЬНОЕ ЧИСЛО

Головин Е.С., Жук Я.С.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Баркова Е.А. – канд. физ.-мат. наук

Более и менее рациональные числа – относительно новая концепция, связанная с анализом иррациональных чисел и возможностью наиболее точного их приближения к рациональным. Данная концепция стала ещё одним способом объяснить причину, по которой число ϕ столь распространено в природе.

В данной работе рассматривается метод записи иррациональных чисел в виде бесконечных цепных дробей. Исходя из этого, приводится объяснение связи значения каждого i -го коэффициента дроби с возможностью их рационализации, то есть нахождения такого дробного числа, которое будет наиболее точно аппроксимировать его.

Целью данного исследования является разработка и анализ метода для определения характеристики аппроксимации числа, а также установления связи иррациональных соотношений между различными величинами в природе.

Метод представления чисел в виде цепных дробей: Цепная дробь – это конечная (в случае рациональных чисел) или бесконечная (в случае иррациональных) последовательность, в которой числа представляются в следующем виде:

$$x = [x] + \frac{1}{x_1 + \frac{1}{x_2 + \frac{1}{x_3 + \dots}}}$$

где x – некоторое нецелое число, $[x]$ – целая часть числа, x_i – коэффициенты дроби. Алгоритм представления чисел в данном виде следующий:

1. Целая часть числа отбрасывается, ведется работа только с нецелой частью.
2. Нецелая часть переворачивается, в итоге получается число, большее единицы. Его целая часть записывается в x_i коэффициент. Второй шаг повторяется до тех пор, пока необходимая точность не будет достигнута, либо не будет получено целое число.

Связь между дробными коэффициентами и степенью рациональности числа. Иррациональные числа при записи их описанным ранее способом будут всегда образовывать бесконечные цепные дроби. Однако при анализе дробных коэффициентов можно найти достаточно близкие по значению рациональные числа. При анализе установлено, что на возможность аппроксимации с высокой точностью может указывать большое значения x_{k+1} коэффициента, на котором планируется отбросить оставшуюся часть цепной дроби. Связано это с тем, что максимальное значение оставшейся бесконечной части цепной дроби равно единице:

$$\frac{x_k}{x_{k+1} + \frac{1}{x_{k+2} + \frac{1}{x_{k+3} + \frac{1}{x_{k+3+\dots}}}}} \leq \frac{x_k}{x_{k+1} + 1}$$

На основании написанного нами выше, отношение значения оставшейся цепной дроби к x_{k+1} не превосходит следующего значения:

$$\frac{1}{x_{k+2} + \frac{1}{x_{k+3} + \frac{1}{x_{k+4} + \frac{1}{x_{k+5+\dots}}}}} < \frac{1}{x_{k+1}}$$

Таким образом, мы получили, что чем больше x_{k+1} , тем меньшую часть значения имеет оставшаяся цепная дробь и, следовательно, тем легче для данного числа находится близкое по значению рациональное число.

Связь между иррациональностью чисел и расположением семян. Рассмотрим задачу расположения семян у цветков с точки зрения оптимальной стратегии для их максимального количества на ограниченной территории. Предположим, что цветы могут выбирать способ размещения. Один из возможных методов - размещение вдоль различных направлений в плоскости, образуя равномерный круг, разделенный на несколько частей. В случае расположения нескольких рядов на ограниченной территории, самым оптимальным решением является такое, при котором каждое семя не находится на территории другой.

Это возможно, если выбрать в качестве частоты вращения иррациональное число. При этом каждый полный оборот будет отклоняться от предыдущего на некоторый угол, так как невозможно поделить окружность на иррациональное число так, чтобы получилось целое число, означающее период. Из всех иррациональных чисел необходимо выбрать то, которое наиболее удалено от целых чисел. Таким образом, найти зависимость степени «кривых» и «прямых» рядов пыльцы от иррациональности числа является необходимым для решения данной задачи.

Проблема заключается в том, что простое увеличение значения иррационального числа может привести к бесконечности, что не соответствует реальности. Наши исследования показали, что существует способ решения, который позволяет найти оптимальное значение, что подтверждают предыдущие исследования в этой области. С помощью анализа поведения расположения семян при «частоте вращения» равной первому и второму приближениям одних из самых известных в математике чисел - π и e , была найдена зависимость, описанная ранее. Кроме того, было найдено число, которое частично выполняет поставленную задачу – максимальное заполнение пространства. Этим иррациональным числом является число 0.116..., которое можно представить в виде бесконечной цепной дроби $[0, 8, 11, 8, 11, 8, 11, \dots]$. Как следует из вышеизложенного, оно должно создавать сложные узоры. Однако, если построить график, используя это число, то мы увидим, что его "цветок" не имеет особо сложной структуры, а выглядит как простой радиальный узор. Это происходит вследствие того, что в цепной дроби этого числа используются два относительно больших по значению различных числа – 8 и 11, которые, чередуясь между собой, создают хороший, однако просматриваемый узор. Таким образом, мы получаем относительно равномерное распределение пыльцы вдоль окружности.

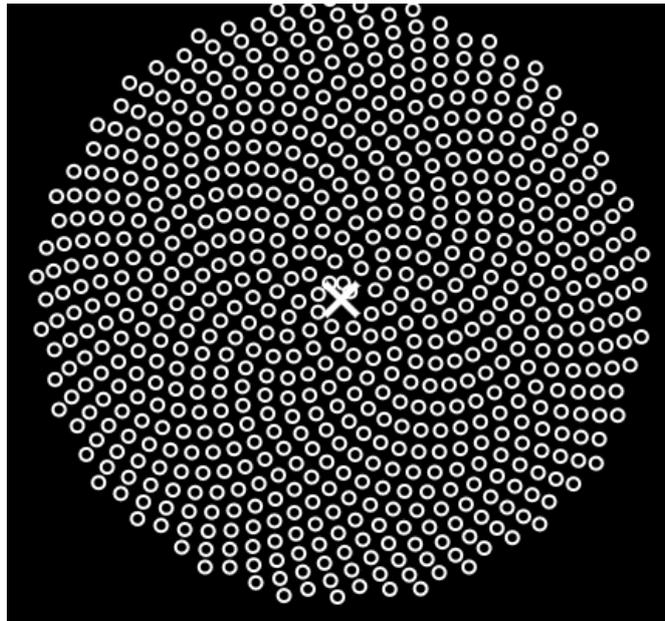


Рисунок 7 – Расположение семян при «частоте вращения» равной числу 0.116...

Следуя изложенному выше, можно с легкостью подобрать такое число, для которого рационализация будет иметь наибольшую погрешность. Таковым является число, у которого в цепной дроби в коэффициентах встречаются только единицы:

$$1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \frac{1}{\dots}}}}$$

При попытках подсчитать данную дробь ее значение стремится к золотому сечению. Одной из самых важных и занимательных особенностей такого представления чисел является то, что чем менее оно рационально, тем более хаотично и равномерно оно заполняет какую-то площадь, а чем более оно рационально, тем больше проявляются паттерны и появляется пустое пространство. Особенность числа ϕ состоит в том, что если взять его в качестве частоты размещения, то пространство заполняется наиболее плотно:

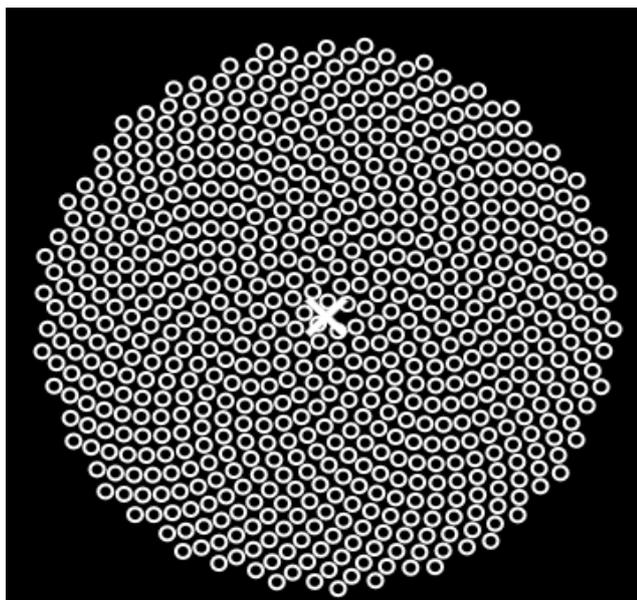


Рисунок 8 - Расположение семян при «частоте вращения» равной золотому сечению

Список использованных источников:

1. Непрерывная дробь [Электронный ресурс]. – Режим доступа https://ru.wikipedia.org/wiki/Непрерывная_дробь
Режим доступа: 03.04.2023.
2. The Golden Ratio: Why it is so irrational [Электронный ресурс]. – Режим доступа: <https://www.numberphile.com/videos/the-golden-ratio-why-it-is-so-irrational> – Дата доступа: 03.04.2023.

ПРЕДАСТАВЛЕНИЕ БИНАРНОГО ОТНОШЕНИЯ С ПОМОЩЬЮ МАТРИЦЫ

Карлюк П.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Рачковский Н.Н. – канд. физ.-мат. наук

Рассмотрены рефлексивные, симметричные, асимметричные, транзитивные бинарные отношения на конечном множестве и их представление с помощью квадратных матриц, а также построение транзитивного пополнения и транзитивной оболочки бинарного отношения.

Под бинарным отношением на множестве A понимают подмножество декартова произведения этих множеств: $M \subseteq A \times A$, то есть $aMb \Leftrightarrow (a; b) \in M$ [1].

Пусть есть конечное множество $A = \{a_i, a_j, \dots, a_n\}$ и бинарное отношение $M \subseteq A \times A$. Тогда определим матрицу M этого бинарного отношения размеров $n \times n$ по правилу:

$$m_{ij} = \begin{cases} 1, & \text{если } (a_i, a_j) \in M, \\ 0, & \text{если } (a_i, a_j) \notin M. \end{cases} \quad (1)$$

Рассмотрим 4 свойства бинарных отношений, а именно рефлексивность, симметричность, асимметричность и транзитивность.

- **Рефлексивность.**
Бинарное отношение M на множестве A называется рефлексивным, если всякий элемент множества находится в отношении с самим собой ($\forall a \in A: aMa$). В матрице бинарного отношения: $m_{ii} = 1$, где $i = 1, 2, \dots, n$.
- **Симметричность.**
Бинарное отношение M на множестве A называется симметричным, если для каждой пары элементов множества (a_i, a_j) выполнение отношения a_iMa_j влечет a_jMa_i ($\forall a, b \in A: aMb \Rightarrow bMa$). Симметричность бинарного отношения означает симметричность матрицы этого бинарного отношения.
- **Асимметричность.**
Бинарное отношение называется асимметричным, если означает выполнение следующего условия: $\forall a_i, a_j \in A: a_iMa_j \Rightarrow \neg a_jMa_i$. Для матрицы асимметричного бинарного отношения справедливо соотношение: $m_{ij} \neq m_{ji}$.
- **Транзитивность.**
Бинарное отношение M на множестве A называется транзитивным, если для любых трех элементов a_i, a_j, a_k выполнение отношений a_iMa_j и a_jMa_k влечет выполнение отношения a_iMa_k ($\forall a_i, a_j, a_k \in A: a_iMa_j \wedge a_jMa_k \Rightarrow a_iMa_k$). Матрица транзитивного бинарного отношения обладает следующим свойством: $m_{ij} = 1, m_{jk} = 1 \Rightarrow m_{ik} = 1$. Определим матрицу M' следующим образом: $m_{ij}' = \min \{1; c_{ij}\}$, где c_{ij} – элементы матрицы $M \cdot M$. Матрица M будет являться матрицей транзитивного бинарного отношения в том и только том случае, когда $M' = M$. Матрица M' определяет бинарное отношение, которое назовем транзитивным пополнением исходного бинарного отношения M [2].

Также к рассмотрению будут взяты отношение эквивалентности, отношение предпорядка и отношение строгого предпорядка.

- **Отношение эквивалентности (\sim)** – бинарное отношение, являющееся симметричным, рефлексивным, транзитивным. Для матрицы отношения эквивалентности справедливы следующие соотношения:

$$m_{ii} = 1; m_{ij} = m_{ji}; m_{ij} = 1, m_{jk} = 1 \Rightarrow m_{ik} = 1. \quad (2)$$

- **Отношение предпорядка (\preceq)** – бинарное отношение, которое является рефлексивным и транзитивным. Матрица отношения предпорядка удовлетворяет следующим условиям:

$$m_{ii} = 1; m_{ij} = 1, m_{jk} = 1 \Rightarrow m_{ik} = 1. \quad (3)$$

- Отношение строгого предпорядка ($<$) – ассиметричное и транзитивное бинарное отношение. Для матрицы отношения строгого предпорядка справедливо [3]:

$$m_{ii} = 0; m_{ij} \neq m_{ji}; m_{ij} = 1, m_{jk} = 1 \Rightarrow m_{ik} = 1. \quad (4)$$

Для отношения предпорядка \leq рассмотрим максимальное по включению входящее в него отношение эквивалентности \sim и отношение строгого предпорядка $<$. Для получения матрицы M_{\sim} отношения эквивалентности \sim применяются операция поэлементного умножения матриц. Поэлементно умножив матрицу M_{\leq} отношения предпорядка \leq на транспонированную к ней матрицу, получим матрицу эквивалентности:

$$M_{\leq} \circ M_{\leq}^T = M_{\sim}. \quad (5)$$

В свою очередь, матрицу $M_{<}$ отношения строгого предпорядка $<$ можно получить, отняв от матрицы M_{\leq} отношения предпорядка матрицу M_{\sim} отношения эквивалентности:

$$M_{\leq} - M_{\sim} = M_{<}. \quad (6)$$

Транзитивное замыкание (транзитивная оболочка) бинарного отношения M на множестве A есть наименьшее транзитивное отношение на множестве A , включающее M . Для приведения матрицы транзитивного отношения к матрице транзитивной оболочки необходимо прибегнуть к описанной выше операции транзитивного пополнения до тех пор, пока не будет выполняться условие:

$$(M_k)^{\cdot} = M_k, k = 1, 2, 3, \dots, n, \quad (7)$$

где $M_1 = M^{\cdot}$; $M_k = (M_{k-1})^{\cdot}$.

Поскольку множество A конечно, то для получения транзитивной оболочки достаточно провести операцию транзитивного пополнения конечное число раз.

Пример. Рассмотрим бинарное отношение на множестве $(a_1, a_2, a_3, a_4, a_5, a_6)$, задаваемое

матрицей $M = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$. Для этого бинарного отношения построим транзитивную оболочку.

$$M^{\cdot} = M \cdot M = \begin{pmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}, M_1^{\cdot} = M_1 \cdot M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 2 & 0 \\ 0 & 0 & 2 & 2 & 0 \\ 0 & 1 & 1 & 2 & 0 \\ 1 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

$$M_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}, M_2^{\cdot} = M_2 \cdot M_2 = \begin{pmatrix} 2 & 0 & 0 & 0 & 2 \\ 0 & 2 & 3 & 3 & 0 \\ 0 & 1 & 2 & 2 & 0 \\ 0 & 2 & 3 & 3 & 0 \\ 2 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

$$M_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}, M_3^{\cdot} = M_3 \cdot M_3 = \begin{pmatrix} 2 & 0 & 0 & 0 & 2 \\ 0 & 2 & 3 & 3 & 0 \\ 0 & 1 & 2 & 2 & 0 \\ 0 & 2 & 3 & 3 & 0 \\ 2 & 0 & 0 & 0 & 2 \end{pmatrix}.$$

$$M_4 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Так как матрица M_3 равна матрице M_4 из этого следует, что транзитивная оболочка построена.

Список использованных источников:

1. Мальцев А.И. Алгебраические системы / Мальцев А.И. – М.: Наука, 1970. – 390 с.
2. Капитонова Ю.В., Кривой С.Л., Летичевский А.А. Лекции по дискретной математике. – Санкт-Петербург, БХВ-Петербург, 2004. – 20 с.
3. Куликов Л.Я. Алгебра и теория чисел: учеб. пособие для педагогических институтов. – М.: Высшая школа, 1979. – 559 с.

МЕТОДЫ ПЕРЕМНОЖЕНИЯ МАТРИЦ. СРАВНЕНИЕ ЭФФЕКТИВНОСТИ АЛГОРИТМА ШТРАССЕНА И ОБЫЧНОГО ПЕРЕМНОЖЕНИЯ МАТРИЦ

Кулик М.И., Киселёв А.С.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Баркова Е.А. – канд. физ.-мат. наук, доцент

Данная научная работа исследует различные способы перемножения матриц и сравнивает эффективность обычного метода и алгоритма Штрассена. Описаны основные принципы алгоритмов и их особенности. Проведены эксперименты на различных размерах матриц и проанализирована теоретическая сложность. Результаты показали, что алгоритм Штрассена может значительно ускорить процесс перемножения больших матриц, но может быть менее эффективен для небольших матриц. Разработана модифицированная версия алгоритма Штрассена на базе стандартного алгоритма Штрассена и обычного алгоритма перемножения матриц, которая работает эффективнее стандартного алгоритма Штрассена в среднем на 28%.

Алгоритм Штрассена [1] использует для перемножения матрицы размера $2^k * 2^k$, где $k \in \mathbb{N}$. Однако любую матрицу можно привести к матрице $2^k * 2^k$, дополнив столбцы и строки матрицы незначительными нулями. Метод разбивает каждую из исходных матриц на четыре блока одинакового размера, вычисляет семь промежуточных матриц и рекурсивно умножает эти промежуточные матрицы для получения результата умножения исходных матриц.

Для сравнения двух алгоритмов перемножения введем термин «единица работы» (далее – ед.р.). Сложение и вычитание принимается за 1 ед.р., умножение же варьируется. В настоящей работе умножение принимается за 2.5 ед.р. Эта величина отражает сложность выполнения данного действия в компьютере. Также введем термин «база рекурсии». Это размер матрицы, до которого выполняется разбиение исходной матрицы на подматрицы.

Очевидно, что при перемножении двух матриц $n * n$ необходимо n^3 операций умножения и $n^2 * (n - 1)$ операций сложения, следовательно для перемножения двух аналогичных матриц необходимо $3.5n^3 + n^2$ ед.р.

Определим зависимость количества операций сложения и умножения матриц для Алгоритма Штрассена от их размера:

$$g(k) = 7^k \tag{1}$$

где $g(k)$ – функция количества умножений, $k = \lceil \log_2 n \rceil$, n – размер матрицы.

Для сложений получим:

$$\begin{cases} f(k) = 18 * 4^{k-1} + 7 * f(k - 1), & k > 0 \\ f(k) = 18, & k = 0 \end{cases} \tag{2}$$

где $f(k)$ – функция количества сложений, $k = \lceil \log_2 n \rceil$, n – размер матрицы.

На основании данных уравнений получим следующий график (рисунок 1):

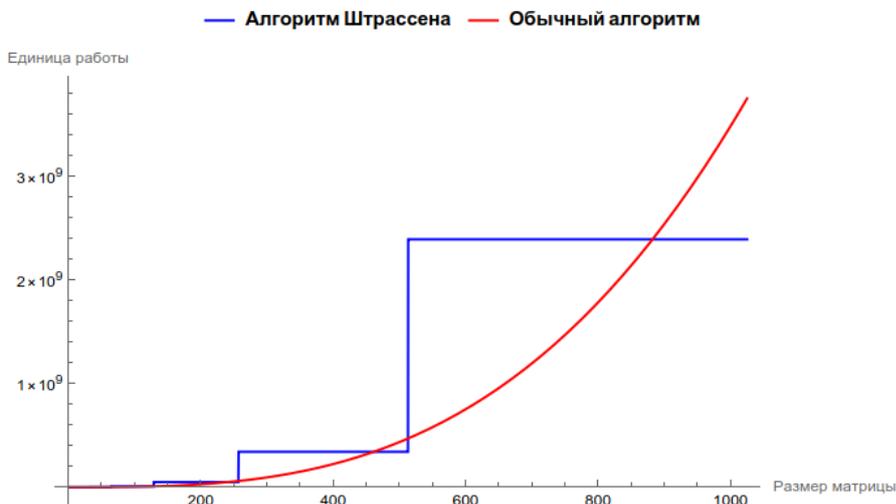


Рисунок 1 – Сравнение двух алгоритмов по количеству ед.р.

На практике были получены следующие результаты (рисунок 2):

Сравнение двух алгоритмов

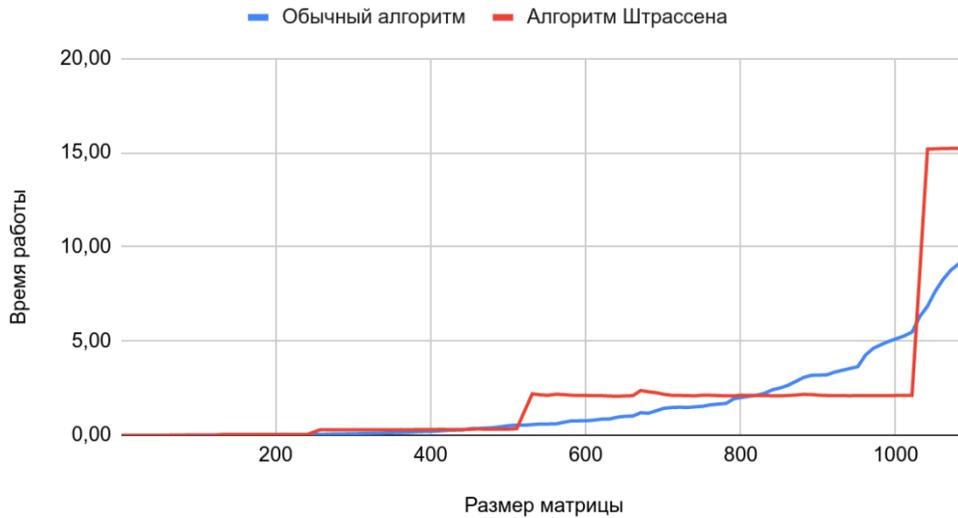


Рисунок 2 – Сравнение двух алгоритмов по количеству ед.р.

При разложении до матрицы размера $2^m * 2^m$ (где $m \in Z, m \geq 0$) (при смешении двух способов перемножения матриц), когда достигается размер матриц 2^m , в ход идет обычный алгоритм, поэтому будет верна данная система уравнений для расчета количества сложений:

$$\begin{cases} f(k) = 18 * 4^{k-1} + 7 * f(k-1), & k > m \\ f(k) = 2^{2m}(2^m - 1), & k \leq m \end{cases} \quad (3),$$

где $f(k)$ – функция количества сложений, $k = \lceil \log_2 n \rceil$, n – размер матрицы, m – база рекурсии.

Для расчета количества умножений при разложении до матрицы размера $2^m * 2^m$ получим систему:

$$\begin{cases} g(k) = 7^{k-m} * 2^{3m}, & k > m \\ g(k) = 2^{3m}, & k \leq m \end{cases} \quad (4),$$

где $g(k)$ – функция количества умножений, $k = \lceil \log_2 n \rceil$, n – размер матрицы, m – база рекурсии.

Аналитическая запись формулы:

$$f(k) = 6 * 4^k * \left(\left(\frac{7}{4} \right)^{k-m} - 1 \right) + 7^{k-m} * 4^m * (2^m - 1) \quad (5),$$

где $f(k)$ – функция количества сложений, $k = \lceil \log_2 n \rceil$, n – размер матрицы, m – база рекурсии.

На основе приведенных формул в программном средстве «Wolfram Mathematica» был построен график (рисунок 3) алгоритма Штрассена (для различных баз рекурсии (для $m = \overline{0, 4}$) и обычного алгоритма.

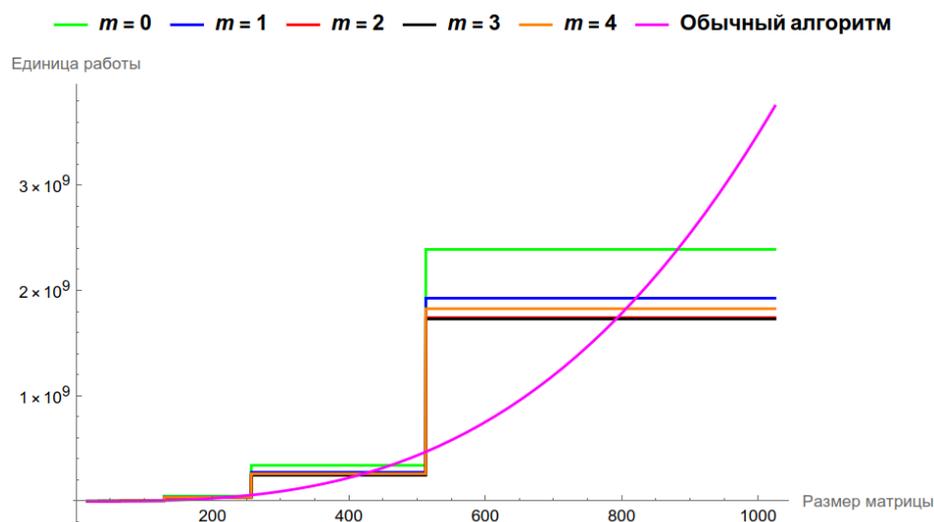


Рисунок 3 – Сравнение алгоритмов Штрассена с базами рекурсии 1, 2, 4, 8, 16

По формуле 5 вычислим количество ед.р. для алгоритма Штрассена и модифицированного алгоритма Штрассена для матриц размера $2^k * 2^k$, где $k = 3, 10$ и рассчитаем абсолютный и относительный выигрыш модифицированного алгоритма Штрассена относительно стандартного алгоритма Штрассена.

В результате были получены данные (таблица 1), согласно которым модифицированный алгоритм Штрассена оказывается эффективнее стандартного алгоритма Штрассена в среднем на 28%.

Таблица 1 – Превосходство модифицированного алгоритма над обычным алгоритмом Штрассена

Размер	Абсолютный выигрыш, ед. р.	Относительный выигрыш, %
8	803,5	31,74007505
16	5624,5	29,80262286
32	39372	28,79838497
64	275601	28,25398973
128	1929203,5	27,95213257
256	13504424,5	27,78254138
512	94530971,5	27,68655289
1024	661716800,5	27,63199953

С использованием разработанного нами программного средства мы проверили эти данные. В результате был получен следующий график (рисунок 4).

Алгоритм Штрассена с разными базами рекурсии

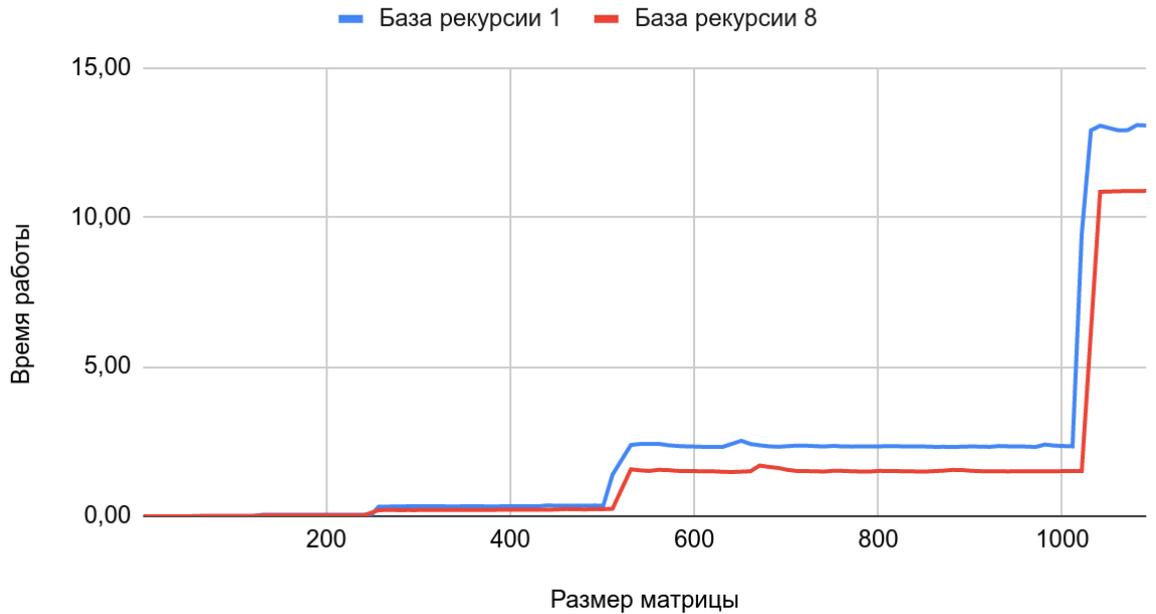


Рисунок 4 – Сравнение алгоритмов Штрассена с базами рекурсии 1 и 8

Таким образом, мы исследовали алгоритмы перемножения матриц и нашли способ их улучшения (модифицировали стандартный алгоритм Штрассена за счёт увеличения базы рекурсии с 1 до 8). Улучшенная версия алгоритма Штрассена работает эффективнее (в среднем на 28%) на всех размерах матриц. Наши исследования позволяют оптимизировать алгоритмы перемножения матриц и повысить их производительность на практике, в частности, в научной сфере, работе с графикой, машинном обучении.

Список использованных источников:

1. Strassen V. *Gaussian Elimination is not Optimal* / Numer. Math / F. Brezzi — Springer Science+Business Media, 1969. — Vol. 13, Iss. 4. — P. 354-356.

БИНАРНАЯ ПРОБЛЕМА ГОЛЬДБАХА

Крутько А.А., Трубач К.И.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Баркова Е.А. – канд. физ.-мат. наук, доцент

Данная работа посвящена рассмотрению вопроса бинарной проблемы Гольдбаха, установлению нового рекорда в количестве проверенных на соответствие гипотезе чисел, а также поиску зависимостей между чётным натуральным числом и относительным количеством его разбиений Гольдбаха.

В 1742 году математик Христиан Гольдбах послал письмо Леонарду Эйлеру [1], в котором он высказал следующее предположение: каждое нечётное число, большее 5, можно представить в виде суммы трёх простых чисел. Эйлер заинтересовался проблемой и выдвинул более сильную гипотезу: каждое чётное число, большее двух, можно представить в виде суммы двух простых чисел. В этом и заключается бинарная проблема Гольдбаха.

Для сбора данных (точек и их количества пар Гольдбаха) нами были использованы встроенные средства языка C#, такие как методы длинной арифметики. Помимо этого, реализованы более эффективные, чем решето Эратосфена, способы проверки числа на простоту (с помощью теста Миллера-Рабина). Также было разработано программное средство, предоставляющее возможность получить указанные точки в удобном для обработки формате. Поскольку установленный рекорд проверенного на соответствие гипотезе числа равен $4 \cdot 10^{18}$ [2], наша программа начала работать именно с этого значения. По состоянию на 02:45 2 апреля 2023 года проверено число 4000000004770320679, то есть рекорд превзойдён практически на 5 миллиардов чисел.

Если пронаблюдать за количеством разложений для разных чисел, становится заметно, что существуют такие числа, у которых пар Гольдбаха значительно больше, чем у их левого и правого соседей, – точки выброса. С использованием упомянутого программного средства нами был получен график, представленный на рисунке 2 и изображающий на плоскости все точки, связывающие чётные натуральные числа и количество их разложений Гольдбаха.

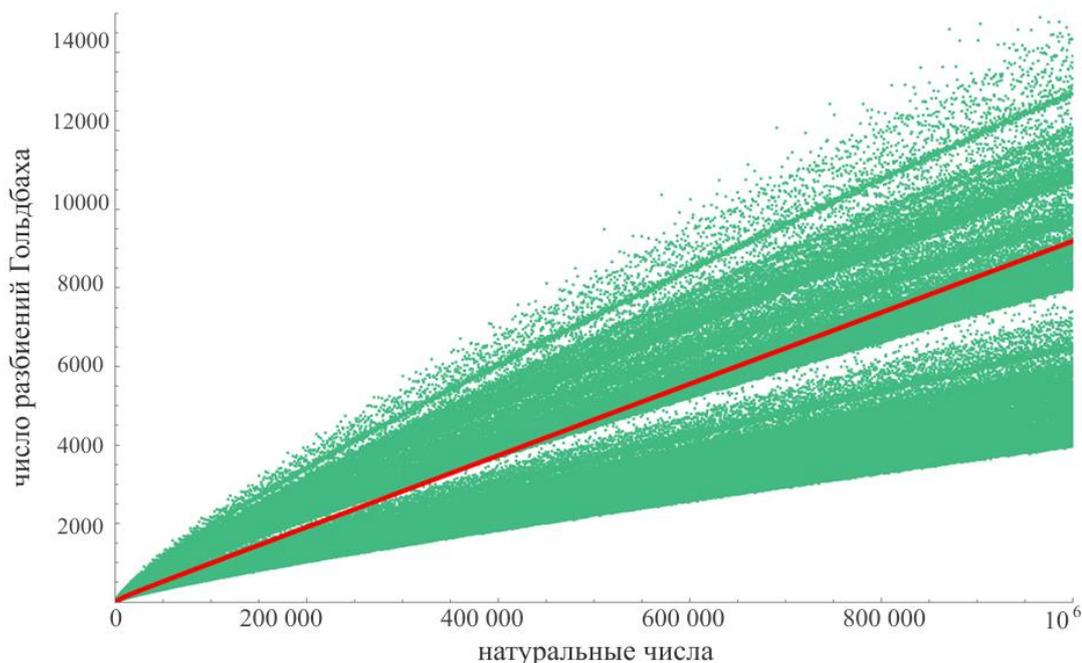


Рисунок 1 – распределение точек, а также усредненное значение (красная линия)

Поскольку любое следующее простое число можно получить из суммы определённых предыдущих простых, а также ввиду того, что для получения любого числа достаточно иметь некоторое количество двоек и троек, очевидно, что, если x среди простых множителей содержит и 2, и 3, оно обладает большим числом пар по сравнению с, например, k , имеющим в разложении только 2. Отсюда вытекает три следствия: на количество пар влияют (в порядке убывания степени

влияния) число уникальных простых множителей, затем – общее число простых множителей, затем – количество чисел-близнецов, которые связаны с конкретным x .

Изучая график (рис. 2), мы смогли обнаружить некоторую интересную зависимость:

$$x = 30n, n \in N \quad (1),$$

Возникновение такого большого числа пар Гольдбаха (последовательность показана на рисунке 3) у элементов этой зависимости обусловлено тем, что само число 30 раскладывается на простые множители 2, 3, 5 – т. е. уникальность его равна 3, а также с ним связаны две пары чисел близнецов (т.е. в разложении числа на сумму простых участвуют пары (11,13), (17,19)). Сами по себе числа-близнецы – простые числа, разность которых равна двум. Как правило, их пары, кроме первых двух, имеют вид $30n \pm 1$, либо $30n + 12 \pm 1$, либо $30n + 18 \pm 1, n \in Z, n \geq 0$.

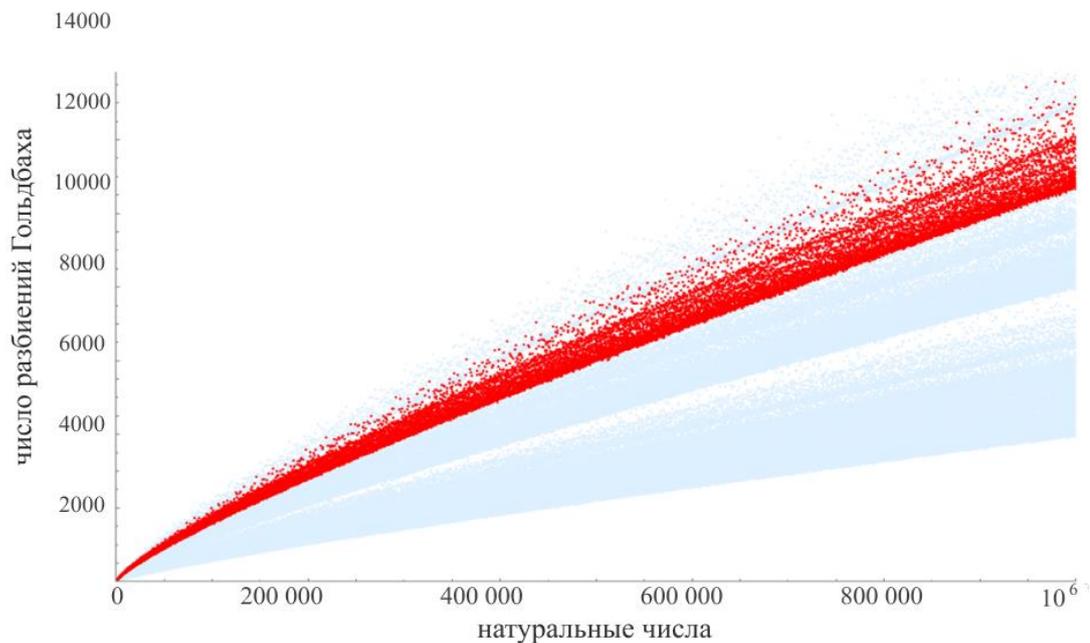


Рисунок 2 – распределение точек последовательности $x = 30n, n \in N$

Из имеющейся информации нам удалось получить три вывода: последовательности вида $x = kn, k = \{30, 42, 48\}, n \in N$ – будут иметь большее по сравнению с остальными последовательностями число пар; последовательность вида $x = (k - 2)n, k = \{30, 42, 48\}, n \in N$ – будет иметь среднее число пар ввиду свойств чисел-близнецов; последовательность вида $x = (k + 2)n, k = \{30, 42, 48\}, n \in N$ – будет иметь наименьшее число пар ввиду наличия только одного уникального простого множителя;

Примеры каждой из приведённых последовательностей изображены на рисунке 3.

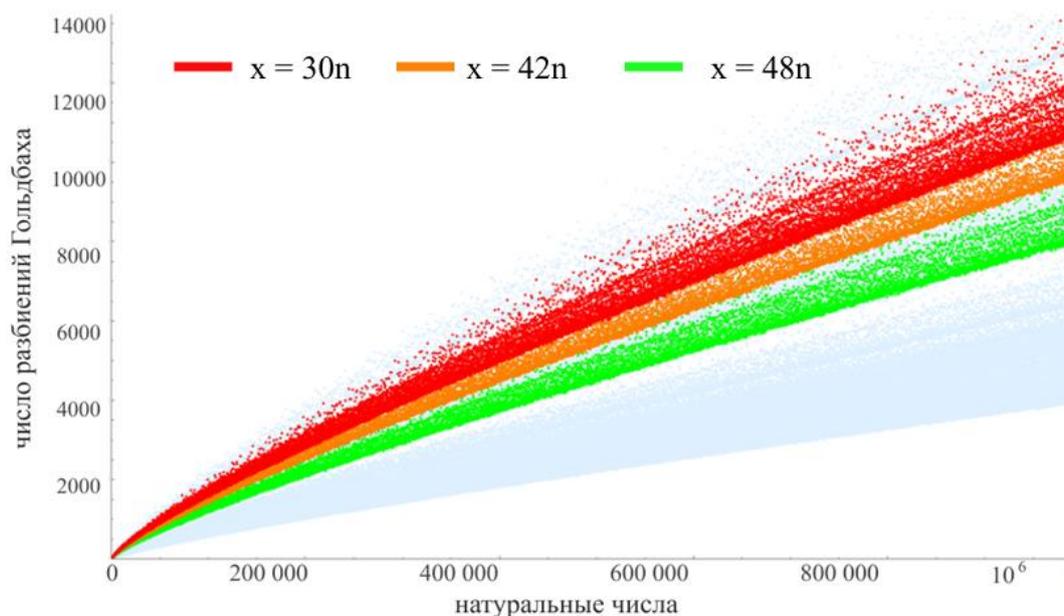


Рисунок 3 – Распределение высших точек графика

На графиках видно, что избранные последовательности чисел представляют собой несколько нечёткую линию. Это происходит ввиду того, что некоторые числа, принадлежащие данной последовательности, дают значительно большее число пар Гольдбаха. Данный феномен связан с тем, что при выборе именно простого n в последовательности $x = (k \pm 2)n$ (где x – число на оси абсцисс; k – чётное натуральное число, задающее кратность; $n \in N$) количество уникальных простых множителей данного x будет на единицу превышать аналогичный параметр для соседних значений x при $n \in N$, что и повышает число разложений Гольдбаха. На рисунке 4 представлены две последовательности, содержащие последовательности с такими множителями, которые обладают четырьмя и пятью уникальными простыми множителями (210 и 2310 соответственно).

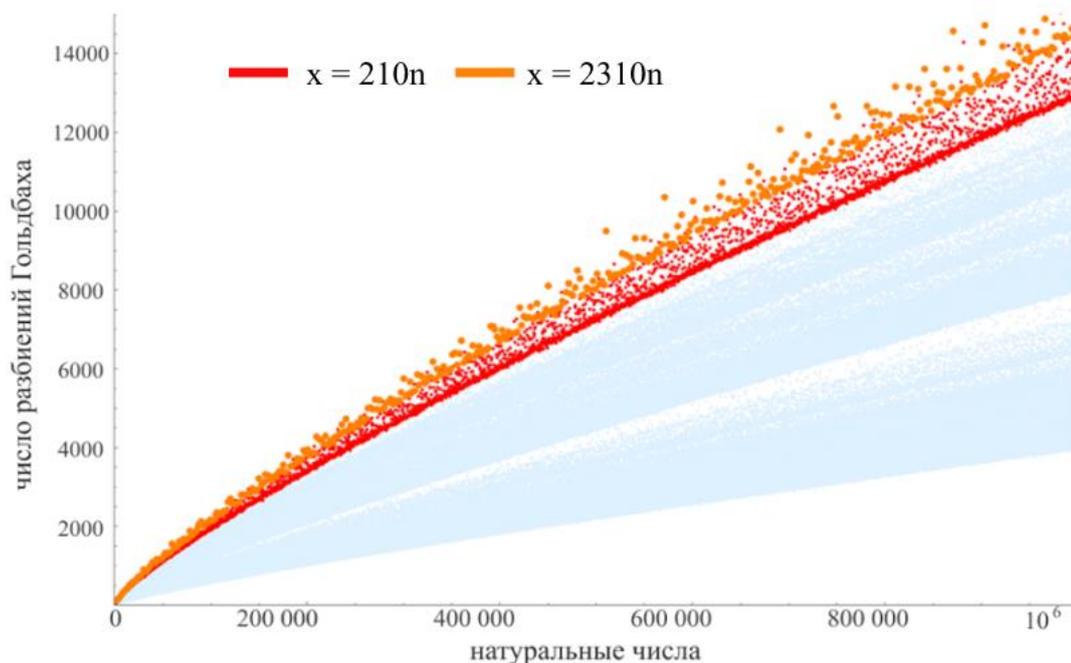


Рисунок 4 – Распределение высших точек графика

Здесь становится по-настоящему очевидно, как связано число уникальных простых множителей данного x с полученным числом пар разбиений Гольдбаха.

Таким образом, гипотеза Гольдбаха предполагает, что любое четное число больше двух может быть представлено в виде суммы двух простых чисел. Эта гипотеза может быть полезна для криптографии, так как разложение чисел на простые множители может быть сложным и

непредсказуемым. Кроме того, изучение количества разбиений Гольдбаха может быть полезно для создания новых криптографических алгоритмов и систем безопасности.

Список использованных источников:

1. Goldbach, C. (1742). Letter to Euler, dated June 7, 1742. In C. Goldbach, *Gesammelte Werke* (Vol. 3, pp. 41-44). Leipzig, Germany: Akademie Verlag, 1979.

2. Проблема Гольдбаха: определение, доказательства и решение [Электронный ресурс]. – Режим доступа: <https://fb.ru/article/452284/problema-goldbaha-opredelenie-dokazatelstva-i-reshenie>. – Дата доступа: 02.04.2023.

РАСПРЕДЕЛЕНИЕ ИГРОКОВ В ТУРНИРНОЙ СИСТЕМЕ

Лицкевич В.В.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Цегельник В.В. – д-р физ.-мат. наук, доцент

This paper aims to explore the use of AI-generated game art and its potential impact on game development, artistic creativity, and ethical considerations. Examples of integration of such technology in development process are described.

В турнирах сила соперников команды или игрока может оказать значительное влияние на их успех. Распределение игроков в турнирной системе может быть разным в зависимости от формата турнира и количества игроков. Рассмотрим распределение игроков в турнирной системе на однократное выбывание.

Посев в турнире – это процесс, при котором игроки или команды размещаются в определённом порядке перед началом турнира. Он может быть основан на рейтинге игроков, результатам предыдущих турниров, а также на других факторах, которые могут влиять на успех игроков. В некоторых случаях посев может быть произведен случайным образом, однако это означает, что в одной части турнирной таблицы могут быть сгруппировано много сильных игроков. Это приведёт к тому, что у сильных игроков будет мало шанса дойти до финала турнира, а слабым игрокам может повести и они дойдут до финала без каких-либо проблем. Это приводит не только к нечестному распределению мест, но и к не интересным для зрителей турнирам [1]. Проводя посев игроков на основе их предыдущих результатов или других объективных критериев, возможно гарантировать что лучшие игроки будут иметь справедливые шансы на победу. Для этого будем использовать метод попарного сравнения.

Допустим, есть n игроков для сравнения. Можно создать матрицу $n \times n$, где запись (i, j) представляет сравнение между игроком i и игроком j . Например, если игрок i сильнее, чем игрок j , вы можете присвоить значение 1 записи (i, j) и значение -1 записи (j, i) . Если элементы одинаково сильны, можно присвоить значение 0 обоим элементам.

Используя эти значения, можно создать матрицу парных сравнений S , где $S_{ij} = 1$, если элемент i сильнее элемента j , $S_{ij} = -1$, если элемент j сильнее элемента i , и $S_{ij} = 0$, если элементы одинаково сильны.

Попарное сравнение можно корректировать с учетом различных факторов, например, сравнивать более чем по одному критерию. Тогда в результате мы получим несколько матриц для каждого из сравнимых параметров. Матрица S будет суммой этих матриц, умноженных на коэффициент важности их сравнимого параметра. Чем важнее параметр, тем больше множитель.

Для ранжирования элементов можно использовать аналитический иерархический процесс, который предполагает вычисление собственного вектора матрицы S . Собственный вектор представляет относительную важность каждого элемента и может быть использован для ранжирования от наиболее важного к наименее важному. Теперь, необходимо составить турнирную таблицу, в которой распределены изначальные позиции игроков. Для этого можно использовать рекурсивный алгоритм. Сначала поставим размер таблицы на 1, и впишем только самого сильного игрока. Затем, увеличим размер таблицы вдвое, поставив предыдущих игроков на нечётные позиции, а новые – на чётные. Новых игроков будем добавлять с конца таблицы в начало. Так будем повторять, пока размер таблицы не превысит или не станет равным количеству наших игроков (Рисунок 1).

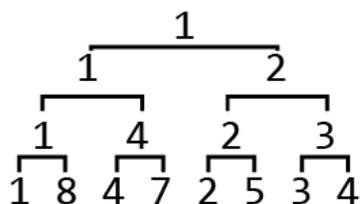


Рисунок 1 – Распределение игроков в турнире и вероятное развитие событий, с учётом силы игроков.

В результате получается турнирная система, в которой игрокам дан равный шанс, что является оптимальным для справедливого турнира.

Список использованных источников:

1. *What Is the Correct Way to Seed a Knockout Tournament?* [Электронный ресурс]. – Режим доступа: http://www.oxfordcroquet.com/tech/knockout3/What_is_the_Correct_Way_to_Seed_a_Knockout_Tournam.pdf. – Дата доступа: 12.03.2023.

ИССЛЕДОВАНИЕ ГИПОТЕЗЫ КОЛЛАТЦА: ОТ ПРОСТЫХ ЧИСЕЛ К СЛОЖНЫМ ПОСЛЕДОВАТЕЛЬНОСТЯМ

Убоженко М.А., Копыток М.Д.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Баркова Е.А. – канд. физ.-мат. наук

Гипотеза Коллатца - это интересная и загадочная математическая гипотеза, которая утверждает, что независимо от изначального числа, последовательность, полученная из него путем многократного применения определенных действий, всегда достигнет единицы. Эта статья рассказывает о разработке программы для анализа гипотезы Коллатца, а также о использовании данных, полученных от программы, для выявления возможных закономерностей в структуре последовательностей.

В данной работе представлено исследование гипотезы Коллатца, которая остается одной из наиболее загадочных проблем современной математики. Гипотеза Коллатца, также известная как проблема $3x+1$, была предложена в 1937 году немецким математиком Лотаром Коллатцем. Она заключается в следующем: начиная с любого положительного целого числа n , можно построить последовательность чисел, умножая это число на 3 и добавляя 1, если число нечетное, или операцию деления на 2, если число четное. Гипотеза заключается в том, что для любого начального значения n , такая последовательность рано или поздно достигнет единицы.

$$cltz(n) = \begin{cases} 3n + 1, & n \equiv 1 \pmod{2} \\ \frac{n}{2}, & n \equiv 0 \pmod{2} \end{cases} \quad (1),$$

где n – любое натуральное число.

Целью данного исследования является создание программы для анализа гипотезы Коллатца на натуральных числах, позволяющей проводить численные эксперименты и проверять гипотезу для различных начальных значений n , а также отметить связь гипотезы с явлениями из реального мира. Разработанная программа базируется на методах длинной арифметики и позволяет проводить анализ последовательности чисел с представлением данных в различной форме. Результаты исследования могут быть полезными для понимания природы гипотезы Коллатца и ее связи с другими областями математики.

Программа позволяет:

- Находить цепочки числа в виде «прямого хода» гипотезы Коллатца
- Построить график для «чисел-градин»
- Шифровать и дешифровать
- Построить столбчатые диаграммы для анализа частоты попадания цифр в числе
- Находить обратные цепочки числа в виде «обратного хода» гипотезы Коллатца

Нахождение цепочки числа в виде «прямого хода» гипотезы Коллатца. Чтобы добиться большей наглядности и расширить диапазон вводимых чисел, были введены элементы длинной арифметики. Для того чтобы алгоритм поиска числа по гипотезе Коллатца работал на больших числах, не входящих в диапазон целочисленных переменных, предоставляемых языком Delphi, а если быть точнее, то выше числа 9223372036854775807, был реализован метод длинной арифметики. Метод заключается в том, что программа разбивает введенное пользователем число на массив из цифр этого числа, затем расчеты ведутся по порядку над каждой цифрой. Таким образом длинная арифметика может работать с числами, содержащими до 256 цифр. По состоянию на апрель 2021 года проверены все натуральные числа до 9 789 690 303 392 599 179 036 включительно [1], и каждое из них продемонстрировало соответствие гипотезе Коллатца. Данная программная реализация позволяет проверить числа до 10^{200} . Математиками было выяснено, что если другой цикл всё-таки и существует, то он состоит как минимум из 168 миллиардов чисел. Результат программы для большого числа приведен на рисунке 1:

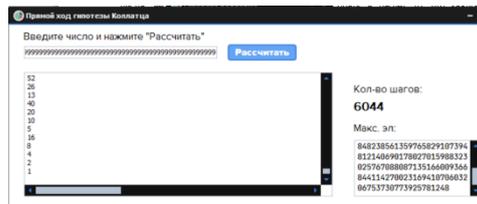


Рисунок 1 – Окно приложения для вывода всей цепочки чисел, количества шагов в ней, а также максимального числа, достигнутого в этой последовательности

Проанализировав гипотезу Коллатца, можно выделить несколько дополнительных гипотез:

- Первая: имеется равное количество чётных и нечётных чисел, так что в любой длинной последовательности вычислений чётные и нечётные N будут появляться одинаково часто. Нечётное N увеличивается второе (чуть больше), а чётное уменьшается всего лишь вдвое. Следовательно, по мере роста числа итераций значение N будет бесконечно возрастать. За каждую итерацию N будет в среднем увеличиваться на $(3N+1)/2$. Для больших значений N это, в сущности, равно $3N/2$.
- Вторая гипотеза основана на тезисе «Чем выше заберёшься, тем ниже скатишься». Справедливость этой мысли подкрепляется тем реальным фактом, что стоит в вычислениях получить точную степень двойки, как последовательность неудержимо скатится к 1. (При делении на 2 числа, представляющего собой 2 в любой степени, кроме самой двойки, постоянно будет получаться всё меньшее и меньшее чётное число.) Среди бесконечного счётного множества целых чисел существует бесконечно много степеней двойки, так что при достаточно долгом вычислении обязательно получится одна из них. В процессе вычислений можно получить сколь угодно большое число N , но крах его неизбежен.

Гистограмма чисел и как гипотеза Коллатца связана с населением стран. Введём такое понятие, как числа-градины. Числа-градины – распространённое название для совокупности рассмотренных последовательностей. Такое название возникло из-за того, что графики последовательностей похожи на траектории движения градин в атмосфере. Можно посмотреть на старший разряд чисел-градин. Например: Для 1 это будет 1. Для 2 это будет 1,2. Для 3 это будет 3, 10, 5, 16, 8, 4, 2, 1. А затем составить гистограмму для этих чисел. По мере того, как мы получаем всё больше данных, соотношение высоты столбиков гистограммы становится всё более упорядоченным. С помощью программы было подсчитаны числа, не превосходящие 1 миллиарда, гистограмма для них изображена на рисунке 2. Чем цифра больше, тем реже она оказывается впереди. Девятка, например, менее чем в 5 процентах случаев. Такой расклад характерен не только для чисел из последовательности. Примеров много: это и население стран, стоимость компаний, числа Фибоначчи. - такое распределение известно как закон Бенфорда [2]. Сам закон утверждает, что в таблицах чисел, основанных на данных источников из реальной жизни, цифра 1 на первом месте встречается гораздо чаще, чем все остальные (приблизительно в 30 % случаях), а также вероятность того, что цифра будет стоять на первом месте в числе тем больше, чем меньше цифра.

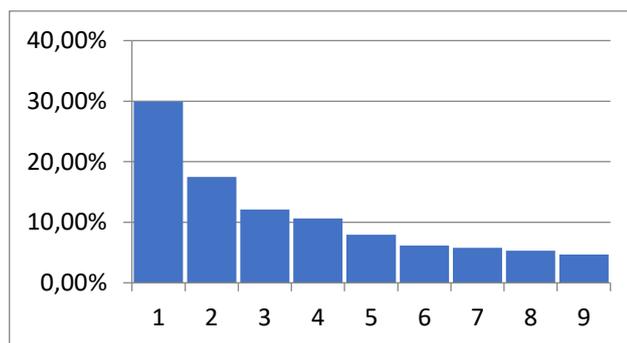


Рисунок 2 – Гистограмма частоты встречи той или иной цифры в числах последовательности в процентном соотношении

Графическое представление гипотезы Коллатца. Для дальнейших пояснений введем несколько новых понятий: Прямой ход – анализ гипотезы, при котором берётся какое-то число и анализируется его дальнейшее поведения до прихода в цикл 4-2-1. Обратный ход – представление гипотезы в обратном порядке, где например числу 10 предшествует либо число 20, которое поделится на 2, либо число 3, которое умножится на 3 и к результату прибавится единица. Отсюда следует вывод, что гипотеза Коллатца, представляет собой бесконечное бинарное дерево (или же ориентированный граф), которое с помощью нашей программы можно увидеть на рисунке 3.

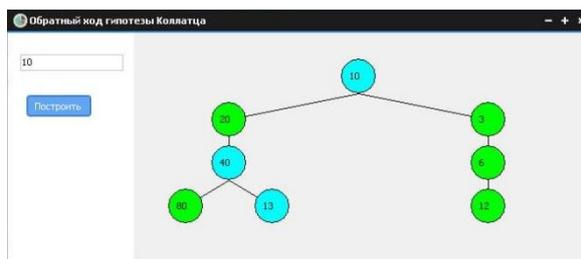


Рисунок 3 – Визуализация обратного хода гипотезы Коллатца для части дерева из последовательности по заданному числу

Проанализируем закономерности в поведении чисел-градин в прямом ходе гипотезы – все значения образуют циклы. Однако путь, который они проделывают, абсолютно случаен. Рассмотрим произвольный участок, например числа 27, изображённого на рисунке 4. В его колебаниях прослеживается сходство с рынком акций, и это не случайно – и то, и другое пример геометрического броуновского движения [3]. Но если в долгосрочной перспективе рынок акций растёт, то последовательность всегда приходит к единице. Формулы гипотезы Коллатца можно использовать для генерации случайных помех или кривых в программном обеспечении. Они просты в использовании, а результат их графиков случаен.

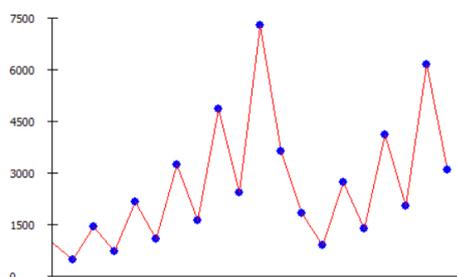


Рисунок 4 – Окно приложения с произвольным участком графика прямого хода гипотезы для числа 27.

Стоит отметить, что если что если кодировать ребро из графика прямого хода, построенное по формуле $N * 2$ как 0, а $\frac{N-1}{3}$ как 1, то можно переформулировать гипотезу для этого варианта – для любого числа существует и притом единственная цепочка действий, приводящая к этому числу. Используя это, можно зашифровать числа, а также расшифровывать число по его цепочкам. Для числа 12 шифр: 001010000, для 5: 10000. Нельзя найти закономерности в постановке единиц и нулей, как и поведение гипотезы – они случайны.

Список использованных источников:

1. Гипотеза Коллатца [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Гипотеза_Коллатца – Дата доступа: 03.04.2023.
2. ЗАКОН БЕНФОРДА: СУЩНОСТЬ И ПРИМЕНЕНИЕ [Электронный ресурс]. – Режим доступа: <https://top-technologies.ru/ru/article/view?id=31987> – Дата доступа: 03.04.2023.
3. Геометрическое броуновское движение [Электронный ресурс]. – Режим доступа: https://bstudy.net/730233/ekonomika/geometricheskoe_brounovskoe_dvizhenie – Дата доступа: 03.04.2023.

СЕКЦИЯ 7
«ФИЛОСОФИЯ, ЛОГИКА И ЖИЗНЬ»

ТРАНСФОРМАЦИЯ СТАНДАРТОВ КРАСОТЫ В СОВРЕМЕННОСТИ

Алексеева К.А., студент гр.151003

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Амоненко С.А. – м. ф.н.

Аннотация. Тема трансформации стандартов красоты в наше время сложна и многогранна. Хотя эволюционные факторы, возможно, сыграли определенную роль в формировании стандартов красоты, общество также оказало на этот процесс значительное влияние. Трансформация стандартов красоты – это непрерывный процесс, который отражает меняющиеся ценности и убеждения общества, что подтверждается тем, что представления о красоте и предпочитаемых типах телосложения сильно различаются в разных культурах на протяжении всей истории. Развитие социальных сетей стало ключевым фактором возникновения новой одержимости недостижимыми стандартами красоты, что привело к переоценке традиционных представлений о красоте в сторону большей инклюзивности и многообразия форм проявления красоты.

Ключевые слова: красота, стандарты красоты, трансформация восприятия красоты, инклюзивность.

Бросив взгляд в зеркало перед уходом из дома, мы часто задаемся вопросом: «Хорошо ли я выгляжу?». Казалось бы, в этом нет ничего парадоксального. Однако немногие пытаются понять, в чем же причина возникновения таких мыслей и как они влияют на наше качество жизни.

Внешность в первую очередь связана с концепцией красоты, которую непросто обусловить. Красота исходит изнутри – и это действительно так, ведь человек может быть весьма обаятельным благодаря своему яркому характеру. Однако люди, что утверждают, что им нет дела до их внешности, обычно являются сторонниками противоположной точки зрения. Часто звучат слова, что «красота в глазах смотрящего». Обозначить красоту становится все сложнее, когда мы разбираемся во всевозможных ее проявлениях в соответствии с различными обществами и верованиями.

Основанием для появления размышлений о внешности и ее привлекательности является то, что в нашем обществе установлены определенные стандарты красоты. Эти, так называемые, эталоны влияют на наше восприятие и ожидания относительно внешнего вида самих себя и других людей.

Изучение трансформации стандартов красоты является одной из актуальнейших тем для нашего общества в силу некоторых причин:

1) Такое исследование позволит выявить, какие факторы определяют наши предпочтения и вкусы, и как они меняются со временем.

2) Стандарты красоты часто являются объектом дискуссии и критики в современном обществе. Многие считают, что каноны красоты являются неестественными и приводят к искажению восприятия собственного тела. Изучение преобразования этих эталонов позволяет лучше понять их социальную и историческую обусловленность, формирует у людей чувство ответственности за свое здоровье и понимание своей ценности как личности.

3) Эталоны красоты могут быть связаны с дискриминацией на основе расы, пола, возраста и других факторов. Важно обнаруживать и преодолевать такие формы дискриминации, поэтому изучение данной темы способствует развитию более толерантного и равноправного общества.

Идеалы красоты менялись на протяжении всей истории человечества: в разные эпохи возникали свои видения эталона. Тем не менее, определенные каноны красоты существовали веками и преобразовывались в очень медленном темпе. Для того чтобы понять причинно-следственную связь появления современных стандартов, необходимо заглянуть в прошлое.

В древней Греции красота не рассматривалась как самостоятельное явление. Считается, что у греков не существовало автономной эстетики или теории, связанной с понятием красоты. Ее совмещали с такими понятиями как благо и умеренность. В то же время, греками воспевались дивные красоты женских божеств. Одним из ярких примеров является статуя богини Афродиты, сочетающей в себе их видение идеала, одним словом – гармоничность: она имела средний по нашим меркам рост – 164 см, объемы бедер, груди и талии – 93, 86 и 69 см. Также именно в этот период появляется довольно необычное, дионисийское понятие красоты: вселяющей-ли ужас, то-ли восторг, выражающее собой буйство, избыточность, уничтожение какой-либо индивидуальности, что можно наблюдать и в XXI веке.

Особое значение приобретает понятие гармонии. С развитием математических наук в античности становились популярными симметричность и пропорциональность. Однако гармония также предполагала равновесие противоположных черт. Получается, что стандарт становился почти недостижимым идеалом. Такая красота радует глаз, но в наше время ее можно достичь лишь путем пластических операций. Можно ли тогда довериться такому стандарту и следовать ему?

Средние века зачастую приписываются темным временам, что связано с их мрачными ассоциациями. Однако это распространенное заблуждение. В работах известных художников, изображающих прекрасных дам, мы можем наблюдать, как свет олицетворял одаренность и изобилие. Женщины на картинах излучают его, искусно играя с тенями. Желтый цвет, ставший символом богатства, использовался художниками для изображения света. Он символизировал не только богатство, но и социальное неравенство, поскольку только те, кто обладал властью, могли быть признаны "одаренными светом". Таким образом, идея света как красоты проявляется и сейчас: один из устоявшихся стандартов красоты включает в себя наличие светлых волос и бледного цвета кожи.

В эпоху Возрождения люди пытаются перейти от критериев исчислимости и меры к поиску нового, воображаемого. Стандарты красоты наконец стали считаться субъективными: они зависят от мнения общества, а не посредством некоего правила. Начиная с эпохи Возрождения, распространяется идея, согласно которой идеалы красоты задаются каждым из нас индивидуально.

В эстетике XVIII века субъективизм получает еще большее распространение. Так как нельзя однозначно выделить критерии для оценки красоты, то одна и та же вещь может показаться красивой одному и безобразной другому.

В Викторианскую эпоху на смену воображаемому приходит практичная и однозначная красота. Эра буржуазии толкует жизнь с практической точки зрения: каждой вещи определяется своя цена, поэтому красоту уже невозможно рассматривать, не принимая во внимание ее показательную стоимость.

Наступает XX век. Происходит частичный переход от практичности к функциональности. Она характеризуется массовостью производства вещей. Новое понятие красоты характеризуется недолговечностью и в то же время воспроизводимостью. Такое описание является многозначным: те же характеристики относятся и к внешности.

По мнению итальянского философа Умберто Эко, первая половина XX века «была ареной драматической борьбы между Красотой провокации и Красотой потребления» [1, с. 414].

Первая хочет вызвать у зрителя эмоциональный отклик, заставить задуматься, подвергнуть сомнению привычные установки. Сегодня красота стремится быть провокационной, вызывать отвращение или восхищение, но никак не оставлять равнодушным. И хотя такая красота не всем нравится, она продолжает эволюционировать и выражать дух времени.

Вторая же заключается в том, что мы можем наслаждаться товарами и услугами, которые предлагает современный мир. Потребление также помогает нам проявлять свою индивидуальность, выделяться из общей массы и выражать себя через вещи, которые мы покупаем или используем. Важно понимать, что красота потребления не заключается только в самом акте покупки, но и в том, как мы используем приобретенные вещи и как они влияют на нашу жизнь и окружающих нас людей.

Стартовой точкой тенденции, известной сегодня как «одержимость весом», стали 20-ые годы. Это период, когда были изобретены весы и зеркала в полный рост. Женщины стали более сознательно относиться к тому, как они выглядят, поскольку могли видеть все свои недостатки. В это же время возникли первые попытки дискриминации. Все они несли один посыл: чем больше вы постараетесь над собой и своей внешностью, тем красивее сможете стать. Это положило начало стремительному преобразованию эталонов красоты в современном мире.

Вторая половина XX века предзнаменовала смену стандартов с частотой в 20-30 лет. Сравнивая с прошлым, можно сделать вывод, что в современности преобразование происходит намного чаще.

Существует множество факторов, влияющих на представление человеком современного стандарта красоты.

С развитием интернета ускорилось и преобразование устоявшихся эталонов красоты. Многообразие информации, представленной в сети, включая советы и рекомендации по уходу за собой, диетические режимы, фитнес упражнения и модные тенденции, играет важную роль в формировании представлений о красоте.

С помощью интернет-форумов можно запросто дискутировать о внешности, высказывать свое мнение, а также найти людей с подобными интересами. Онлайн-пространство стало средством социализации с миллионами людей по всему миру, которое определяет новые тенденции.

Средства массовой информации часто устанавливают и укрепляют стандарты красоты с помощью рекламы и фильмов. Под этим влиянием многие начинают верить, что изображенные там люди имеют идеальный тип телосложения. СМИ преподносят определенные стандарты красоты как норму, оказывая давление в первую очередь на молодежь.

Сегодня наиболее мощное влияние на понимание красоты оказывают сайты знакомств и социальные сети. Instagram, YouTube и TikTok - популярные платформы, которые пропагандируют определенные стандарты красоты и давят на людей, заставляя их соответствовать им.

На самом деле, наше восприятие внешности очень динамично: оно меняется с каждым новым человеком. Стандарт красоты — это видимость. Красивым мы обычно считаем тот тип внешности, на который чаще обращаем внимание. Это объясняется работой мозга.

В основе лежит такое явление, как последовательная зависимость[2] – это эффект, при котором непосредственно предшествующее испытание влияет на реакцию участников на текущий стимул. Недавний опыт восприятия и моторная память влияют на поведение человека. Чтобы произошло последовательное смещение, мозг должен сохранять информацию в течение времени между событиями, чтобы смешать прошлую и текущую информацию. В этом есть и положительный эффект: с каждым новым человеком наше восприятие становится точнее и увереннее.

В современном мире внешность, как и раньше, есть социальный маркер. Самодостаточность определяет возможности и доступность к различным ресурсам, таким как косметика, парфюмерия и другое. Некоторые люди могут позволить себе продукты или процедуры, улучшающие их внешность, что приводит к повышению стандартов красоты.

Открытия в области медицины и технологий привели к созданию косметических процедур, которые могут изменить чью-либо внешность. Эти достижения оказывают влияние на стандарты красоты, поскольку люди стремятся выглядеть более "идеально" при помощи данных воздействий, не исключая хирургических вмешательств.

Разнообразие и инклюзивность (репрезентация людей с ограниченными возможностями или людей, не соответствующих гендерному признаку) являются важными факторами трансформации стандартов красоты. По мере того, как общество становится все более неоднородным, растет понимание того, что красота проявляется в самых разных формах. Это привело к смещению от традиционных стандартов красоты, которые часто основывались на узких, евроцентричных идеалах, в сторону более всеобъемлющего и разнообразного понимания красоты.

Именно эти факторы оказали огромное влияние на скорость трансформации стандартов красоты в современности.

Есть интерес отдельно рассмотреть события, которые оказывают прямое влияние на представление эталона красоты. Таким примером может стать конкурс красоты Мисс Вселенная, который проводится ежегодно с 1952 года[2], производящий значительное воздействие на стандарты красоты во всем мире. Конкурс пропагандирует определенный идеал красоты, который обычно включает в себя высокую, стройную фигуру, чистую кожу и симметричные черты лица.

Хотя конкурс "Мисс Вселенная" подвергался критике за пропаганду узких и нереалистичных стандартов красоты, он также сыграл определенную роль в некой их трансформации. Поощряя разнообразие и культурные ценности, а также предоставляя женщинам возможность использовать свою платформу для отстаивания своих целей, конкурс помог расширить наше понимание красоты и бросить вызов устоявшимся стандартам красоты.

Например, в последние годы конкурс был направлен на то, чтобы сделать его более инклюзивным, позволив участвовать трансгендерным женщинам и привлекая к участию более разносторонних участниц. Кроме того, многие победительницы конкурса "Мисс Вселенная" использовали платформу для пропаганды важных идей и содействовали более широкому пониманию красоты. Например, Мисс Вселенная 2019, ЗоэбиниТунзи, использовала свою платформу для продвижения натуральных волос и оспаривания устоявшихся нездоровых стандартов красоты.

Трансформация стандартов красоты имеет значительное влияние на нашу культуру и общество. Переход от узкого и стандартизированного представления о красоте к более широкой и инклюзивной концепции красоты имеет ряд положительных последствий.

Развитие образа здорового тела является одним из главных последствий трансформации стандартов красоты. Новые стандарты красоты ставят акцент не только на внешний вид, но и на образ здорового тела, здоровое питание, спорт и физическую активность.

Рост любви к себе и принятие своего тела также является важным фактором трансформации эталонов красоты. Новые каноны ставят акцент на то, что каждый человек уникален и прекрасен в своей индивидуальности. Это способствует улучшению самооценки и увеличению уверенности в себе.

Самый насущный вопрос, стоящий на повестке дня перед человеком в XXI веке: «Как мне выделиться из толпы?». Данный тренд является основным фактором переосмысления стандартов красоты, появления новых эталонов и главной причиной, отказа в современном мире от одного устоявшегося стандарта, ведь сейчас ценится оригинальность и креативность. Люди начинают принимать себя, перестают зависеть от навязываемых образов. Совершенствоваться, быть не такими, как другие – вот главный тренд, задающий стандарты красоты сегодня. В современном обществе в понимании стандартов красоты все больше распространяются элементы и веяния.

Татуировки становятся все более популярными в XXI веке, и в настоящее время многие считают их элементами новых стандартов красоты. Их можно рассматривать как форму самовыражения и способ улучшить свою внешность. Они также могут быть использованы в память о важных событиях, людях или верованиях. Действительно, тату стали более приемлемыми в

обществе, и многие люди теперь рассматривают их как форму искусства. Они больше не рассматриваются как символ бунта или отклонений, какими они, возможно, были в прошлом. Также некоторые люди делают татуировки специально для того, чтобы подчеркнуть свою красоту или скрыть предполагаемые недостатки. Например, человек может выбрать татуировку на груди или на лице, чтобы скрыть шрам.

В последние годы андрогинная внешность приобрела популярность в качестве одного из стандартов красоты, бросая вызов традиционным представлениям о гендере и красоте. Андрогинность относится к сочетанию типично мужских и женских характеристик, в результате чего создается образ, который нельзя однозначно классифицировать как мужской или женский. Ее можно рассматривать как форму гендерного самовыражения, которая бросает вызов традиционным гендерным ролям и ожиданиям. Андрогинность также позволяет людям выражать себя новыми творческими способами, бросая вызов традиционным стандартам красоты и способствуя инклюзивности.

Одним из новых явлений в современном мире является феминистическое движение за бодипозитив. Оно появилось как ответвление движения за принятие жира и превратилось в самостоятельную силу. Феминизм второй волны принес с собой принятие всех размеров тела в знак протеста против нереалистичных стандартов телосложения. Движение за принятие жира было посвящено уничтожению стереотипов о жирности и изменению убеждений в том, что жир является первопричиной плохого здоровья.

Сегодня социальные сети поощряют девушек становиться подтянутыми и худощавыми. Фигура в виде песочных часов уже много лет является популярным стандартом красоты. Ее можно описать как форму тела с узкой талией и более широкими бедрами и бюстом, создающую соблазнительный силуэт, который часто ассоциируется с женственностью и привлекательностью. Она заставляет многих женщин мечтать о том, чтобы иметь такие пропорции. Но это не всегда реально, поскольку каждый организм индивидуален и многие аспекты влияют на формирование фигуры. Тоже касается и концепции «благородного лица» как стандарта красоты, восходящего к древним грекам, которые верили, что физическая красота является признаком добродетели. В наше время представление о данном стандарте часто ассоциируется с сильным, симметричным лицом, которое излучает уверенность и грацию. Черты, которые обычно ассоциируются с «благородным лицом», включают высокие скулы, четко очерченную линию подбородка и прямой нос. Они часто воспринимаются как привлекательные, поскольку считается, что они связаны с хорошим здоровьем и наследственной физической формой.

Современный мир красоты находится в постоянном движении, наподобие маятника, мотаясь от одного стандарта к другому. В круговом движении повторяются модные тенденции, и через их призму мы строим свое понимание красоты. Инклюзивная красота стала ярким примером того, что все больше людей желают увидеть отражение себя в рекламе и на подиуме, не уделяя такого внимания физическим параметрам и характеристикам моделей.

Новые стандарты красоты уделяют внимание не только внешности, но и духовным факторам, что, в частности, связано с развитием феминистического движения. Теперь, чтобы соответствовать им, девушкам не нужно тратить огромные суммы на пластические операции. Они могут достичь новых высот, зная и уважая свою природную красоту. Важно отметить, что изменились не столько сами стандарты красоты, как образ их восприятия и мышление людей.

Сегодня нет единой формулы красоты, и это замечательно. Все чаще мы признаем, что в поиске красоты ключевым является наше отношение к собственной индивидуальности. Вместо того чтобы стремиться к идеалу, мы можем просто принять себя такими, какие мы есть, уже сегодня.

Список использованных источников:

1. Эко У. *История Красоты* / У. Эко. – М.: СЛОВО/SLOVO, 2017. – 440 с.
2. Куц А. К., Максименко В. А., Храмов А. Е. Влияние «сенсорной предыстории» на обработку головным мозгом неоднозначных визуальных стимулов / А. К. Куц, В. А. Максименко, А. Е. Храмов. – *Известия высших учебных заведений. Прикладная нелинейная динамика* – Том 30, N1. – 2022 – стр. 57-75.

НЕОФАШИСТКИЕ ИДЕИ В МОЛОДЕЖНОЙ СРЕДЕ: ОСНОВНЫЕ ФОРМЫ И ПРИЧИНЫ РАСПРОСТРАНЕНИЯ

Аксёненко М.А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шкундич А.О. – преподаватель

Аннотация. Автор статьи анализирует причины распространения идеологии неонацизма для молодёжи в современном мире. Актуальность исследования обусловлена противоречивыми тенденциями развития современной цивилизации. Выявлены факторы и закономерности формирования неонацизма как идеологии и политической практики. Показаны проявления неонацизма в современных политических процессах. Выявляются признаки и основные характеристики идеологии неонацистских объединений.

Ключевые слова: Неонацизм, идеология, безопасность, молодёжь.

Неонацизм – это понятие, объединяющее современные праворадикальные движения, которые в политическом и идейном отношении являются преемниками фашистских организаций, распущенных после Второй мировой войны 1939-1945 гг. Стремясь нередко отгородиться в обстановке неблагоприятного для [фашизма](#) соотношения сил от скомпрометировавших себя фашистских движений прошлого, неонацисты выявляют на практике тем не менее свою глубокую идейно-политическую связь с формами фашизма, сложившимися между двумя мировыми войнами. Важнейшие отличительные черты всех неонацистских политических течений и организаций — антикоммунизм, антисоветизм, крайний национализм, расизм (открытый или более, или менее прикрытый), критика с ультраправых позиций либеральных правительств (даже самых консервативных), действующих в рамках демократической парламентской системы; применение насильственных, террористических методов политической борьбы [2].

Одним из основных признаков неонацизма в молодежной среде является попытка создания сильного, единого и дискриминационного государства, которое диктуется принципом национального единства. Они признают единственную национальную культуру, язык и традиции, которые должны быть обязательными для всех граждан. Стоит отметить, что отличия неонацизма от первоначального фашизма заключается в основном в радикализации методов первого, сохраняя общий посыл и основные идеи. Привлекательность неонацизма, как и первоначального фашизма, для современного человека можно объяснить следующими признаками:

1. На фоне экономических и социально-политических кризисов неонацизм обещает навести порядок и обеспечить социальное благополучие. Не всех, но многих — и это социальное благополучие «своих». При известных в истории фашистских режимах временно преодолевался кризис экономики, а социальное положение населения улучшалось. Можно было бы неонацизм назвать в этом отношении социально ориентированным учением. Но это социальное благополучие избранных достигалось за счет подавления и эксплуатации как части своего народа, так и других наций.

2. Неонацизм апеллирует к великодержавию. В рамках фашистского проекта человек ощущает сопричастность великим свершениям. Неонацизм призывает к борьбе с внешними врагами. Он пользуется в этом отношении имманентным, существующим у каждой нации, запросом на имперскость и патриотизм.

3. Неонацизм в противовес индивидуализирующему либерализму содержит солидаризационные послы. Люди объединяются в корпорации, корпорации в государство. Акцентируется ощущение, что человек не одинок, что он социален.

4. Неонацизм восстанавливает утрачиваемые и размываемые при либерализме групповые идентичности, и прежде всего идентичности этнические. Этничность не является химерой или псевдоконструктом, как иногда полагают. Она реальность, игнорирование которой вызывает запрос на ее воспроизводство, что и используется неонацизмом.

5. Неонацизм апеллирует к истории, к традициям соответствующей общности, ее традиционным ценностям. Он обращается к архетипам коллективной памяти — и это дает свой отклик.

6. Неонацизм сбрасывает маски с буржуазного общества. Он резко осуждает гедонизм, потребительство, торгашескую мораль.

7. Неофашизм отвергает ценностный релятивизм. Его ответом является жесткая ценностная поляризация. Гей-пропаганде и шоу-культуре фашизм противопоставляет традиционное половое разделение социальных ролей и культуру высокого стиля.

8. Неофашизм разоблачает нефункциональность институтов демократии в либеральном обществе, их имитационно-симулятивный характер. Неофашистская критика состоит в разоблачении того факта, что в реальности вместо демократии при либерализме реализуется де-факто власть олигархии. Неофашизм противопоставляет псевдодемократизму иерархизм, как единственно жизнеспособную, с точки зрения неофашистских идеологов, систему.

9. Неофашизм разграничивает лучших и худших. Им утверждается принцип, что властвовать должны лучшие. Мыслитель должен мыслить, воин воевать, купец заниматься торговлей — и не смешивать свои социальные функции. Недовольство части общества в отношении существующих принципов элитной кооптации также ведет к поддержке фашистского движения.

10. Неофашизм, мобилизуя социум, активно оперирует образом врага внутреннего и внешнего. Враги ведь существуют объективно и замалчивание этого факта, например, либералами-космополитами толкает народ в объятия неофашизма. Тема борьбы с врагом почти всегда имеют поддержку, привлекает под фашистские знамена пассионариев. Оперирование образом врага получает особый отклик в контексте национального унижения от поражений, кризисов.

11. Неофашизм преподносит в образах высокой героики. В этом отношении он противопоставляется миру «обывателей». Пропагандируемые фашистами героические образы резонируют прежде всего в молодежной среде.

12. Неофашизм реализует запрос коренных жителей на ограничение в правах чужаков-мигрантов.

13. Неофашизм выступает за регулирование экономики, а соответственно, борется с безработицей, спекулятивным капиталом. Он оппонирует либеральному пониманию государства как ночного сторожа, стихии рынка и, с определенными оговорками, самой классической системе капитализма [1, с. 41].

Так, молодёжь, которая ещё не определилась с целями и ориентирами в жизни, не имея необходимого опыта «подкупается» подобными популистскими идеями, выступающими, грубо говоря, «за всё хорошее, против всего плохого». Однако, это «хорошее» получают только привилегированные классы методами дискриминации и эксплуатации большинства.

Неофашистские организации, которые бывают легальными, часто нарушают законы, осуществляют пропаганду ненависти и дискриминацию на основе расы, гендера, сексуальной ориентации и религии. Члены неофашистских организаций часто выступают с агрессивными демонстрациями, нападают на других людей и часто используют насилие как средство решения конфликтов.

На данном этапе мировой истории, существует тенденция нарастания популярности неофашистских партий и движений. С конца Второй мировой войны сменилось уже несколько поколений, поэтому в странах Запада историческая память всё сильнее отходит на второй план. Так, существуют партии, организации и известные политики открыто заявляющие о своей принадлежности к идеям неофашизма:

1. В США существует «[Партия прав национальных штатов](#)», основанная в 1958 году Эдвардом Ридом Филдсом, «[Американская нацистская партия](#)», основанная [Джорджем Линкольном Рокуэллом](#) который был основной фигурой в национал-социалистическом движении в послевоенной Америке и именовался «американским Гитлером». С 1962 ряд легально осуществляющих свою деятельность неонацистских движений и организаций объединены во «[Всемирный союз национал-социалистов](#)». К числу легально действующих неонацистских организаций относится «Национал-социалистическое движение 88 (НСД88)», «Национальный альянс», «Белые националисты Запада», «Белая революция», «Рыцари Белой Камелии [ККК](#)», «[Combat 18](#)». Также в 1978 году был создан [Институт пересмотра истории, отрицающий Холокост](#) и связанный с неофашизмом. По состоянию на 2020 год в США действовало около 500 ультраправых групп расистской и неофашистской направленности [4].

2. Альтернатива для Германии — ультраправая политическая партия в Германии. Основами идеологии являются национализм, евроскептицизм, национал-консерватизм, исламофобия, антикоммунизм и антифеминизм [4].

3. Северная Лига за независимость Падании — итальянская правая политическая партия, выступающая за паданский национализм, национал-консерватизм, регионализм, федерализм, евроскептицизм, антиглобализм [4].

4. Британская национальная партия — британская праворадикальная политическая партия. Выступает за «Третий путь», расизм, антисемитизм, исламофобия [4].

5. Новая сила — итальянская ультраправая политическая партия.

6. Йоббик — правая националистическая политическая партия Венгрии [4].

7. Национальное объединение — ультраправая консервативная националистическая политическая партия во Франции. Партия имеет 89 депутатских мандатов из 577 в нижней палате французского парламента [4].

Следует отметить, что представители некоторых из приведённых в пример партий состоят в Европейском парламенте. Это наглядно показывает, что тенденция роста популярности неонацистских идей характерна не только для молодёжи, но и для политиков, которые должны быть своеобразным примером для подражания подрастающему поколению, поэтому косвенно данная популярность также склоняет молодых людей к соответствующей идеологии. Так, не имея выраженного политического представительства, молодёжь становится на путь неонацистских субкультур и движений:

1. НС-Скинхэды — молодёжная ультраправая субкультура, представители которой придерживаются неонацистской идеологии. Внешними особенностями являются характерная одежда, а именно бомбер, синие джинсы, клетчатая рубашка, белые подтяжки, тяжёлые высокие кожаные ботинки и бритая голова, татуировки с нацистской символикой. Чаще всего, действия данной субкультуры носят крайне экстремистский характер [5].

2. Футбольные фанаты — субкультура, оправдывающая свои действия пристрастием к футболу, чаще всего представителям данного движения характерны идеи национализма, шовинизма и расизма. Действия нельзя назвать экстремистскими, но они противозаконны и грубо нарушают общественный порядок через насилие. Зачастую многие НС-скинхэды и футбольные фанаты принимают участие в деятельности тех или иных неонацистских политических и военных организациях [4].

В настоящее время, наиболее активно неонацистские движения развиваются в Украине и Польше. В Украине с 2014 года сформировалось множество военизированных организаций и батальонов, например, «Азов», «Правый сектор», «УНА – УНСО», «Белый молот», «Сич». Относящие себя к ультраправой идеологии люди не раз были замечены на высоких государственных постах. В Польше основным признаком неонацизма является риторика и действия властей. Как и в Украине, происходит полная декоммунизация, сносятся памятники, относящиеся к советскому периоду, переосмысливается история. В период кризиса с беженцами из Ближнего Востока и стран Северной Африки, Польша наотрез отказалась принимать мигрантов, делая акцент на этнических, культурных и религиозных различиях». А 11 ноября 2017 г., в День независимости Польши, в Варшаве прошёл самый крупный в Европе за все послевоенные годы марш ультраправых. Они шли в балаклавах и с прочей неонацистской атрибутикой - от факелов до лозунгов: «Будет белая Европа братских наций!», «Не радужная, а национальная Польша» и «Молимся за исламский холокост».

Основными причинами появления и распространения неонацистских идей в обществе являются:

1. Экономические кризисы, обостряющие классовые противоречия общества и разогревающие протестные настроения большинства. Однако экономические кризисы, взятые сами по себе, совершенно необязательно являются одним из условий, благоприятствующим установлению фашистского режима; они могут стать таким условием только в совокупности с другими факторами, а потому как за началом экономического кризиса не всегда следует утверждение фашизма у власти, так и победа фашистов может состояться и без экономического кризиса [3].

2. Социально-психологическое отчуждение, усиливающееся в эпоху монополистического капитализма и особенно обостряющееся во времена экономических кризисов, характеризуется атомизацией индивидов, страхом за собственное будущее и будущее своих близких, неуверенностью в завтрашнем дне и мучительным ощущением перманентной внутренней тревоги. Выходом из такого психологически-подавленного состояния отчужденного существования, становящимся особенно невыносимым в кризисные времена, может стать обретение иллюзорной «общности» путем растворения в «нации», «родине», а также чувства уверенности в будущем, обеспечиваемом «сильной властью», дарующей «стабильность» и «порядок». Очевидно, что подобная социально-психологическая обстановка создает питательную почву для утверждения фашистского режима и его легитимации в глазах масс [3].

3. Отсутствие или неразвитость демократических традиций. Данный фактор не следует переоценивать, так, как олигархия той или иной страны с наиболее глубоко развитыми либерально-демократическими порядками, пойдёт на установление фашистского режима, если соответствующим образом сложатся общественные обстоятельства. Тем не менее, устойчивые буржуазно-демократические институты и традиции политического управления дают монополистам больше простора для лавирования в ситуации, когда рост протестных настроений начинает приближаться к опасному рубежу [3].

Таким образом, неонацизм в молодёжной среде представляет серьёзную угрозу для нашего общества. Необходимы обширные образовательные и воспитательные программы, направленные

на формирование толерантности и уважения к правам и свободам каждого гражданина. Также необходимо использовать правовые механизмы, чтобы предотвратить распространение неонацистской пропаганды и насилия, обеспечить безопасность и содействовать созданию правильного политического контекста, где не будет места для такого вида преступности.

Список использованных источников:

1. Багдасарян, В. Э. Современный фашизм: облики и проявления / В. Э. Багдасарян, С. С. Сулашкин. – М: Наука и политика, 2017. – 328 с.
2. Большая советская энциклопедия [Электронный ресурс]. – Режим доступа: <https://gufo.me/dict/bse/>. – Дата доступа: 07.04.2023.
3. Хомутский, В. Ренессанс фашизма в Европе / В. Хомутский [Электронный ресурс]. – Режим доступа: https://historicaldis.ru/blog/43007293494/«Renessans-fashizma»-v-Evrope?utm_referrer=mirtesen.ru#42167486099. – Дата доступа: 07.04.2023.
4. Праворадикальные группировки и политические партии [Электронный ресурс]. – Режим доступа: https://www.ru.civic-nation.org/italiya/obshchestvo/pravoradikalnye_gruppirovki_i_politicheskie_partii/. – Дата доступа: 07.04.2023.

ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ ФОРМИРОВАНИЯ ИНФОРМАЦИОННОГО ОБЩЕСТВА В РЕСПУБЛИКЕ БЕЛАРУСЬ

Бейнар А.В., студент гр.150502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Миськевич В.А. – канд. филос. наук, доцент кафедры философии БГУИР

Аннотация. В работе рассматриваются основные проблемы и перспективы формирования информационного общества в Республике Беларусь. Обращается внимание на задачи технологического прогресса Беларуси. Рассматривается фундаментальный характер преобразований во всех сферах жизни человека, связанных с информационно-коммуникационными технологиями и цифровизацией.

Ключевые слова. Информационное общество, общество знаний, модель белорусской экономики, цифровизация, ИКТ, новые технологии.

В условиях развития информационного общества в Республике Беларусь, когда знания и информация приобретают доминирующую роль во всех сферах жизнедеятельности, необходимо правильно понимать проблемы и перспективы этого современного этапа цивилизации. Информационное общество — это дальнейшая эволюция постиндустриального общества, в котором большинство работающих занято производством, хранением, переработкой и реализацией информации, особенно высшей её формы — знаний. В таком обществе стандарты жизни, формы труда и отдыха, система образования и рынок находятся под значительным влиянием достижений в сфере информации и знания. Основная идеологическая концепция нового этапа выражается в избытке и легкой доступности информации. Однако информация — лишь инструмент знания, сама по себе знанием она не является. Современное общество характеризуется поиском идентичности, новых смыслов и новых истин. И в связи с этим оно все чаще и чаще обращается к информационным технологиям, чтобы найти средства, которые позволят удовлетворить его растущие потребности в новых быстроменяющихся реалиях. Стремление к реализации этой потребности привело к тому, что информационные технологии все больше проникают во все сферы жизни общества.

Развитие информационного общества — один из национальных приоритетов Республики Беларусь. Начиная с 2010 года развитие информационного общества является одним из основных факторов обеспечения конкурентоспособности и инновационного развития национальной экономики, совершенствования системы государственного управления, повышения зрелости гражданского общества. Все это диктует необходимость выработки новых подходов к созданию научно-технической основы экономики, определяющих будущее динамичное поступательное движение Беларуси по инновационному пути. Нужна долгосрочная стратегия формирования и развития модели белорусской экономики, основанной на интеллекте. Таковой является «Беларусь Интеллектуальная» и Стратегия «Наука и технологии: 2018–2040»[1]. Настоящая Стратегия определяет принципы государственной политики Республики Беларусь в сфере информатизации и основные направления развития информационного общества с учетом совокупности факторов, влияющих на его прогресс. К 2040 г. Беларусь должна обрести новое качество роста экономики и выход на мировой уровень конкурентоспособности на базе интеллектуализации и цифровизации производств, развития высокотехнологичных и наукоемких услуг, основанных на достижениях отечественной науки.

Стратегия развития информационного общества в Республике Беларусь основывается на исторически сложившейся системе образования, в значительной степени ориентированной на подготовку кадров естественнонаучных и технических специальностей, что связано с обслуживанием наукоемких производств, в том числе в области информационно-коммуникационных технологий (ИКТ). ИКТ стали необходимым инструментом социально-экономического прогресса, одним из ключевых факторов инновационного развития экономики. В условиях бурного роста в Республике Беларусь отрасли ИКТ университетское образование должно удовлетворять ее запросам и развивающегося информационного общества в целом. Стратегической целью дальнейшего развития информатизации в Республике Беларусь является совершенствование условий, содействующих трансформации сфер человеческой деятельности под воздействием ИКТ, включая формирование цифровой экономики, развитие информационного общества и совершенствование электронного правительства Республики Беларусь.

С 2017 г. Беларусь взяла курс на цифровизацию: предполагается, что IT-технологии должны стать одной из главных составляющих новой экономической модели. Цифровизация — это

внедрение цифровых технологий в разные сферы жизни для повышения её качества и развития экономики. Она помогает выполнять рутинные задачи и принимать решения без участия человека. Ввиду преимуществ, которые сегодня предоставляют цифровые технологии и создаваемые с их помощью цифровые продукты, в Беларуси реализуются масштабные проекты (например, проекты строительства цифровой экономики), направленные на цифровизацию производства, сельского хозяйства, транспорта, бизнеса, государственного управления, медицины, образования и др. Оценить уровень цифровизации Беларуси позволяют различные индексы развития цифровой экономики и статистические данные. В Беларуси 85,1% населения – по данным на 2022 г. – являются пользователями интернета. [2]

По состоянию науки и инновационной деятельности, Беларусь входит в число государств с высоким уровнем научно-технического развития. За последние годы удалось сохранить и укрепить научный потенциал страны. Научные исследования и разработки в Беларуси в 2022 г. выполняли 445 организация и около 26 тысяч человек [3]. Все это свидетельствует об актуальности задач расширенного воспроизводства научно-технического потенциала, и, прежде всего, оптимизации кадрового состава белорусской науки преимущественно на основе притока молодых ученых. Важно обеспечить дальнейшее наращивание компетенций в новых прорывных областях и увеличение влияния науки на экономический рост в долгосрочной перспективе. Отдельные резиденты Парка высоких технологий Беларуси участвуют в разработке систем машинного обучения в рамках проектов аутсорсингового программирования.

Перспективы формирования информационного общества в Республике Беларусь:

- заложена и совершенствуется правовая система управления процессами информатизации;
- развивается национальная информационно-коммуникационная инфраструктура, позволяющая оказывать телекоммуникационные и информационные услуги на основе технологий широкополосного доступа;
- обеспечение прозрачности и удобства коммуникаций между гражданами, бизнесом и государством путем повсеместного перевода данных коммуникаций в электронную форму.
- дальнейшее формирование единого информационного пространства для оказания электронных услуг на основе интеграции информационных систем;
- развитие и внедрение технологий блокчейн, больших данных (проект «Электронный рецепт»), интернет вещей («умные» фонари, плата за проезд BelToll, («БелАЗ» оснащает свои изделия датчиками износа), облачных технологий (ActiveCloud и beCloud).

Помимо вышеназванных факторов, есть комплекс ограничивающих технических, правовых, организационных, экономических и других факторов. Ключевые проблемы формирования информационного общества в Беларуси:

- высокая стоимость технологий;
- подавляющее большинство технологий импортируется из более технологически развитых стран;
- существенные различия в уровне использования ИКТ («цифровое неравенство») между различными социальными группами населения, городской и сельской местностью, а также между различными отраслями экономики;
- недостаточный уровень компьютерной грамотности государственных служащих и населения в целом;
- отсутствие достаточного количества квалифицированных кадров среди населения;
- утрата информационным обществом устойчивости (из-за возрастания роли информации, малые группы могут оказывать существенное влияние на всех людей).

Очевидно, что процессы цифровизации будут (даже невысокими темпами) продолжаться. Новые технологии открывают новые перспективы. Появилась возможность качественно иного управления большими комплексами организаций и производством систем, требующим координации деятельности большого количества людей. Задачи развития информатизации, сформулированные в соответствии с факторами развития и направлениями использования ИКТ, в целом решены. Свидетельством этого являются оценки международных организаций, осуществляющих мониторинг развития сферы ИКТ в мире. Беларусь должна войти в состав мировых лидеров по перспективным направлениям научно-технологического развития, включая интеллектуальные технологии, умные материалы, машины и их системы для реального сектора и социокультурной сферы, а также разработки межотраслевого характера, основанные на применении нано-, био-, космических и IT-технологий. В ряде наукоемких высокотехнологичных областей Беларусь сегодня обладает ноу-хау мирового уровня.

Список использованных источников:

1. Стратегия развития информатизации в Республике Беларусь на 2016 – 2022 годы [Электронный ресурс]. – 2023. – Режим доступа: <https://edu-grodno.by/wp-content/uploads/2017/10/STRATEGIYA-razvitiya-informatizatsii-v-Respublike-Belarus-na-2016.pdf>

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. Чем живет виртуальная Беларусь [Электронный ресурс]. – 2023. – Режим доступа: <https://neg.by/novosti/otkrytj/digital-2022-ispolzovanie-interneta-i-socsetej-v-belarusi/>

3. Приоритеты о основные достижения белорусской науки. [Электронный ресурс]. – 2023. – Режим доступа: <https://lida.gov.by/ru/2023-edi-ru/view/prioritety-i-osnovnye-dostizhenija-belorusskoj-nauki-nauchno-texnologicheskaja-bezopasnost-janvar-22643/>

УДК 140.8

СОВРЕМЕННЫЕ ГЛОБАЛИЗАЦИОННЫЕ ВЫЗОВЫ И РИСКИ

Венгеренко В.В., магистрант гр.256201

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Чуешов В.И. – д-р филос. наук, проф. кафедры философии

Аннотация. Глобализация – объективная реальность современного мира. Невозможно отрицать происходящие в мире процессы унификации, стандартизации, интеграции во всех сферах человеческого общества. Являясь неким парадигмальным понятием, в рамках которого осмысливаются современные проблемы, глобализация представляет собой феномен, вопрос о сущности которого до сих пор является дискуссионным.

Ключевые слова. Глобализация, глобалистика, глобальные проблемы, цивилизация.

Глобализация представляет собой совокупность сложных интеграционных процессов, охватывающих постепенно все сферы человеческого общества [1]. Сам по себе этот процесс является объективным, исторически обусловленным всем развитием человеческой цивилизации. Глобализация является предметом изучения глобалистики.

Глобалистика – междисциплинарная область научных исследований, направленных на выявление сущности, тенденций и причин процессов глобализации, порождаемых ею глобальных проблем и поиск путей утверждения позитивных и преодоления негативных для человека и биосферы последствий этих процессов [2]. В более широком смысле термин употребляется для обозначения совокупности научных, философских, культурологических и прикладных исследований различных аспектов глобализации и глобальных проблем, включая полученные результаты таких исследований, а также практическую деятельность по их реализации в экономической, социальной и политической сферах, как на уровне отдельных государств, так и в международном масштабе. Глобалистику не следует понимать как какую-то отдельную или специальную научную дисциплину, которые во множестве возникают, как правило, в результате дифференциации научного знания или на стыке смежных дисциплин. Она рождена противоположными, интеграционными, процессами, характерными для современной науки, и представляет собой сферу исследований и познания, где различные научные дисциплины и философия, по большей части в тесном взаимодействии друг с другом, каждая с позиции своего предмета и метода, анализируют всевозможные аспекты глобализации, предлагают решения глобальных проблем, рассматривая их как в обособлении друг от друга, так и в качестве целостной системы [2].

В современной глобалистике термин «глобализация» употребляется, как правило, для характеристики интеграционных и дезинтеграционных процессов планетарного масштаба в области экономики, политики, культуры, а также антропогенных изменений окружающей среды, которые по форме носят всеобщий характер, а по содержанию затрагивают интересы всего мирового сообщества.

Проблема глобализации занимает обширное место в работах отечественных и зарубежных исследователей – философов, футурологов, экономистов, политологов. В них она предстает и как реальный и противоречивый феномен современности, и как основная тенденция развития будущего. Разброс мнений в оценках глобализации обусловлен противоречивостью тех процессов, в которых она протекает.

Предпосылкой появления феномена глобализации стало развитие научного и технического знания и развитие техники, давшее возможность отдельно взятому индивиду воспринимать органами чувств объекты, находящиеся в различных точках земли и вступать с ними в отношения, а также естественно воспринимать, осознавать сам факт этих отношений.

Можно отметить две крайности в понимании как самого феномена глобализации, так и истории его появления. Одна из них состоит в том, что планетарный характер социальных связей и отношений трактуют неправомерно расширительно, пытаясь усмотреть их уже в первобытном обществе, и с этой точки зрения даже ранние этапы развития человечества характеризуют как глобальные. Другая крайность заключается в том, что глобализацию понимают слишком узко, когда современные процессы общественного развития рассматриваются в отрыве от их генезиса, то есть не учитываются история и динамика становления международных структур и транснациональных связей. Такой разброс мнений и разница взглядов на глобализацию объясняются не только сложностью проблемы, но и недостаточной разработанностью этой темы [3]. В итоге затрудняется взаимопонимание между людьми, тормозится междисциплинарное взаимодействие, создаются серьезные препятствия на пути осмысления истинных причин глобализации и порождаемых ею

глобальных противоречий. Здесь же кроются причины и многих конфликтов, обусловленных тем, что мир все больше становится единым, целостным, взаимосвязанным, в то время как механизмы, призванные регулировать общественные отношения на глобальном уровне (мировое правительство, мировое государство, объединенные силы поддержания международного порядка и тому подобное), отсутствуют. Вполне очевидно, что без глубокого анализа и достаточно ясного понимания современных процессов глобализации трудно рассчитывать на успешное преодоление указанных выше проблем.

Научно-технический прогресс и становление глобального общества не только позволяют решать проблемы, но и вызывают возникновение новых глобальных же проблем. Их преодоление становится условием выживания как самого человека, так и в целом биосферы. К числу наиболее важных проблем на современном этапе можно отнести следующие:

1. Проблема предотвращения войны и сохранения мира.
2. Обеспечение экологического равновесия между человеком и окружающей природой.
3. Обеспечение продовольствием и другими необходимыми средствами – энергетическими, сырьевыми ресурсами – растущего населения планеты.
4. Необходимость обеспечения крепкого здоровья в условиях обострения экологических проблем и информационного этапа цивилизации (например, гиподинамия).
5. В условиях крушения биполярного мира обострилась угроза со стороны международной преступности и терроризма, которые обнаруживают все признаки перерастания в полномасштабную войну против человечества и гуманности как принципа мироустройства. Террористы и международная преступность, взяв на вооружение современные средства связи и технологий, способствуют усугублению угроз безопасности и стабильности в мире.

В современных условиях чрезвычайно актуален вопрос соотношения научно-технической революции и возможностей человека и человеческого сообщества в целом, а не только отдельных наиболее «продвинутых» его представителей и групп. Более того, часть наиболее образованных и успешных представителей развитых стран не склонны принимать ценностей своего типа и готовы разрушать собственные устои на стороне цивилизационного противника. Постиндустриальная цивилизация вслед за индустриальной продолжает разрушать традиционное сообщество, что нарушает относительный социальный баланс, хрупкое равновесие между цивилизацией и природой. Проблемы и трудности очевидны. Однако интеллектуальное сообщество, во всяком случае, та его часть, которая не стремится к обособлению от остального мира, готово предложить ему позитивные, оптимистические сценарии будущего. Наиболее перспективными и гуманными представляются следующие:

- перенесение фундаментальных достижений науки и техники в страны третьего мира;
- выявление универсальных компонентов антропологии, ментальности как основы межкультурного диалога;
- включение в фундамент общечеловеческого будущего базовых принципов традиционного сообщества.

Отрицательные стороны глобализации связаны не только с неконтролируемыми последствиями научно-технического прогресса. Одной из самых существенных угроз является, по общему признанию, тотальная унификация человеческого сообщества [4]. Именно это обстоятельство зачастую вызывает активное и временами неадекватное сопротивление процессу.

Но даже убежденные сторонники глобализации не склонны отрицать наличие и прочность существующих культурных и прочих границ и различий. Нет никаких признаков того, что отдельные религии и культурные целостности сольются воедино в результате миграций и глобализации, что человек потеряет своеобразие, «собственное лицо». Однако, таким образом, сохраняется основа для углубления противоречий не только в настоящем, но и в будущем.

Расширение географии и нарастающая плотность контактов в сфере международной торговли в современном обществе связаны, прежде всего, с появлением на экономической арене электронных денег, в связи с чем с полным правом можно говорить о формировании глобальной электронной экономики.

Развитие технологий с конца XX века позволило сформировать глобальную экономику взамен мировой (основанной лишь на функционировании глобальных потоков капитала), где возникновение глобальных рынков товаров и услуг дало возможность любой компании реализовывать свою продукцию в мировом масштабе. Современная глобализация мировой экономики выражается во множестве процессов (рисунок 1). Глобализация экономики детерминирует все процессы и требует соблюдение ряда факторов, под нее необходимо приспособить национальные экономические институты с новыми требованиями, усилить могущество владельцев капитала, инвесторов, многонациональных корпораций и глобальных финансовых институтов. Необходимо поддерживать формирование новых международных механизмов накопления и движения капиталов, способствовать органическому вхождению в этот процесс и поддерживать виртуализацию экономических границ между государствами в условиях глобализации для развития международной

торговли и мировой экономики. Технологическая глобализация может рассматриваться как элемент экономической глобализации, поскольку международный технологический обмен, отображающий мировое движение технологий, тесно связан с развитием производительных сил, которое, в свою очередь, представляет собой исходный пункт развития экономической глобализации.

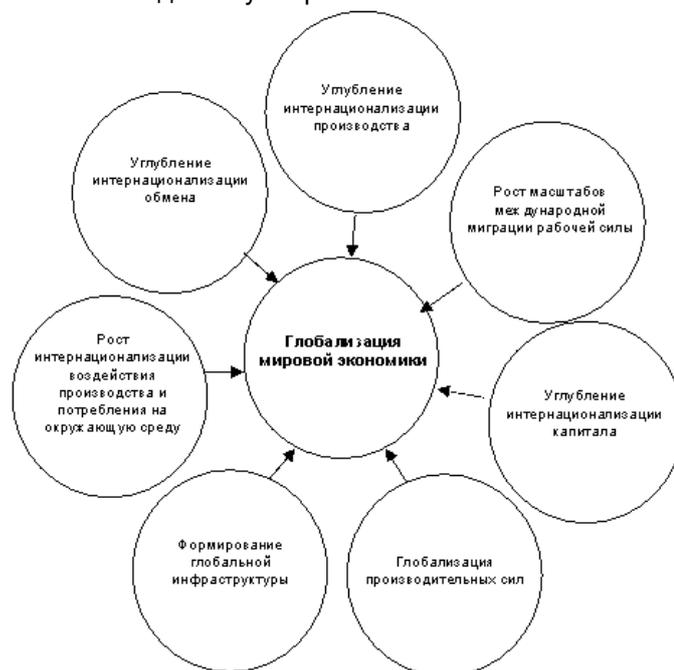


Рисунок 1 – Процессы современной глобализации мировой экономики

Помимо прочего, глобальная экономическая система предусматривает формирование глобального рынка труда [5]. И хотя на сегодняшний день еще рано говорить об окончательно сформировавшейся системе мирового рынка труда, труд все же постепенно становится глобальным ресурсом.

Таким образом, глобализация выражается в резком расширении и усложнении взаимосвязей и взаимозависимостей народов и государств. Это новая стадия общественного развития в общепланетарном масштабе, ставшая возможной благодаря достижениям науки и техники.

Существует множество трактовок понятия глобализации, не меньше и точек зрения ученых относительно ее происхождения, то есть выяснить суть феномена глобализации достаточно сложно. Сама этимология слова «глобализация» говорит о процессах планетарного масштаба, которые затрагивают различные сферы общественной жизни, имеют объективный характер и разворачиваются в естественно-историческом контексте.

Процессы глобализации, как объективное следствие предшествующего развития, являясь необратимым процессом, приводят к коренным изменениям в развитии современного общества, которые в своей совокупности становятся причиной возникновения новых для человечества проблем, таких как кризис идентичности, проблема мультикультурализма, ускорение времени и темпа жизни, изменение языковых традиций. Современное информационное общество при этом продолжает оставаться зависимым от сырьевых отраслей хозяйства, неинформационных ресурсов, производства материальных благ.

На сегодняшний день можно с уверенностью утверждать, что альтернативы глобализации нет, но есть альтернативы (варианты) в рамках глобализации. Глобализацию отменить нельзя, но возможно, более того, нужно выбрать свой путь вхождения в нее. Можно констатировать, что процесс глобализации явился реакцией на кризис существующей цивилизации, но при этом современный этап глобализации имеет стихийный характер, с тенденцией обострения рисков. Однако это не исключает положительных сторон этого процесса, реализация которых требует кооперации, совместного действия. Глобализация заключает в себе множество возможностей, и от их выбора будет зависеть сохранение целостности и единства мира.

Список использованных источников:

1. Радевич, Е.В. Трансформация дискурса глобализации в современном социально-гуманитарном знании / Е.В. Радевич // Журнал Белорусского государственного университета. Философия. Психология. – 2018. – № 2. – С. 29-34.
2. Глобалистика как научная дисциплина [Электронный ресурс]. – Режим доступа: <https://moodle.kstu.ru/mod/book/view.php?id=19628>. – Дата доступа: 15.03.2023.
3. Чумаков, А.Н. Глобализация. Контуры целостного мира : монография. – 3-е изд., перераб. и доп. / А.Н. Чумаков. – М. : Проспект, 2021. – 456 с.

4. Унификация всех культур: глобализация стирает границы [Электронный ресурс]. – Режим доступа: <https://mentamore.com/socium/unifikaciya-vsex-kultur-globalizaciya-stiraet-granicy.html>. – Дата доступа: 18.03.2023.

5. Линдси, Б. Глобализация: повторение пройденного. Неопределенное будущее глобального капитализма / Б. Линдси. – М. : Социум, 2020. – 417 с.

6. Философия, глобализация, интеграция / под ред. В.И. Чуешова. – Минск: Академия управления при Президенте Республики Беларусь, 2006. – 203 с.

UDC 140.8

MODERN GLOBALIZATION CHALLENGES AND RISKS

Vengerenko V. V.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Tchouechov V. I. – PhD, professor of the Department of Philosophy

Annotation. Globalization is an objective reality of the modern world. It is impossible to deny the processes of unification, standardization, and integration taking place in the world in all spheres of human society. Being a kind of paradigm concept within which modern problems are comprehended, globalization is a phenomenon, the question of the essence of which is still debatable.

Keywords. Globalization, globalistics, global problems, civilization.

ФИЛОСОФИЯ И СОВРЕМЕННАЯ НАУЧНАЯ КАРТИНА МИРА

Глинистый Р.Р, Матошко А.О.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Научный руководитель – Малахина Г.И., кандидат философских наук, доцент

Аннотация. Анализ философских оснований современной научной картины мира. Определение понятия «научная картина мира». Выявление принципиальных особенностей современной научной картины мира.

Ключевые слова. философия, мир, картина мира, научная картина мира.

Анализ «картины мира» как особого компонента научного знания предполагает предварительное выяснение смыслов исходных терминов — «мир» и «картина мира». Мир — это Вселенная во всей ее совокупности форм материи в земном и космическом пространстве, т.е. все то, что существует вокруг нас[1].

Научные картины мира отводят человеку определенное место во Вселенной и помогают ему ориентироваться в окружающей действительности. Они возникают как в рамках повседневной жизни, так и в ходе духовной деятельности человеческих сообществ. К основным картинам мира относятся: философская, религиозная, научная, художественная и др.

Философская картина мира — это обобщенная, выраженная философскими понятиями и суждениями, теоретическая модель бытия в ее соотнесенности с человеческой жизнью, осознанной социальной активностью и соответствующая определенному этапу исторического развития.

Главная тема философии — соотношение человека и мира, взятое во всех ракурсах: онтологическом, теоретико-познавательном, ценностном и деятельностном. Именно поэтому философские картины мира множественны и не похожи одна на другую. Их всегда объединяют между собой интеллектуальность рассмотрения и вечное сомнение в собственных утверждениях, постоянная критичность. Это резко отличает философское представление о мире от обыденных или религиозных взглядов и роднит философию с наукой.

Религиозная картина мира появляется вместе с рождением теологических систем христианства, иудаизма и ислама. В рамках религиозной картины мира реализуются.

С функции религии: мировоззренческая, компенсаторная, коммуникативная, регулятивная, интегративная и др.

Современные религии не отрицают достижений естествознания и науки в целом. Они не отрицают теорий, связанных со строением материи и не выступают против практического применения науки. Но они всегда подчеркивают, что дело науки — изучать только физический мир, природу. За рамками изучения природы простирается, по их мнению, область религии и, быть может, философии.

Научная картина мира возникает как альтернатива религиозной. Мир и человек здесь рассматриваются как объекты исследования. Первая научная картина мира (механическая) сформировалась в Новое время под сильным влиянием математического естествознания.

Основные элементы научной картины мира: научные знания о природе, научные знания об обществе, научные знания о человеке и его мышления. Она основана на данных астрономии, физики, химии, биологии и выступает обычно как антитеза архаической (прежде всего, религиозной) картине мира.

Основная форма человеческого познания в современном мире — наука. В наши дни наука оказывает все более значимое и существенное влияние на реальные условия нашей жизни, в которых нам так или иначе надлежит ориентироваться и действовать.

Картина мира, рисуемая современным естествознанием, необыкновенно сложна и проста одновременно.

Системный способ объединения элементов выражает их принципиальное единство: благодаря иерархичному включению систем разных уровней друг в друга, каждый элемент любой системы оказывается связан со всеми элементами всех возможных систем. Подобным образом организуются и научная картина мира, и создающее ее естествознание.

Кроме системного подхода для современной научной картины мира характерны эволюционный и синергетический взгляд на мир. Глобальный эволюционизм — это учение признающее невозможность существования Вселенной и всех порождаемых ею менее масштабных систем вне развития, эволюции.

Самоорганизация — наблюдаемая способность материи к самоусложнению и созданию все более упорядоченных структур в ходе эволюции. Синергетика (или теория самоорганизации) рассматривает мир как сложную саморазвивающуюся систему.

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

Говоря о роли философии и её роли в научном познании, следует указать на две крайние модели, которые сложились в решении ею гносеологических проблем: умозрительно-философский подход, позитивизм.

Реализация философских принципов в научном познании означает вместе с тем их переосмысление, углубление, развитие. Тем самым путь реализации функций философии есть не только способ решения фундаментальных проблем развития науки, но и способ развития самой философии, всех ее идей, воззрений и методологических принципов[2].

Список использованных источников:

1. *Философия: учеб. пособие / Андрейченко Г.В, Грачева В.Д. - Ставрополь: Изд-во СГУ, 2001. – 245 с..*
2. *Философия: учеб. пособие / Коллектив авторов. Литагент Высшейшая школа / Бородич А.А., Бодаков А.В и др. 2012. - с. 384.*

ФЕНОМЕН ТОЛПЫ В КОНТЕКСТЕ ЦИФРОВЫХ КОММУНИКАЦИЙ

Зинковская Е. В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Гуринович С. В.

Аннотация. Каждый индивид в своей жизни сталкивается с толпой, является её частью или просто наблюдателем. Предложенный материал направлен на раскрытие термина толпы, установления основных ее характеристик, а также рассмотрения толпы в цифровой среде.

Введение. Интерес к исследованию феномена толпы относится к научной литературе 19 – начала 20 века и связан с такими именами как Г. Лебон, З. Фрейд, Хосе Ортега-и-Гассет. Изучается данный феномен в социальной психологии, социальной философии, социологии и психоаналитике.

Основная часть. Г. Лебон – психолог, социолог, антрополог, историк – и Хосе Ортега-и-Гассет – философ, социолог – рассматривают толпу синонимично массе. В толпе индивид не осознает свои поступки. Его действия не рациональны. Индивид, входящий в толпу, импульсивно реагирует на внешние раздражители, подвергаясь манипуляциям. Воздействовать на индивидов в толпе можно четырьмя способами: заражением, подражанием, внушением, убеждением.

В толпе индивид чувствует себя более уверенно, чем вне ее. Это происходит благодаря многократному «отражению» эмоционального состояния. Толпа не имеет структуры и организации поведения. В исследованиях Г. Лебона, З.Фрейда и других преобладала толпа реальная. Она рассматривалась как негативная и агрессивная.

В классическом подходе толпа – это неорганизованная группа людей, в которой психические правила меняются, а инстинкты становятся основным регулятором поведения.

В современном информационном обществе термин толпа осваивает принципиально новое пространство, а именно цифровое. В научной литературе такой толпе соответствуют понятия «виртуальная толпа» или «цифровая толпа». По сравнению с традиционной толпой, в этой толпе акцентируется внимание на позитивные стороны.

Г. Рейнольд, американский социолог, выделил новый тип толпы – «умная толпа» («smart mob»). Об «умной толпе» он говорит как о новом виде толп, консолидирующихся индивидов в синхронизированное целое на основе интернет технологий [2, с. 125]. Действия данной толпы планируются и управляются в интернете, а реализуются в реальном мире. Данная толпа организуема, дисциплинирована. Г. Рейнольд утверждает, что здесь рациональность индивида не уменьшается, а Д. Шуровьеский – журналист, писатель в «The New Yorker» – говорит, что она и вовсе увеличивается. Индивиды «умной толпы» способны совместно решать какие-либо проблемы. Их действия приводят не к разрушению, а к созданию чего-либо нового, улучшенного.

Д. Хау рассматривает новую толпу как «аккумулятор ресурсов». Он выдвинул термин «краудсординг», под которым понимал технологию социального действия, в основе которой лежит делегирование полномочий в решении разного рода задач массе интернет-пользователей, объединенных на различных сетевых площадках и выполняющих работу бесплатно или за небольшое вознаграждение [2, с. 126]. Одним из самых масштабных проектов, осуществляемых с помощью краудсординга, является википедия.

Несмотря на большое количество достоинств, «умная толпа» имеет свои недостатки. Примером является «троллинг», «буллинг». Распространение ложной информации, травля негативно сказываются на репутации данной толпы. Однако повышенная образованность и рассудительность индивидов «умной толпы» привела к блокированию других «негативных» толп и суггерендов.

Виртуальная толпа также является инструментом маркетинга, цель которого заставить индивидов данной толпы приобрести представленный товар. Происходит своего рода манипуляция.

Заключение. Обычная нам толпа сейчас сменяет «умной толпой». Она саморегулирующаяся, саморазвивающаяся, четко организована, использует свои возможности для достижения общего и собственного блага. Она не предполагает физического контакта. Цель данной толпы – обладание информацией. Происходит формирование цифровой солидарности. Негативные стороны сменяются положительными. Таким образом, феномен толпы в обществе информационных технологий становится более сложным и многозначным в структуре социальной коммуникации.

Список использованных источников:

1. Г. М. Андреева, *Социальная психология. М., Изд-во Моек, ун-та, 1980. 416 с. – С. 162- 167, 171.*
2. *Изв. Сарат. ун-та. Нов.сер. Сер. Философия. Психология. Педагогика. 2019. Т. 19, вып. 2. С – 125-128.*
3. Кара-Мурза С.Г., *Манипуляция сознанием. — М.: Изд-во: Эксмо, 2005., С – 232.*

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

4. *Психология стихийных социальных объединений: монография* / Д. С. Горбатов, В. В. Агафонова, Л. Т. Андреева, В. В. Волкова, Е. А. Горбатова / Под научн. ред. Д. С. Горбатова. — СПб.: издательство СПбГИПСР, 2021. С. — 9-10, 12-13.

5. Г. Лебон, *Психология толп*. — М.: Институт психологии РАН, Издательство «КСП+», 1999, С. — 127, 128, 131-132, 134-135.

ИСТИНА И ПУСТОТА ИЛИ КОНЦЕПЦИЯ ИСТИНЫ В ФИЛОСОФИИ ПОСТМОДЕРНИЗМА

Игнатюк Н.С., студент гр.215401/магистрант

*Белорусский государственный университет информатики и радиоэлектроники, г. Минск,
Республика Беларусь*

Малыхина Г.И. – канд. фил. наук, доцент

Аннотация. Постмодернизм – направление в западной философии конца 20-го века, характеризующееся широким скептицизмом, субъективизмом и релятивизмом, общим разочарованием в разуме. Постмодернистское движение возникло как ответ на модернистское представление об объективной истине. В постмодернистской интерпретации, истина является продуктом языка, культуры и динамики власти. Эта перспектива бросает вызов традиционным представлениям об истине и знании, и имеет значение для понимания современной реальности.

Ключевые слова. Постмодернизм, истина, нарратив, метанарратив, антифундаментализм, дискурс, деконструкция, симулякр, гиперреальность, гипертекст.

Введение

Постмодернизм часто описывается как движение, которое бросает вызов великим нарративам современности, включая веру Просвещения в объективную истину и прогресс. Согласно постмодернистам, истина не является фиксированной, а «объективная» реальность зависит от языка, культуры и властных отношений. Истина и «объективная реальность» являются продуктами дискурсов, нарративов и социальных структур, которые формируют наше понимание мира. Поэтому значение истины в постмодернизме всегда зависит от контекста, в котором она производится и интерпретируется. Эту перспективу часто называют антифундаментализмом, что означает, что нет абсолютных оснований для знания или истины. Антифундаменталисты оспаривают обоснованность оснований дискурса, задавая такие вопросы, как «Что, в свою очередь, гарантирует истинность вашего основания (то есть отправной точки)?».

Постмодернистский взгляд на истину возник в середине XX века, прежде всего в ответ на доминирование научного и рационалистического мышления. Постмодернисты утверждали, что эти способы мышления не учитывают сложность и разнообразие человеческого опыта, и игнорируют роль языка и культуры в формировании нашего понимания реальности. Поэтому постмодернисты стремились деконструировать традиционные представления об истине и знании и бросить вызов властным структурам, которые, по их мнению, поддерживали этот образ.

Истина и Власть

Постмодернисты утверждают, что истина — это не нейтральная или объективная перспектива, а продукт конкретных властных отношений. Более того, производство и распространение истины всегда опосредовано соотношением сил. Таким образом, доминирующие дискурсы, которые определяют, что представляет собой истина, отражают ценности и интересы тех, кто имеет право определять истину.

Постмодернисты стремятся бросить вызов властным структурам, которые поддерживают эти доминирующие дискурсы, и создать пространство для альтернативных способов познания и мышления. Поэтому постмодернисты утверждают, что мы должны критически относиться к тому, как властные структуры формируют наше понимание реальности. Это означает, что мы должны подвергать сомнению доминирующие дискурсы и искать альтернативные мировоззрения, которые бросают им вызов.

Истина и Реальность

В современном мире реальность представляется изменчивым понятием, постоянно меняющимся и эволюционирующим. В современном мире грань между правдой и вымыслом, фактом и мнением, реальностью и симуляцией, и даже добром и злом становится все труднее различить. Французский философ Жан Бодрийяр одним из первых заговорил о данном явлении. Он бросил вызов традиционным представлениям об истине, утверждая, что сама реальность стала настолько искаженной, что теперь она находится за пределами нашего понимания. В своей основополагающей работе «Симулякры и симуляция» Бодрийяр предполагает, что мы больше не живем в обществе, где истина является высшей ценностью, а скорее в обществе, где гиперреальность и симуляция взяли верх. Этот сдвиг, утверждает он, имеет глубокие последствия для нашего понимания идентичности, политики и культуры.

Истина и Язык

Постмодернисты утверждают, что язык — это не «зеркальная» среда, отражающая реальность, а социальная конструкция, которая формирует наше понимание реальности. Поэтому значение языка (слов, которые его составляют) всегда зависит от контекста, в котором он используется. Более того, смыслы (слова), наполняющие язык, не фиксированы, а постоянно искажаются и меняются. Поэтому постмодернисты стремятся деконструировать язык. Деконструкция языка включает в себя критический анализ предположений и предубеждений, лежащих в основе использования языка. Постмодернисты утверждают, что язык является не нейтральным инструментом общения, а продуктом конкретных исторических и культурных контекстов. Таким образом, язык отражает ценности и интересы тех, кто имеет право определять, что представляет собой истина. Деконструируя язык, постмодернисты стремятся выявить властные отношения, которые формируют наше понимание реальности, и бросить вызов доминирующим дискурсам, которые поддерживают эти властные отношения.

Заключение

В заключение, постмодернизм оказал значительное влияние на наше понимание реальности и на то, как мы думаем о знании и истине. Он бросил вызов великим нарративам современности и подчеркнул случайность и сложность человеческого опыта. Более того, постмодернизм подчеркнул роль языка, культуры и властных отношений в формировании нашего понимания реальности, также оказал значительное влияние на различные области, включая литературу, искусство, философию и социальные науки: постмодернистские писатели и художники стремились бросить вызов доминирующим дискурсам в искусстве и пытались создать «новые панорамные площадки с видом на всеобщие проблемы», а философы-постмодернисты и социологи стремились разработать новые способы интерпретации знаний и истины, которые были бы более инклюзивными и критическими.

Также можно выделить некоторые основные термины и идеи при рассмотрении концепции истины в философии постмодернизма:

- 1) гипертекст — понятие структурной лингвистики и постмодернизма, обозначающее максимально широкую языковую реальность;
- 2) «смерть автора» — уничтожение всех табу, идолов и авторитетов, поднимающихся над читателями гипертекста;
- 3) прибежание к «коллажированию» (примером этого может являться любой текст: «герменевтический выхлоп множественных текстов»), парадоксам (если правдивыми могут быть все высказывания, значит правильно и то, что все высказывания ложны) и мозаичному мышлению;
- 4) принципиальное сомнение в гносеологических презумпциях реализма, предполагающее, что далеко не все может быть «показано», «изображено», «иконизировано»;
- 5) «гибридизация» и смешение различных жанров и форм творческой деятельности и растворение философского мышления в стихии художественно-литературных дискурсов.

Список использованных источников:

1. *Sim S. Postmodernism and Philosophy / Сим. С. Постмодернизм и философия // THE ROUTLEDGE COMPANION TO POSTMODERNISM: сб. ст.; отв. Редактор Сим С. — Лондон: Иконные книги, 1998. — с. 3-15.*
2. *Welche Gefahren birgt die Postmoderne? / Чем опасен постмодернизм? — Режим доступа: <https://www.gotquestions.org/Deutsch/Gefahren-der-Postmoderne.html>*
3. *Postmodernism / Постмодернизм. Режим доступа: <https://www.britannica.com/topic/postmodernism-philosophy>*
4. *Печенкина О.А. Этика симулякров Жана Бодрийяра: автореферат диссертации по философии. Тула, 2006.*
5. *Керимов Т.Х. Постмодернизм // Современный философский словарь. М.; Бишкек; Екатеринбург, 1996.*
6. *Пономаренко И.Н. Постмодернизм и гипертекст // Известия высших учебных заведений. Северо-Кавказский регион. Общественные науки. — 2005 — №7 — С.98-103.*
7. *Малыхина, Г.И. Философия и методология науки: учеб. пособие / Г.И. Малыхина, В.И. Чуешов, В.И. Миськевич. — Минск: БГУИР, 2017 — С.260.*
8. *Зеленков А.И. Конспект лекций по курсу «Философия в современном мире». — Минск: БГУ — 2007 — С.61.*

АНАЛИЗ НАПРАВЛЕНИЙ ФИЛОСОФИИ ТЕХНИКИ

Ильина А.Ю.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Чуешов В.И. – д-р. филос. наук, профессор

Аннотация. Техника рассматривается как предмет философской рефлексии. Анализируются основные направления философии техники.

Считается, что техника зародилась вместе с человеческим обществом – значительно раньше научного знания. Техника порождается человеком и является неотъемлемой частью его жизни, опосредуя любую человеческую деятельность. Техника существует повсюду в различных формах и все больше определяет жизнь отдельного человека и общества в целом. Она преобразует не только человеческую деятельность, но и в ходе нее – самого человека [1]. Современная цивилизация является техногенной (порожденной техникой), и была бы невозможна без техники. Философия техники осмысливает отношения «человек-техника», исследует техническое миропонимание, изучает феномен техники в целом, ее внутреннее развитие и место в общественном развитии, принимая во внимание широкую историческую перспективу [2], ищет пути разрешения кризиса техники.

Результатом осознания противоречивой природы феномена техники стало появление ряда направлений в философии техники. Среди них выделяют технократические, считающие технику абсолютным благом, способным решить любые проблемы человечества, и технокритические, настороженно относящиеся к технике и негативно оценивающие перспективы ее развития [3].

В современной философии техники выделяют четыре крупных направления: сциентистское, социологическое, антропологическое и религиозное. Данные направления анализируют взаимосвязь техники с наукой, обществом, человеком и верой соответственно [2].

Сциентистское (от англ. «science» – наука) направление рассматривает технику как реализацию научных знаний, часть системы «наука-техника».

Социологическое направление анализирует взаимоотношения техники и общества и делится на две ветви: техницизм и антитехницизм. Техницизм утверждает всемогущество техники, ее способность решить любые проблемы современного общества. В антитехницизме техника предстает источником всех бед человечества. Техницизм и антитехницизм последовательно сменяют друг друга во времени. Несмотря на кажущуюся взаимоисключаемость позиций техницизма и антитехницизма, обе концепции исходят из определяющей роли техники в развитии общества (технологический детерминизм), давая при этом ей диаметрально противоположные оценки. Техницизм можно охарактеризовать как технократический утопизм, а антитехницизм – как экологический пессимизм [2].

Антропологическое (от греч. «anthropos» – человек) направление рассматривает техническую среду как способ существования человека. Считается, что человек техникой восполняет свою биологическую недостаточность [2].

Религиозная философия техники является попыткой найти в религиозной вере спасение от технического пессимизма [2] и рассматривает технику как реализацию божественной воли.

Выделяются и обсуждаются три философских взгляда на отношения техники и общества [4]:

- техника – автономная сила, определяющая общество (техника определяет общество);
- техника формируется человеческими ценностями (человек и общество определяют технику);
- козволюция техники и общества (техника и общество не определяют друг друга).

Все три точки зрения присутствуют в текущем обсуждении технических изменений и их влияния на общество. Считается, что третья точка зрения обладает наибольшим потенциалом, поскольку интегрирует идеи двух других взглядов и наиболее адекватно описывает реальность.

В заключение следует отметить, что техника выступает средством достижения других целей и сама по себе нейтральна. Техника противоречива – за то, пользу или вред она принесет, ответственен человек. Поэтому современный человек должен быть не только образованным профессионалом, но также нравственным и гуманным. Техника может оцениваться с позиций техницизма и антитехницизма, однако с диалектической точки зрения наиболее полно суть техники отражает синкретическая программа, принимающая научный прогресс как данность и рассматривающая и положительные результаты развития техники, и негативные последствия.

Список использованных источников:

1. Ишелев, А. И. Техника и технология / А. И. Ишелев // Актуальные проблемы философии науки и техники : сборник научных трудов, Ульяновск, УлГТУ, 2018 – Ульяновск, 2018 – С. 45-48.

59-я научная конференция аспирантов, магистрантов и студентов БГУИР

2. Абабилова, Л. С. *Философия техники: предмет, основные сферы и главная задача* / Л. С. Абабилова, А. Г. Дронкина // *Позиция. Философские проблемы науки и техники*. – 2019. – № 13. – С. 152-157.
3. Малыгина, Г. И. *Философия и методология науки : учеб. пособие* / Г. И. Малыгина, В. И. Чуешов, В. И. Мискевич – Минск : БГУИР, 2017. – 274 с.
4. Van de Poel, I. *Three philosophical perspectives on the relation between technology and society, and how they affect the current debate about artificial intelligence* / I. van de Poel // *Human Affairs*. – 2020. – Vol. 30.

ЭТИЧЕСКИЕ ОСОБЕННОСТИ ИНТЕРНЕТ-КОММУНИКАЦИИ

Карлюк П. А.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Бархатков А. И. – канд. филос. наук, доцент

Аннотация. С развитием технологий и увеличением количества пользователей сети «Интернет», коммуникация через сеть стала неотъемлемой частью нашей повседневной жизни. Однако интернет-коммуникация связана с рядом этических проблем, которые могут негативно сказаться на обществе в целом.

Этические особенности интернет-коммуникации в основном связаны с тем, что при заходе пользователя во всемирную паутину, могут размываться определенные социальные границы, и это является причиной изменения его поведения. Безусловно, свобода является важной ценностью сетевого сообщества, однако свобода одного пользователя не может нарушать личных границ другого. Этические проблемы интернет-коммуникации являются следствием перевоплощения пользователя. Их можно условно поделить на две основные группы.

К первой группе будут относиться этические проблемы, связанные с понятием «сетевой этикет», которое представляет собой неологизм, появившийся вместе с популяризацией среди населения и, соответственно, увеличением числа пользователей Интернета. Сетевой этикет – это совокупность правил, служащих для уважительного и уместного общения в сети Интернет. Среди важных и распространенных этических проблем, связанных с сетевым этикетом, можно выделить следующие явления: игнорирование, флейм, флуд, спам, оффтоп. Игнорированием считается намеренное оттягивание ответа на какой-либо запрос. В свою очередь, флеймом называется бурное обсуждение чего-либо с последующим отклонением от изначальной темы и, в конечном итоге, переход на личности. Флуд – это сообщения, которые не несут никакой полезной нагрузки. Сообщения, которые приходят от неизвестных лиц, которым вы не давали на это разрешения, называются спамом. Оффтоп – это запись, не соответствующая тематике обсуждения. Вышеизложенные этические проблемы интернет-коммуникации, представляющие собой нарушения сетевого этикета, чаще всего встречаются в личном или групповом общении. Они являются характерными для новых, чаще всего молодых, пользователей сети Интернет.

Помимо тех проблем, которые относятся к первой группе, существуют более серьезные и опасные. Они относятся ко второй группе этических проблем интернет-коммуникации, среди которых можно выделить следующие: защита личных данных, пиратство, злоупотребление технологиями и ложная информация. Одной из наиболее важных этических проблем в интернет-коммуникации является защита личных данных пользователей. Эта проблема возникла из-за неопытности подавляющего большинства абонентов Сети. В Интернете пользователи могут оставлять свои личные данные на различных сайтах и сервисах, что может привести к их утечке или неправомерному использованию. Другой важной этической проблемой является нарушение авторских прав. В Интернете многие пользователи используют контент, который находится под авторской защитой, не получая на это разрешения. Это может привести к ущербу авторам и компаниям, которые инвестировали средства в создание контента. Такая этическая особенность интернет-коммуникации, как злоупотребление технологиями, касается множества отраслей человеческой жизни. Интернет не должен использоваться для взлома, фишинга, мошенничества и других форм преступной деятельности. И последней проблемой является ложная информация. Интернет хранит в себе огромные объемы информации, которая может быть как истинной, так и ложной, что без должной проверки может причинить вред пользователю.

Таким образом, этические проблемы интернет-коммуникации являются актуальными и требуют серьезного внимания. Для их решения необходимо разработать и прививать правила хорошего тона. Например, к первой группе этических проблем, можно отнести такие правила, как обращение на «вы», оформление сообщений и изучение привычек определенного сетевого сообщества, перед тем, как стать его участником. В свою очередь, проблемы, которые принадлежат ко второй группе, требуют индивидуального подхода к каждой. Для решения проблемы защиты личных данных необходимо следить за безопасностью своей информации, не доверять её хранение сомнительным сервисам и использовать надежные пароли. Для решения проблемы пиратства необходимо следить за тем, какой контент можно использовать без разрешения авторов, и использовать только тот контент, который находится в свободном доступе. Для борьбы с злоупотреблением технологией в Интернете необходимо установить механизмы защиты от кибератак и киберпреступлений. Организации, предоставляющие платформы для общения в Интернете, должны установить системы защиты, которые помогут предотвратить фишинг, взлом и другие формы кибератак. Также важно обучать людей базовым мерам безопасности в Интернете, таким как использование надежных паролей, обновление программного обеспечения и установка антивирусного программного обеспечения. Для борьбы с ложной информацией в Интернете необходимо установить механизмы проверки фактов и контроля качества информации. Это может быть достигнуто с помощью алгоритмов искусственного интеллекта и машинного обучения.

ФЕНОМЕН ЭВТАНАЗИИ: СВОБОДА ВОЛИ ИЛИ АНТРОПОЛОГИЧЕСКИЙ КРИЗИС СОВРЕМЕННОГО ОБЩЕСТВА?

Радюкевич А.И., студент гр.153502, Кирзнер А.П., студент гр.153502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Лагунова Е.Н. – преподаватель

Аннотация. Эвтаназия - одна из сложнейших этического-философских проблемы противоречивое социальное явление. В своей работе мы рассматриваем, как меняет феномен эвтаназии представления человека и общества XXI века о жизни, смерти, свободе и ответственности.

Ключевые слова. Эвтаназия, свобода воли, ответственность, ценность жизни, уникальность личности.

Эвтаназия, или «милосердная смерть», представляет собой акт завершения жизни человека, проводимый с целью облегчения его страданий. Существует два основных вида эвтаназии: активная и пассивная. Активная эвтаназия – это процесс активного завершения жизни путем введения летальной дозы лекарств. Пассивная эвтаназия – это прекращение медицинского лечения и ухода за пациентом, что приводит к естественному смертельному исходу.

В мире существует несколько международных и национальных организаций, которые занимаются защитой прав пациентов и выступают за легализацию эвтаназии. Некоторые из них: World Federation of Right to Die Societies, Compassion & Choices, Dignitas, Exit International. Эти организации поддерживают право людей на выбор в отношении своей конечной жизни и работают над изменением законодательства и общественного мнения по этому вопросу.

Швейцария имеет длительную историю практики помощи в самоубийстве, которая является легальной с 1942 года. Помощь в самоубийстве осуществляется в Швейцарии организациями, такими как Exit и Dignitas, которые помогают людям, желающим завершить свою жизнь, получить летальную дозу медикаментов. Однако проведение эвтаназии все еще является незаконным в Швейцарии, за исключением некоторых кантонов, где это разрешено при определенных условиях. В Нидерландах проведение эвтаназии было разрешено на практике с 1973 года, а в 2002 году был принят закон о легализации. С течением времени и в некоторых других странах частично или полностью была разрешена эвтаназия, однако до сих пор она остается предметом острых дебатов и споров в обществе.

Существует множество случаев, которые показывают, что эвтаназия может быть использована для облегчения страданий неизлечимо больных людей, которые не могут быть вылечены и чье состояние безнадежно. Например, в 2012 году в Нидерландах эвтаназия была проведена для молодого человека, который страдал от неизлечимой болезни нервной системы. Он выразил желание уйти из жизни, когда его страдания станут невыносимыми. Его семья и врачи тщательно обсудили его состояние и приняли решение провести эвтаназию. Процедура прошла безболезненно и с уважением к желанию пациента.

В то же время есть случаи, когда позже оказывается, что эвтаназию провели зря, или когда этот вопрос оспаривается в суде. К примеру, в 2016 году в Нидерландах врачи провели эвтаназию для женщины, которая страдала от болезни, вызванной отравлением железом. Однако, позже стало известно, что она могла быть вылечена, и ее муж обратился в суд, утверждая, что проведение эвтаназии было незаконным. Суд принял решение, что врачи не провели достаточно тщательного обследования пациентки и признал проведение эвтаназии незаконным. Следует отметить, что такие случаи являются редкими и требуют дополнительных исследований, чтобы определить, была ли проведена эвтаназия в соответствии с медицинскими и юридическими процедурами.

Статистика по проведению эвтаназии в мире сильно различается в зависимости от страны и региона, так как законодательство, практики здравоохранения и культурные и социальные нормы значительно отличаются. В Нидерландах в 2020 году было проведено около 7,000 случаев эвтаназии, что составляет около 4% от общего числа смертей в стране. В других странах, например, в Германии и Франции, эвтаназия все ещё запрещена законом, хотя были приняты решения, которые предоставляют определенные права на самоопределение в конце жизни. В США, где законодательство по эвтаназии регулируется отдельными штатами, только в нескольких из них эта процедура легальна.

При анализе феномена эвтаназии можно увидеть противоречивые понятия, описывающие и объясняющие эту проблему: свобода воли, ответственность, ценность жизни, уникальность личности, социальный долг.

Свобода воли – это одно из основных прав, которое закреплено во многих международных документах, включая Всеобщую декларацию прав человека. Свобода воли означает, что каждый человек имеет право принимать решения относительно своей жизни и здоровья. Однако, в контексте эвтаназии, свобода воли может быть ограничена, так как эвтаназия может быть запрещена законом и медицинской этикой. С другой стороны, ответственность – это понятие, связанное с последствиями принятых решений. В контексте эвтаназии, врач, принимающий решение о проведении процедуры, несет ответственность за ее результаты и может быть привлечен к уголовной ответственности, если она была проведена незаконно или ненадлежащим образом, также ответственность – это осознание человеком того, что решение об эвтаназии принимается им самостоятельно. Ценность жизни – это понятие, которое может быть интерпретировано по-разному в разных культурах и обществах. В контексте эвтаназии, споры вокруг ценности жизни связаны с тем, должны ли люди иметь право на самостоятельное решение о своей жизни, и может ли жизнь, которая не приносит больше никакой радости, считаться ценной. В то же время, ценность жизни может противоречить праву на эвтаназию, так как человеческая жизнь считается ценностью самой по себе, и ее нельзя просто так отбросить. Уникальность личности – это то, что делает каждого человека уникальным и ценным. Этот аспект также может противоречить эвтаназии, так как смерть человека приведет к невозможной потере его уникальности. Социальный долг также может столкнуться с правом на эвтаназию, так как мы все несем ответственность за заботу о наших близких, и отказ от помощи может быть считаться неэтичным.

Некоторые из аргументов за легализацию эвтаназии: уважение к правам человека (считается, что каждый человек имеет право на свободу выбора, в том числе и на то, как и когда закончить свою жизнь); лучшее качество жизни (эвтаназия может предоставить человеку возможность избежать долгой и мучительной болезни); снижение страданий (эвтаназия может помочь избежать страданий и боли, которые сопутствуют терминальной болезни).

Некоторые из аргументов против легализации эвтаназии: религиозные соображения (многие религии считают, что только Бог имеет право на то, чтобы решать, когда человек должен умереть); риск злоупотребления (есть опасение, что её применение не будет строго контролироваться); ошибка в диагнозе (иногда бывает сложно определить, насколько неизлечима болезнь и может ли человек поправиться).

Также следует рассмотреть вопрос о развитии медицины в связи с легализацией эвтаназии. С одной стороны, легализация эвтаназии может способствовать развитию паллиативной медицины, которая нацелена на облегчение страданий больных терминальными заболеваниями. Она может также стимулировать развитие новых методов лечения и заботы о пациентах в конечной стадии заболевания, что в свою очередь может улучшить качество медицинской помощи и увеличить уровень ухода за пациентами. С другой стороны, некоторые оппоненты эвтаназии считают, что ее легализация может привести к ущемлению прав больных на должную медицинскую помощь, поскольку некоторые врачи могут выбирать путь эвтаназии вместо более трудоемких и дорогостоящих методов лечения. Это может привести к сокращению инвестиций в медицинскую науку и исследования, так как финансирование медицины может стать меньше, если будет сокращено число пациентов, получающих долгосрочное лечение.

Сопоставляя всё вышеизложенное, можно утверждать, что эвтаназия – это проблема, которая не имеет легких решений. Каждый случай является уникальным, и его нужно рассматривать в контексте конкретных обстоятельств. В то же время необходимо помнить, что у каждого человека есть право на свободу выбора и решение, как он хочет умереть, может быть принято только им самим. С другой стороны, есть риск злоупотребления и возможность ошибок в диагнозе. В любом случае, эта практика должна быть тщательно регулируемой и контролируемой, поэтому важно найти баланс между защитой жизни и уважением к индивидуальной свободе выбора.

На наш взгляд, можно сделать вывод, что эвтаназия как ответственное решение цивилизованного человека XXI века может быть легализована. Во-первых, у каждого человека должно быть право на самоопределение и свободу выбора, в том числе и в отношении своей жизни и смерти. Это основополагающее право человека на свою жизнь, которое должно быть уважаемо даже в тех случаях, когда человек решает завершить свою жизнь. Во-вторых, эвтаназия может оказаться самым гуманным решением в некоторых случаях, когда больной находится в состоянии невыносимых страданий и не имеет никаких шансов на выздоровление. В таких случаях, эвтаназия

может помочь избавить больного от боли и страданий, которые могут продолжаться долгие месяцы или даже годы. В-третьих, легализация эвтаназии может помочь урегулировать этот вопрос и сделать его более прозрачным и безопасным для всех участников. Благодаря законодательному регулированию, будет установлено, какие случаи могут быть рассмотрены для эвтаназии, и какие процедуры должны быть выполнены перед ее проведением, что поможет предотвратить возможные злоупотребления. Однако, прежде чем принимать решение о легализации эвтаназии, необходимо учитывать мнение всех заинтересованных сторон и проводить дополнительные исследования, чтобы убедиться, что эвтаназия может быть реализована в безопасной и этичной форме.

Список использованных источников:

1. Рыбин В.А. *Философские основания проблемы эвтаназии: методологический анализ* / В.А.Рыбин. – Екатеринбург: Изд-во Екатеринбургского гос. ун-та, 2006. – 41 с
2. Рыбин В.А. *Эвтаназия как проблема современной медицины и медицинской этики* / В.А.Рыбин // *Человек в современных философских концепциях: материалы международной научной конференции (Волгоград, сент. 1998 г.)*. – Волгоград, 1998. - С.336-341.

ИСКУССТВЕННЫЕ ЯЗЫКИ И СФЕРЫ ИХ ПРИМЕНЕНИЯ

Курьян А.И.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

*Мозалевская Д.А. – магистр философских наук,
преподаватель кафедры философии БГУИР*

Аннотация. В работе исследуется сущность естественных и искусственных языков. Приведена классификация искусственных языков, обозначены недостатки и преимущества их практического использования. Отмечены варианты дальнейшего развития и повышения эффективности применения искусственных языков в IT-сфере.

Ключевые слова. Естественные языки, искусственные языки, языки программирования.

Языки играют огромную роль во взаимодействии с окружающим миром, обмене информацией между людьми. Так, в ходе человеческой эволюции и необходимости передачи культурно-исторического опыта формируются естественные языки, которыми мы пользуемся с самого детства на ежедневной основе (русский, английский, язык жестов и другие). Ключевая цель естественных языков – общение. Однако, с постепенным ускорением темпа жизни общества, возрастанием объема информации, усложнением социальных отношений усиливается потребность в решении конкретных задач (познавательных, коммуникативных, технических). Языки, которые создаются специально, с определённой целью называются «искусственными». Например, написание кода с помощью языка программирования, запись формулы в физике для летящего предмета, наглядное демонстрирование реакций химических веществ с помощью уравнений. Существует больше тысячи искусственных языков (эсперанто, химические нотации, нотная грамота, языки программирования и другие) [1].

Искусственные языки применяются в исследовательской работе, создании новых продуктов, анализе или структурировании в блоки нужной информации, в художественном творчестве.

Цели создания искусственных языков могут быть разными. Рассмотрим следующие виды:

- **Философские и логические языки**
Философский язык Джона Уилкинса. Автор данного языка положил основу метрической системе, а также предложил заменить десятичную систему счисления на восьмеричную по причине лучшей делимости на два, что предвосхитило бинарную систему современных компьютеров [2]. Токипона («Язык добра») – язык, созданный лингвистом Соней Ланг, которая хотела проверить насколько простым может быть язык: сколько слов на самом деле нужно, чтобы описать мир и понять смысл жизни, друг друга [3]. Простая минималистская структура и краткий словарь всего из 120 слов, который не содержит никаких негативных коннотаций, были разработаны на основе гипотезы Сепира–Уорфа, согласно которой структура языка влияет на мировоззрение или познание его носителей. В 2022 году ISO 639-3 (код для представления названий языков) принял код «too1» для Токипона, классифицируемый как мировой язык [4].
- **Искусственные языки для межнационального общения**
Одним из самых известных языков данной категории является Эсперанто – это язык с самым большим количеством носителей. Обычно его изучают, чтобы позже было легче выучить европейские языки, на которых и строится эсперанто. Хотя изначально в плане Людвига Заменгофа – создателя эсперанто – данный язык должен был стать универсальным для человечества, а также способствовать мирному общению и существованию разных культур и народов [5].
- **Искусственные языки в литературе и кино**
По праву, самые известные языки – это Языки Толкина. Но они отличаются от многих других языков данного типа, поскольку сначала Джоном Рональдом Руэлом Толкиным были созданы сами языки. И только после этого были написаны такие известные произведения как «Хоббит» и «Властелин колец». Дж. Р. Р. Толкин ещё с детства увлекался созданием искусственных языков, и как признался позднее, написал книги лишь для того, чтобы его языкам было место, где жить.

Важно отметить, что в повседневной жизни не каждый человек будет пользоваться искусственным языком. Представим ситуацию: с работы пришёл отец и принёс детям игрушку, но, прежде чем отдать её, мужчина просит угадать, что же это за игрушка. Дети не понимают своего

УДК

ЧЕЛОВЕК В СИСТЕМЕ СОЦИАЛЬНЫХ КОММУНИКАЦИЙ. ЛИЧНОСТЬ И ОБЩЕСТВО

Нгуен В.Т.А., магистрант гр. 215101

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Малыхина Г. И. – кандидат философских наук, доцент

Аннотация. В тезисах рассматривается человек в многообразии его характеристики в системе социальных коммуникаций; личность, общество и их взаимоотношение.

Ключевые слова. человек, коммуникация, личность, общество

Человек – это высшая ступень живых организмов на Земле, субъект общественно-экономической деятельности, культуры и социальных отношений, носитель социально значимых качеств.

Как целостное существо (в единстве биологического, социального и духовного) человек представляет собой уникальное явление. Рассматривая человека в многообразии его характеристик, пользуются такими понятиями, как: «индивид» (отдельный представитель рода человеческого); «индивидуальность» (совокупность отличительных черт личности); «личность» (человек в результате социализации и самообразования). Склонности и способности, темперамент и характер, талант и гениальность и т. д. являются специфическими параметрами человеческой индивидуальности. Личность как социокультурная характеристика человека раскрывается в общественных науках через понятия «социальные качества», «социальный статус», «социальная роль», «социальный характер» и др. Человек как личность есть сознательная и ответственный субъект деятельности, активно преобразующий природу и самого себя.

Современный человек включен в сложную и разветвленную систему социальных коммуникаций. Коммуникация (от лат. «сообщение, передача») – процесс установления и развития контактов между людьми, порождаемый потребностями в совместной социальной деятельности. Проблема коммуникации занимала важное место в философии экзистенциализма [1].

Без преувеличения, коммуникацию можно считать необходимым и всеобщим условием жизнедеятельности человека и одной из фундаментальных основ существования общества. Общество – не столько совокупность индивидов, сколько те связи и отношения, в которых данные индивиды находятся друг с другом [2].

Посредством социального общения человек осуществляет индивидуальный выбор и ранжирование социокультурных программ поведения, деятельности и общения. В процессах общения происходит формирование социальной позиции человека, его социального статуса и моделей поведения. В современном мире актуализируются проблемы использования новых коммуникативных технологий, в связи с чем возрастает значение философской аргументации как важнейшего инструмента формирования рационального характера человеческого общения (культурного, научного, политического и др.).

В условиях развития современных информационно-коммуникационных технологий на наших глазах формируется новый тип человеческой коммуникации – «сетевое общество», в котором еще в большей степени, чем в технотронном обществе утрачиваются непосредственные связи с реальным субъектом и реальным миром. В нем человек все в большей степени имеет дело с анонимными адресатами и ситуациями коммуникации, в которых стираются пространственно-временные координаты аргументатора и аудитории, превращая процесс живого общения в виртуальную коммуникацию [1].

Коммуникация обладает всеми необходимыми для этого свойствами: она является подлинно социальной (и единственной подлинно социальной) операцией. Она оказывается подлинно социальной, поскольку, хотя и предполагает некоторое множество соучаствующих систем сознания, но (именно поэтому) не может быть приписана никакому отдельному сознанию как единство [3, с.44].

Н. Луман полагает, что процесс коммуникации может быть осуществлен вне конкретного человека. «Не человек способен коммуницировать – коммуницировать способна лишь система» [3, с. 110]

Исследователи отмечают, что осмысление коммуникационной проблематики шло по определенным направлениям: англоамериканскому, направленному на лингвистический анализ и

«прояснение языкового опыта» (Л. Витгенштейн); французскому, не ограничивающемуся языковой коммуникацией, но включающему различные социальные проблемы коммуникации современного общества (осмысление феноменов идеологии и власти, критика капитализма, осмысление дискурсивных практик); «философией диалога» (М. Бубер, Э. Левинас, М. Бахтин, Ф. Розенцвейг, Ф. Эбнер, М. Библер и др.); немецкой школой коммуникативистики (изучение различных средств коммуникации: газетного дела, книг, фильмов, радио и пр.; специфики политической коммуникации) [4].

Общество, следовательно, понимается, прежде всего, как система, а форма системы, как говорилось выше, представляет собой всего лишь различие системы и окружающего мира. Но это не означает, что общей теории систем будет достаточно для того, чтобы в ходе логической процедуры суметь раскрыть то, что нам представляется обществом. Помимо нее должны получить определение и особенности социальных систем, а затем - внутри теории социальных систем - следует выявить то, что составляет особенность системы общества, а значит, те следствия, которые влечет за собой обозначение нами общества как всеохватывающей социальной системы [3, с.43].

Каждое общество формирует свой стандарт личности. Социология общества определяет психологические типы данного общества.

Личность имеет многоуровневую организацию. Высший и ведущий уровень психологической организации личности - ее потребностно-мотивационная сфера - это направленность личности, ее отношение к обществу, личности, к самой себе и своим трудовым обязанностям. Для человека существенно не только его положение, но и способность реализовать свои отношения. Это зависит от уровня развития деятельностных возможностей человека, его способностей, знаний и умений, его эмоционально-волевых и интеллектуальных качеств.

Человек не рождается с готовыми способностями, характером и т. д. Эти свойства формируются в течение жизни, а на определенной природной основе. Наследственная основа организма человека (генотип) определяет его анатомо-физиологические особенности, основные качества нервной системы, динамику нервных процессов. В биологической организации человека, его природе заложены возможности его психического развития. Но человек становится человеком только через овладение опытом предшествующих поколений, закрепленным в знаниях, традициях, предметах материальной и духовной культуры. Природная сторона человека не должна противопоставляться его социальной сущности. Сама природа человека есть не только продукт биологической эволюции, но и продукт истории. Биологическое в человеке нельзя понимать как наличие в нем какой-то «животной» стороны. Все естественные биологические задатки человека являются человеческими, а не животными задатками. Но формирование человека как личности происходит только в определенных социальных условиях.

Формирование и развитие личности происходит в процессе социализации (освоения и усвоения норм жизни в обществе), воспитания и социальной деятельности. Важнейшие черты личности: социальность, активность, деятельность, креативность, ответственность и др [1].

Особенностью личности является ее замкнутость. Сознание своей замкнутости позволяет личности быть свободной от произвольных преходящих социальных институтов, диктата власти, не терять самообладания в условиях социальной дестабилизации и тоталитарных репрессий. Автономность личности связана с ее высшим психическим качеством - духовностью. Духовность есть высшее проявление сущности человека, его внутренней приверженности человеку, нравственного долга, подчинения высшему смыслу бытия. Одухотворенность личности выражается в ее сверхсознании, потребности в стойком отвержении всего низменного, беззаветной преданности высоким идеалам, обособленности от недостойных побуждений, сиюминутном престиже и псевдообщественной активности. Но чем примитивнее общество, чем сильнее его тенденция к всеобщей уравниловке, тем больше в нем людей, слепо подчиняющихся требуемым нормам. Человек, говорящий готовыми лозунгами, перестает заботиться о своем личном самостроении.

Качества личности определяются кругом ее практических отношений, вовлеченностью в различные сферы жизни общества. Творческий человек выходит за пределы ближайшего социального окружения, формирует себя на более широкой социальной базе. Перспектива общества может проявляться в личности. Он может олицетворять будущее общество, опережая его нынешнее состояние. Обособленность личности означает ее независимость от узких рамок замкнутой группы, это показатель развития личности[5].

Список использованных источников:

1. *Малыхина, Г. И. Философия и методология науки : учеб. пособие / Г. И. Малыхина, В. И. Чушов, В. И. Миськевич. – Минск : БГУИР, 2017. – 274 с.*
2. *Василик, М. А. Основы теории коммуникации / М. А. Василик. – М., 2006.*
3. *Луман, Н. Общество как социальная система / Н. Луман. – М. : Логос, 2004. – 232 с.*
4. *Назарчук, А. В. Теория коммуникации в современной философии / А. В. Назарчук. — М., 2009.*
5. *Еникеев, М.И. Личность и общество [Электронный ресурс]. – Режим доступа: <https://psyera.ru/2886/lichnost-i-obshchestvo> – Дата доступа: 10.03.2023.*

UDC

**HUMAN IN THE SYSTEM OF SOCIAL COMMUNICATIONS.
INDIVIDUAL AND SOCIETY**

Nguyen V.T.A., master student gr. 215101

Belarusian State University of Informatics and Radioelectronics,

Minsk, Republic of Belarus

*Miskevich V.I. – Candidate of Philos. sciences,
associate professor*

Annotation. The thesis considers a person in the variety of his characteristics in the system of social communications, personality, society and their relationship.

Keywords: Man, communication, personality, society

АРГУМЕНТАЦИЯ И ЕЕ РОЛЬ В НАУКЕ И ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Никанав М.Ю.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Малыхина Г.И.. – канд. филос. наук, доцент, зав. кафедрой философии

Аннотация. Рассматривается история возникновения аргументации. Анализируются аспекты изучения аргументации. Акцентируется внимание на роли аргументации в науке и профессиональной деятельности в условиях современного общества.

Введение. Учение об аргументации возникло еще в древней Греции, где активно развивались различные формы государственного управления, судебной и политической практики. Ученые того времени начали анализировать логику, психологию и этику различных форм убеждения, а также выделять наиболее эффективные методы ведения споров и дискуссий, которые были широко распространены в греческом государстве, особенно в Афинах. Умение использовать убедительные аргументы являлось гарантией победы в суде, политической карьеры и социальной выгоды.

Основная часть. В современной логической литературе понятие аргументация определяется как интеллектуально-речевая деятельность. Аргументация - это приведение доводов с целью изменения позиции или убеждения другой стороны[1].

При изучении аргументации обычно выделяют два аспекта - логический и коммуникативный. В логическом аспекте основная задача аргументации заключается в обосновании какого-то утверждения, позиции или формулировки через использование других утверждений, называемых "аргументами". Если говорить в целом об эффективной аргументации, то при этом реализуется также и коммуникативный аспект, в случае если собеседник соглашается с аргументами и способом доказательства или опровержения (т.е. обоснования) исходного утверждения (т.е. тезиса).

В настоящее время аргументация является отдельной областью междисциплинарных исследований. Современная теория аргументации - это комплексная дисциплина, которая объединяет в себе целый ряд наук, занимающихся изучением человеческой коммуникации и познания, включая философию, логику, историю, социологию, лингвистику, психологию и другие. Наиболее интересными направлениями развития аргументации являются новая риторика Тулмина и Перельмана, а также интеррогативная концепция диалога Хинтикки.

В науке аргументация является ключевым инструментом для развития новых теорий и исследований. Научные исследования должны быть основаны на доказательствах и аргументах, чтобы обеспечить достоверность их результатов. В процессе научного исследования, ученые используют аргументацию, чтобы доказать свои гипотезы, сделать выводы и развивать новые теории. Аргументация также помогает ученым дискутировать и обмениваться мнениями, что способствует развитию научных знаний и открытий.

В профессиональной деятельности аргументация также является важным инструментом. Независимо от того, работаете ли вы в области бизнеса, права, медицины или других профессий, способность аргументировать свои идеи и убеждения является критической для успешного выполнения задач. Аргументация помогает развивать более эффективные решения, принимать обоснованные решения, убеждать и влиять на других людей.

Однако, для того чтобы аргументация была эффективной, она должна быть основана на доказательствах и логических аргументах. Это означает, что необходимо проводить исследования, собирать и анализировать данные, проводить эксперименты и анализировать результаты. Только после того, как вы получите достаточно доказательств и информации, вы можете начать строить свою аргументацию.

Также важно помнить, что аргументация должна быть обоснована научными принципами и этическими нормами. Не следует использовать аргументацию для манипуляции другими людьми или принимать решения, которые могут нанести вред другим.

Заключение. Аргументация играет ключевую роль как в науке, так и в профессиональной деятельности. Она позволяет обосновывать и защищать свои идеи, принимать обоснованные решения и находить компромиссы в диалоге с другими людьми. Особенно важно умение аргументировать в условиях современного информационного общества, когда количество информации растет в геометрической прогрессии, и критическое мышление и умение обосновывать свою точку зрения становятся необходимыми навыками.

Список использованных источников:

1. Малыхина, Г. И. Логика: учебник / Г. И. Малыхина. – Минск :Вышэйшая школа, 2013. – 335 с.

РОЛЬ ФИЛОСОФИИ В ФОРМИРОВАНИИ ЛИЧНОСТИ И СОВРЕМЕННОГО НАУЧНОГО МЫШЛЕНИЯ

Никанав М.Ю.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Малыхина Г.И. – канд. филос. наук, доцент, зав. кафедрой философии

Аннотация. Рассматривается роль философии в жизни современного общества. Акцентируется внимание на роли философии в формировании личности и современного научного мышления.

Введение. Философия - это наука, которая занимается поиском ответов на самые сложные вопросы, касающиеся смысла жизни и сущности мира. В процессе своего развития философия формировала и формирует общие принципы, которые используются в различных областях человеческой деятельности. В данной статье рассматривается роль философии в формировании личности и современного научного мышления.

Основная часть. Роль философии в формировании личности, определяется, прежде всего, тем, что она выступает в качестве теоретической основы мировоззрения, а также тем, что она решает проблему познаваемости мира, наконец, вопросы ориентации человека в мире культуры, в мире духовных ценностей[1].

Наука - это область знаний, которая стремится объяснить природу нашего мира и открыть новые знания. Философия помогает ученым задавать важные вопросы о том, как мы можем получить знания, и какие методы науки являются эффективными. Она также помогает ученым определить основные принципы, которые лежат в основе науки, такие как обоснованность и доказуемость. Философия помогает ученым анализировать социальные, этические и политические аспекты научных исследований.

Современное научное мышление тесно связано с философией. Многие научные теории и идеи базируются на философских концепциях, которые были разработаны еще в древности. Например, концепция материи и формы, разработанная Аристотелем, была использована в физике И.Ньютона. Концепция духа и материи, разработанная Декартом, была использована в философии и в науке. Также философия играет важную роль в формировании методологии научного исследования. Философия помогает установить правильный подход к научному исследованию, определить цели и методы исследования, а также формировать и обосновывать полученные результаты.

В современном мире философия является неотъемлемой частью нашей жизни. Она помогает людям понять смысл жизни и различные явления окружающей действительности. Философия обучает людей логическому мышлению и помогает им стать аналитически мыслящими. Эти навыки помогают не только в научной работе, но и в повседневной жизни.

Кроме того, философия способствует интеллектуальному развитию человека, в том числе и научному. С помощью философии мы можем понять логику, которой следует научное исследование и осознать его место в обществе. Философия также помогает ученым понимать, как различные научные дисциплины взаимодействуют между собой, и как они могут взаимно дополнять друг друга.

Важной ролью философии в современном обществе является создание критического взгляда на окружающую действительность. С помощью философского анализа мы можем выявлять логические противоречия, ложные утверждения и скрытые предположения в рассуждениях оппонентов, их возможные проявления в общественной жизни. Философия помогает формировать критическое отношение к медиа и социальным сетям, которые часто далеки от достоверности, и нередко спекулируют на ее пассивном восприятии при помощи софизмов и паралогизмов.

Однако, философия не только помогает науке, но и сама является формой научного знания. Философия позволяет нам исследовать теории и концепции, которые лежат в основе наших знаний, и анализировать, как они взаимодействуют друг с другом. Это позволяет ученым развивать новые теории и концепции, которые могут привести к новым открытиям и достижениям.

Заключение. Философия играет важную роль в формировании личности и современного научного мышления. Она помогает человеку развивать критическое мышление и аналитические способности, формировать свое мировоззрение и определять свои ценности.

Список использованных источников:

1. Роль и значение философии в жизни общества [Электронный ресурс]. – Режим доступа: https://minkultrd.ru/news/news_list/Rol_i_znachenie_filosofii_v_zhizni_obschestva - Дата доступа : 12.03.2023.

НАУКА КАК ВАЖНЕЙШАЯ ФОРМА ПОЗНАНИЯ. ФУНКЦИИ НАУКИ В СОВРЕМЕННОМ МИРЕ

Никитина В.А.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Малыхина Г.И. – канд. филос. наук, доцент, зав. кафедрой философии

Аннотация. Рассмотрена наука как важнейшая форма человеческого познания. Акцентировано особое внимание на особенности и отличительные черты научного знания. Определены функции науки в современном обществе и их влияние на качество жизни человека.

Введение. Культура любого общества в значительной степени зависит от его знаний о реальности. Рациональные знания являются ответами на вопросы о сущности явлений, причинах их возникновения, изменения и функционирования, а также об их возможном использовании и преобразовании. Человечество накопило многовековой опыт практического познания мира, который позволяет отвечать на различные вопросы об окружающем мире и его процессах.

Основная часть. Наука – это важнейшая форма познания, которая является основой современного мира. Она представляет собой систему знаний, которые обеспечивают высокую степень достоверности и объективности в познании мира. Наука является источником научных знаний и своей главной целью ставит их производство. Важно отметить, что эти знания отличаются от житейских, обыденных знаний. Последние, по большей части, имеют индуктивный характер, а научные знания являются результатом научно-познавательной деятельности и обладают рядом характеристик: объективность, обоснованность, системность, рациональность, универсальность, формально-логическая непротиворечивость и др. Помимо этого, научные знания отличаются от мифологических знаний.

В то же время наука может быть определена не только как система научных знаний, но и как специализированный вид социальной деятельности по их производству. Этим занимаются ученые в различных областях науки: логико-математических, естественно-научных, социально-гуманитарных, технических.

Согласно пониманию науки как социокультурного феномена и важнейшей формы культуры, она выполняет ряд функций, таких как: когнитивно-познавательная, культурно-мировоззренческая, инновационно-практическая, духовно-образовательная, социально-управленческая, прогностическая и др.

Основной из них является познавательная функция, которая состоит в поиске и производстве новых знаний и законов об объективной реальности. Культурно-мировоззренческая функция выражается в способности науки влиять на мировоззрение людей и менять их представление об окружающей действительности. Инновационно-практическая связана с переустройством мира в области научно-технического прогресса, результатом которой является появление индустриального общества с постоянно развивающимися и улучшающимися техническим оснащением и организацией рациональных форм труда, управленческих методов. Духовно-образовательная функция заключена во влиянии на культурные сферы человеческой жизни: за основу в современных образовательных программах взяты именно научные знания, а на протяжении многих веков создавались новые этические и правовые концепции, а также концепции справедливого общества. Социально-управленческая функция обеспечивает анализ социальных проблем и возможностей их решения, проектирование моделей социального управления. Прогностическая функция заключается в возможности предсказывать развитие событий будущего на основе знаний о различных процессах. [1]

Наука обладает особым статусом в современной культуре и тесно взаимодействует с другими формами познания, что имеет практическую значимость для человека. Специфика науки позволяет внедрять научные методы в процессы управления культурой и другими областями.

Заключение. Таким образом, наука является важнейшей формой познания и выполняет ряд социальных функций в современном мире. Наука не ограничивается только технологическими разработками и экономическими выгодами. Она также является важным фактором социальной стабильности и прогресса. Научные знания способствуют формированию рационального мышления, критического взгляда на окружающую действительность, что позволяет людям более глубоко понимать социальные, политические и экономические процессы, что способствует прогрессивному развитию общества.

Список использованных источников:

Конференция «Компьютерные системы и сети»

1. *Малыхина, Г. И. Философия и методология науки: учеб. пособие / Г. И. Малыхина., В.И. Чуешов, В.И. Миськевич//Минск: БГУИР, 2017. – 274с*

ЦИФРОВОЕ ГИПЕРПОТРЕБЛЕНИЕ И ЕГО ВЛИЯНИЕ НА ПСИХИКУ СОВРЕМЕННОГО ЧЕЛОВЕКА

Петров В.В., студент гр.150501

*Белорусский государственный университет информатики и радиоэлектроники 1
г. Минск, Республика Беларусь*

Амоненко С. А. – м. ф. н.

Аннотация. В современном цифровом мире с каждым днём всё более становится актуальна проблема деградации мышления. Обширное количество контента и его примитивность формирует условно-рефлекторную положительную связь, в следствии чего большинство людей потребляют огромные массивы данных, которые они не то что не анализируют, а вовсе "свайпают", не задумываясь. В статье поднимается вопрос, как это влияет на психику современного человека и к чему это нас приведет.

Ключевые слова. цифровое гиперпотребление, цифровое слабоумие, информационная псевдобебильность.

Современный мир предоставляет множество возможностей для потребления информации, которые тяжело было даже вообразить столетие назад. Однако данное явление обладает как очевидными положительными последствиями, так и более скрытыми негативными следствиями для человеческой психики. Так, например, с недавнего времени в тренде стал огромный сервис с контентом под названием "TikTok". Данный сервис служит для просмотра исключительно коротких видео. Идея крайне проста, оттого и гениальна, в том смысле, что на сегодняшний день трафик в данном сервисе превышал все рекорды и его смог перебить только ChatGPT.

Но почему и откуда возникает так много людей, жаждущих просто посмотреть обычные и короткие видео? Рассмотрим эксперимент нейробиолога из Кембриджского университета Вольфрама Шульца.

«Эксперименты Шульца с обезьянами были основаны на ранних экспериментах Павлова, посвященных выработке условных рефлексов у собак. Шульц проигрывал обезьянам громкий звук, ждал несколько секунд, а затем выдавливал им в рот несколько капель яблочного сока. В ходе эксперимента Шульц наблюдал за электрической активностью отдельных производящих дофамин нейронов в мозге обезьян. В первое время нейроны не возбуждались, пока животное не получало сок. Однако как только обезьяны поняли, что звук предвещает получение сока, те же самые нейроны стали возбуждаться, когда раздавался звук, то есть в момент, предвещающий награду, а не после ее получения. По мнению Шульца, дофаминовая система обучения интересна тем, что в ней все основано на предвкушении.

Предвкушение награды помогает нам формировать привычки. Хорошие, адаптивные привычки помогают нам выживать, позволяя выполнять множество важных задач автоматически, даже не думая о них. Адаптивные привычки поощряются высвобождением дофамина в префронтальную кору и полосатое тело, то есть те области мозга, которые задействованы в контроле поведения, вознаграждении и мотивации. Высвобождение дофамина не только дарит человеку чувство удовольствия, но и вырабатывает у него условный рефлекс. Как мы знаем, обусловливание формирует долговременные воспоминания, которые позволяют нам узнавать стимул при следующей встрече с ним и реагировать соответствующе. Если это позитивный стимул, как бывает в случае с адаптивными привычками, обусловливание мотивирует нас стремиться к нему. Например, если вы съели банан и сочли его вкусным, в следующий раз, увидев банан, вы почувствуете желание его съесть» [1, с. 255-256].

Видео из сервиса "TikTok" работают точно по такому же принципу, то есть схема следующая:

1. Предвещение награды.
2. Выработка дофамина.
3. Её получение.

Так и происходит эскалация примитивного контента: наш мозг при выборе более сложной и лёгкой задачи выбирает простую. Условно говоря, если у пользователя стоит выбор: прочесть умную статью или посмотреть смешные картинки, естественным образом он выберет второе. Можно утверждать, что многие пользователи делают закладки умных статей и после читают, однако большинство людей возвращается лишь к 4-ом процентам отложенных вещей.

Как следствие, современные рекламодатели делают максимально примитивные вещи для более широкого охвата фокуса потребителей, т.к. внимание человека стоит миллиарды долларов на данный момент. В связи с этим, одним из наиболее интересных и проблемных кейсов, становится случай детей, которые с раннего возраста проводят колоссальное время в мобильном телефоне.

Мозг ребёнка пластичен и способен более хорошо усваивать, генерировать уникальные интеллектуальные модели. Для этого ему необходимо развиваться и у многих родителей возникает иллюзия, что гаджет поможет им в данном начинании и будет развивать мозг ребёнка.

Также есть важное отличие: мозг ребёнка должен настроиться на соответствующую конфигурацию, он крайне сильно отличается от мозга взрослого человека.

Для соответствующей настройки определённых областей, он должен взаимодействовать с объектами из реального мира, то есть иметь тактильный контакт, а не полностью работать с абстракциями. Если мозг полностью работает конкретно с абстракцией, это не становится какой-то частью интеллектуального багажа, а лишь проходит мимо него, не оставляя ничего в совокупности.

Может показаться, что эти абстракции никак не влияют на мозг и лишь пролетают мимо. Но современные сервисы генерируют огромное количество контента, вследствие чего дети теряют концентрацию внимания и не способны его сфокусировать на определённых вещах: прочесть книгу, выполнить домашнее задание и т.д.

Вернёмся ко времени, когда только начали появляться телевизоры и дети начали смотреть различные каналы. В то время провели исследование в США. В качестве исследуемых были дети, которые смотрели телевизор больше часа и менее часа. По данному исследованию получили следующий результат: 50% детей, которые смотрели телевизор менее часа поступили в вуз, а из другой группы лишь 10% поступили в вуз.

Возьмём данные из современного мира:

В среднем семилетний ребёнок год своей жизни провёл за экраном. Восемнадцатилетний молодой человек в среднем провёл 4 года за экраном. В среднем подростки проводят 8 часов за экраном.

У людей также возникает иллюзия, что растут талантливые, креативные дети, которые способны порождать что-то уникальное. На этот случай также есть тест на креативность Торренса. Он является более объективным, т.к. охватывает различные области мозга, а также может предсказать перспективы человека в будущем.

Результаты данного теста проводились с 1960 года.

С 1960 до 1984 года, не было такого массивного объёма информации, следовательно, и мышление подростков было другим. Результат – рост креативности.

С 1984 до 1990 года, стабилизировался рост креативности.

С 1990 года идёт неуклонное снижение выполнения теста.

Исследовательница, которая проверила более 300 тысяч отчётов добавила следующий комментарий относительно результатов теста: «Дети стали менее эмоциональными, менее вербально-активными, обладают меньшим чувством юмора и воображением.»[2, p. 291]

Интересные данные по Facebook из 2015 года:

34% ложилось спать вместе с Facebook.

21% просыпался ночью, чтобы проверить, что у него происходит в Facebook.

27% пользователей пользовались им, сидя в туалете.

Цифровая зависимость является медицинским диагнозом.

Недавние исследования мозга, сделанные на тему изображений мозга, показывают, что цифровые приборы влияют на кору передних долей мозга (которая контролирует организацию действий) так же, как кокаин. Технология настолько сверхвозбуждает, что повышает уровень дофамина – нейромедиатор, наиболее связанный с зависимостью. На МРТ (магнитно-резонансная терапия) у Интернет-зависимых появились изменения в белом веществе мозга (меньше связей нервных клеток в частях мозга, которые занимаются эмоциями, процессом принятия решений и самоконтролем), также были найдены изменения в сером веществе мозга, которое было снижено. [3]

Также уровень кортизола (гормон стресса) растёт, если у человека с подобным диагнозом отобрать гаджет. Примечательным также является то, что подобные люди не могут отключить уведомления, а при их отключении, уровень кортизола также растёт.

Для борьбы с цифровой зависимостью современная наука изобретает различные симуляторы айфона, чтобы человек мог справляться с уровнем кортизола и уходить от возникшей зависимости.

Отсюда вытекает другое заболевание: информационная псевдодебильность.

Основоположником термина «информационная псевдодебильность» стал врач-психотерапевт Андрей Курпатов[4]. Занимаясь вопросами возникновения и исследуя такие состояния человека как вегетососудистая дистония, невроз, панические атаки, он выяснил, что одной из основных причин невротических нарушений является агрессивная атака на мозг большого количества разрозненной информации.

В процессе исследования данных проблем Курпатов на основании исследования групп пациентов, страдавших шизофренией в разные исторические периоды, ввел в психологию термин информационная псевдодебильность. При анализе Курпатов создал определённые временные рамки для этого термина: для больших 60-70 годов прошлого столетия характерен

сложноорганизованный бред, в конце 90-х произошел перелом в проявлениях этого психического заболевания. Этот феномен совпал со временем компьютерного бума, массового использования интернета. Для людей, заболевших на рубеже 20 века, характерны более примитивные бредовые идеи с преобладанием бытовой тематики.

В конце 20 века—начале 21 века информационная псевдодебильность претерпела патоморфоз из-за нахождения нашего мозга в постоянном информационном поле. Патоморфоз заключается в сложности выявления структуры бреда, проявляющейся через отношения в информационном поле. Больные не только не понимают, что находятся в бреде, но также не особо придают этому бреду значения. А. Н. Алехин отмечает, что структура нарушена в интеллектуальных объектах, которые строит наш мозг[5]. Выражается это нарушение в пассивности, импульсивности действий, отсутствии достаточного внимания, непонимании абстрактных рассуждений, утилитарном и конкретном мышлении.

Основным отличием обычной дебильности от информационной псевдодебильности является отсутствие у человека с дебильностью возможности построения более сложных интеллектуальных объектов в силу недостаточности серого вещества.

У человека с информационной псевдодебильностью количество серого вещества в норме и его мозг можно реабилитировать, но у него нет на это мотивации из-за постоянного необдуманного потребления информации.

По причине огромного количества информации, получаемого обществом посредством гаджетов, страдает конвергентное мышление, которое отвечает за принятие рационального решения. Это значит, что мозг получает информацию, но не анализирует, не осознает, не структурирует ее. Таким образом, мозг теряет способность мыслить абстрактно и переходит к упрощенному, клиповому мышлению.

Понятие информационного псевдодебилизма обладает схожими чертами с термином цифровое слабоумие. Между тем, два понятия не обозначают одно и то же. Цифровое слабоумие — полноценная болезнь, выражающаяся в таких синдромах как дефицит внимания, патологическая гиперактивность и встречающаяся в основном у детей. Эта болезнь обладает приобретенным характером и обусловлена проблемами формирования детского мозга. Цифровое слабоумие в отличие от информационной псевдодебильности неизлечимо.

В случае последовательного и системного ограничения информации можно вернуть способность к качественному мышлению, хотя в условиях реалий современного мира сделать это сложно. Процесс выхода из состояния информационной псевдодебильности усложняется также тем фактом, что человек не понимает своей зависимости от чрезмерного потребления информации.

Эти данные позволяют нам связать информацию, полученную с помощью теста Торренса, и явление патоморфоза шизофрении. Необходимо понимать, что у бреда шизофреников, как и у мысли обычного человека имеется своя структура. В обоих случаях мы наблюдаем ухудшение мысленного процесса и мышления, в общем и целом. Почему так происходит? Мышление, в первую очередь, это навык создания сложных интеллектуальных объектов. Но как оно может развиваться, когда в обществе происходит эскалация примитивного контента?

Одним из примеров негативных феноменов, вызванных процессом гиперпотребления информации, является “иллюзия знания”. Мозг работает по странному принципу: до всего до чего человек может дотянуться, находится в его поле влияния. Если обезьяне дать лопатку, то поле её влияния увеличивается и она вводит это в схему своего тела. Так и знания, которые не принадлежат человеку, он интегрирует в себя и ошибочно считает, что знает многое. Многие задачи человек решает при помощи уже созданных для этого устройств: навигаторы, поисковики и т.д. Это приводит к отсутствию необходимости мышления и анализа информации, что влечет за собой деградацию человеческого интеллекта.

С этим связана и проблема заглушения префронтальной коры мозга современного человека, при потреблении информации. Существуют исследования, в которых человека клали в фМРТ и анализировали, какие зоны мозга у него активизируются в случае, когда он принимает решения самостоятельно и в случае, когда он принимает решение с наличием мнения эксперта. Проблема состоит в том, что во втором случае, у людей не активизировалась зона принятия решения: префронтальная кора головного мозга. В эксперименте она активизировалась только в том случае, когда была соответствующая надпись: “мнение экспертов недоступно”. Так и построены современные рекомендательные сервисы – префронтальная кора головного мозга просто не включается и человек становится ведомым и вовсе отказывается думать.

Хорошим примером изменения формы восприятия информации, служит динамика развития современных социальных сетей. Изначально существовала блог-платформа LiveJournal, в котором писали объёмные тексты. Далее Facebook и ВКонтакте, где тексты сократились, после и Twitter, где всего имеется 140 символов, а потом Instagram, где нет как такового текста, лишь картинки. Такая “эволюция” подачи текстового контента происходит не случайно, она является следствием того, как функционирует современный информационный мир и человеческая психика.

Характерной чертой сегодняшнего чтения текста с экрана является F-скроллинг. Человек читает начало, пытается разобраться о чём говорится в тексте, после переходит к началу последующих нескольких абзацев, потом пролистывает его полностью и пытается найти смысл, и само собой не находит. Подобные тексты кажутся вялыми и неинтересными человеку, так как есть другие более динамичные сервисы по производству контента, в результате само чтение становится фактически вытеснено.

Все это приводит нас к тому, что современного человека нет времени, чтобы думать. Он постоянно находится в сети, у него имеется неограниченное количество новостей: уведомления, мессенджеры, социальные сети и т.д. Время человека тратится на процесс организации интеллектуальной работы, а не на саму интеллектуальную работу. Например, мозг современного офисного сотрудника вообще не может сконцентрироваться на определённой задаче, т.к. для концентрации на задаче мозгу необходимо 23 минуты, а время появления новых задач занимает пару минут.

Таким образом, современный цифровой мир генерирует следующую картину для психики человека:

1. Упрощение знаний.
2. Отсутствие ощущения будущего.
3. Наличие рекомендательных сервисов, блокирующих интеллектуальную деятельность.
4. Избыток раздражителей, которые не дают сфокусироваться на определённой интеллектуальной задаче.

Однако, на мой взгляд, главным отрицательным следствием гиперпотребления информации является отсутствие построения сложных взаимоотношений между человеком и информацией, и между самими людьми. За долгие годы до построения интернета выстраивание отношений между людьми было весьма сложным интеллектуальным процессом. Чтобы найти соответствующую информацию или получить знания, необходимо было решать непростые задачи. Однако сейчас это не играет большой роли, поскольку большинство задач за нас решает интернет, что позволяет в принципе игнорировать других людей, рассматриваемых как ненужных для решения задач повседневной жизнедеятельности. Это привело к цифровой аутизации человека и значительному уменьшению социальных взаимодействий.

Список использованных источников:

1. Кандель, Э. *Расстроенная психика. Что рассказывает о нас необычный мозг.* / Э. Кандель. – М.: Corpus, 2018. – 380 с.
2. Bonnie Cramond, Deborah Bandalos. *A Report on the 40 Year Follow-Up of the Torrance Tests of Creative Thinking: Alive and Well in the New Millennium* / Cramond B., Bandalos D. – *Gifted Child Quarterly*. – 2005 – N 49(4) – 283-291 p.
3. *Цифровая зависимость [Электронный ресурс]. – Режим доступа: <https://tarkvanem.ee/ru/cifrovaj-mir/cifrovaja-zavisimost>. – Дата доступа: 26.03.2023*
4. А. Курпатов. *Информационная псевдодобильность [Электронный ресурс]. – Режим доступа: <https://www.brain-games.ru/blogs/avkurpatov/andrey-kurpatov-informatsionnaya-psevdodebilnost/> – Дата доступа: 28.03.2023*
5. Алехин А.Н., Грекова А.А. *Особенности формирования мышления в условиях цифровой среды* / А. Н. Алехин, А. А. Грекова – *Клиническая и специальная психология*– 2019 – Т. 8. №. 1. – с. 162–176

ПРОБЛЕМЫ СОЦИАЛИЗАЦИИ ЛИЧНОСТИ В ЦИФРОВОМ МИРЕ. ЛИЧНОСТЬ И МАССОВАЯ КУЛЬТУРА.

Санец А.В. магистрант гр.217101

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

*Научный руководитель: Малыхина Г.И. – Зав. кафедрой философии, канд.
философских наук, доцент.*

Аннотация. В настоящее время социализация личности происходит в условиях становления цифрового мира, что приводит к ряду проблем. В данной статье рассматриваются основные проблемы социализации личности в цифровом мире и связь между личностью и массовой культурой.

Введение. Социализация личности – это процесс адаптации личности к социуму, в котором она живет. В различные эпохи социализация личности происходит в различных социально-исторических условиях [1]. Сегодня этот процесс происходит в контексте становления цифрового мира. Так как цифровой мир является новой реальностью для человека, то социализация в нем может приобретать новые черты и привести к ряду проблем.

Основная часть. Одной из основных проблем социализации личности в цифровом мире является изоляция от физической реальности. Люди, которые проводят много времени в интернете, зачастую теряют навыки общения в реальной жизни. Некоторые люди могут проводить в интернете большую часть своей жизни, не встречаясь с живыми людьми. Это может привести к социальной изоляции и, в конечном итоге, к психическим проблемам и расстройствам [2].

Еще одной проблемой социализации личности в цифровом мире является информационное перенасыщение. Молодежь, «живущая» в интернете, часто сталкивается с большим количеством информации. Её потребители нередко становятся зависимыми от интернета, постоянно проверяя свои социальные сети и электронную почту. Это так же может привести к стрессу и ухудшению здоровья пользователей [3].

Другим препятствием на пути успешной социализации является проблема низкой самооценки. Молодые люди, для которых интернет становится своего рода «зеркалом жизни», зачастую сравнивают себя с другими людьми, которые выкладывают успехи своей жизни в интернете. В отдельных случаях это также может привести к низкой самооценке и депрессии [4].

Связь между личностью и массовой культурой также важна для понимания проблем социализации личности в цифровом мире. Массовая культура создает некие стереотипы и идеалы, которым люди стремятся соответствовать. При этом в интернете люди могут создавать свой образ и показывать себя такими, какими они хотят быть. В случае завышенных ожиданий это может привести к комплексу неполноценности и разочарованию. [5]

Заключение. Социализация личности в условиях цифрового мира является сложной задачей. Изоляция, информационный перегруз и низкая самооценка – это основные проблемы, с которыми сталкиваются люди, проводящие много времени в интернете. Связь между личностью и массовой культурой дополняет эту проблему, создавая нереальные ожидания и идеалы. Для решения этих проблем необходимо научить людей правильно использовать интернет и не забывать о реальной жизни.

Список использованных источников:

- [1] Социализация [Электронный ресурс]. – Режим доступа: <https://psihomed.com/sotsializatsiya/> (Дата обращения: 21.03.2022).
- [2] Online Social Networking and Mental Health [Электронный ресурс]. – Режим доступа: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4183915/> (Дата обращения: 24.03.2022).
- [3] Online social networking and addiction [Электронный ресурс]. – Режим доступа: <https://pubmed.ncbi.nlm.nih.gov/22016701/> (Дата обращения: 24.03.2022).
- [4] Как социальные сети могут повлиять на нашу самооценку [Электронный ресурс]. – Режим доступа: <https://med-heal.ru/archives/35329> (Дата обращения: 24.03.2022).
- [5] The Relationship Between Online Social Networking and Depression [Электронный ресурс]. – Режим доступа: <https://pubmed.ncbi.nlm.nih.gov/27732062/> (Дата обращения: 24.03.2022).

ПРЕДМЕТ И РОЛЬ ФИЛОСОФИИ В РАБОТАХ В.В. ОРЛОВА

Серяков Р.С., студент гр.142801

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Шкундич А.О – преподаватель

Аннотация. Автор рассматривает предмет и роль философии, её взаимозависимость с другими науками на основе работы В.В. Орлова. Диалектический материализм требует развития и углубления в соответствии с развитием частных наук. И результатом этого развития станет взаимозависимое сосуществование научной философии и частных наук.

Ключевые слова. Философия, позитивизм, диалектический материализм.

«Большинство из нас не думают об этом всё время, но, время от времени, почти каждый задумывается — зачем мы здесь? Откуда мы появились? Исторически, это считалось вопросами философии. Но философия мертва. Философы не поспевают за современными достижениями науки, в особенности физики. Теперь учёные приняли эстафету открытий в нашем квесте познания» [4]. Данная цитата С. Хокинга является иллюстративной, так как учёный и писатель репрезентирует реальность современного общества и отражает существующее общественное мнение о философии. Анализ цитаты даёт понять, что в настоящее время неясным остаётся статус научной философии. Это происходит из-за непонимания её предмета и отношения к частным наукам, однако ответы на данные вопросы не требуют изобретения чего-то нового, и уже были рассмотрены различными философами. Данная работа будет основана на исследованиях одного из таких философов, представителя Пермской университетской школы научной философии – В. В. Орлове. В своих работах В.В.Орлов чётко формулирует предмет научной философии и отношение этой философии и частных наук. Разбор его теоретического наследия позволяет как ответить на поставленные вопросы, так и вновь заявить о важности философии для научного сообщества.

Начать стоит с определения предмета философии, так как вопрос заключается в том, что изучает данная наука. Зачастую, в первом приближении, сравнивая философию с физикой, химией, биологией, историей и другими частными науками, предмет философии определяется как мир в целом и процесс его познания. Однако такая оценка философии содержит крупную неточность: мир как целое изучается системой наук, включая философию. Поэтому правильнее предмет философии определить как наиболее общие (всеобщие) стороны, свойства, законы мира в целом и его познания. В то время как частные науки имеют своим предметом какую-либо частную, особенную область действительности. Взаимозависимость философии и частных наук — одна из важнейших закономерностей развития науки. И её как раз можно определить с помощью философии, а именно с помощью философского категорического аппарата, так как в основе этой взаимозависимости лежит взаимосвязь всеобщего, которое является предметом философской науки, и особенного, которое является предметом частных наук. С точки зрения научной диалектики всеобщее существует только через особенное, во всей массе особенных, а особенное — только в связи с всеобщим. То есть предмет философии существует только через предмет частных наук, а вот предмет частных наук только в связи с предметом философии. Эта диалектика определяет глубокую зависимость философии от частнонаучного знания и, в свою очередь, не менее важную зависимость частных наук от философских знаний.

Разобрав предмет философии и её взаимозависимость с частными науками, становится понятно, что задача философа не сводится к самостоятельному изучению частных наук, философия не заменяет частные науки и не соревнуется с ними за право первенства, о чем говорит С. Хокинг. Задача философа — это формулирование понятий и концептуальных схем описания, всеобщих черт, свойств и законов реального мира, основывающихся на обобщении знаний полученных частными науками. Так, биология и история изучают реальные процессы эволюции жизни и развития человеческого общества, философия, всецело опираясь на биологическую теорию эволюции и историческую теорию, раскрывает всеобщую сущность и законы всякого развития, которые выходят за пределы возможности частнонаучного способа мышления. И потому требовать от философов знаний частнонаучных специалистов - бессмысленно, как и бессмысленно требовать от физика быть специалистом в научной философии. В.И. Ленин писал об этом следующим образом: «Ни единому из этих профессоров, способных давать самые ценные работы в специальных областях химии, истории, физики, нельзя верить ни в едином слове, раз речь заходит о философии. Почему? По той же причине, по которой ни единому профессору политической экономии, способному давать самые ценные работы в области фактических, специальных исследований, нельзя верить ни в одном слове, раз речь заходит об общей теории политической экономии. Ибо эта последняя - такая же партийная наука в современном обществе, как и гносеология» [2]. Но это не значит, что философы

могут уходить в пещеры, отрываясь от развития фундаментальных наук и общества, ибо философия, исходя из выведенной ранее зависимости, не имеет другого материала, для развития своих теорий, кроме совокупного материала, естественных, технических и общественных наук. Философское исследование, используя специфические философские формы абстракции, связанные с категориями философии, производит дальнейшую глубокую проработку эмпирического и теоретического материала науки. И в этом заключается зависимость философии от науки.

Но чтобы раскрыть зависимость науки от философии, необходимо рассмотреть господствующее в современной науке философское направление – позитивизм. Охарактеризовать его можно как философское направление, исходящее из тезиса о том, что всё подлинное, «положительное» знание может быть получено лишь как результат отдельных специальных наук или их синтетического объединения, и что философия как особая наука, претендующая на самостоятельное исследование реальности, не имеет права на существование. С. Хокинг рассчитывает получить знания о мире в целом простым сложением частных наук. Это можно сказать и об основоположнике позитивизма О. Конте утверждавшем, что каждая наука – сама себе философия. И тут мы снова можем обратиться к научной философии, а конкретно к разработанным в ней категориям части и целого.

Суть в том, что знание о мире в целом является органическим целым, то есть не сводимым к простой сумме своих частей и не тождественному ей. Целое характеризуется новым качеством и свойствами, не присущими отдельным частям, но возникающим в результате их взаимодействия в определённой системе связей. Частнонаучное описание, как часть целого, неполно и должно дополняться философским объяснением. Но раз мы говорим, что каждое частнонаучное решение крупных вопросов имеет философскую компоненту, то чем же руководствуются современные учёные сознательно или несознательно отказавшиеся от научной философии? Суть в том, что человек так или иначе использует мышление, чтобы прийти к своим специальным выводам, но мышление невозможно без категорий, и даже если они заявляют, что мыслят фактами, а не категориями – это лишь значит, что за этими фактами они не могут увидеть логических категорий, выработанных до них.

Ф. Энгельс писал об этом следующим образом: «Естествоиспытатели воображают, что они освобождаются от философии, когда игнорируют или бранят ее. Но так как они без мышления не могут двинуться ни на шаг, для мышления же необходимы логические категории, а эти категории они некритически заимствуют либо из обыденного общего сознания так называемых образованных людей, над которым господствуют остатки давно умерших философских систем, либо из крох прослушанных в обязательном порядке университетских курсов по философии (которые представляют собой не только отрывочные взгляды, но и мешанину из воззрений людей, принадлежащих к самым различным и по большей части к самым скверным школам), либо из некритического и несистематического чтения всякого рода философских произведений, - то в итоге они все-таки оказываются в подчинении у философии, но, к сожалению, по большей части самой скверной, и те, кто больше всех ругает философию, являются рабами как раз наихудших вульгаризированных остатков наихудших философских учений» [3].

Итак, разрабатывая теория и метод своей науки, учёный исходит из представлений об уже существующем материальном мире, пользуется уже выработанными категориями, являющимися итогом тысячелетнего развития. Поэтому игнорирование современной научной философии приводит к её стихийному принятию. Однако влияние философии не ограничивается ролью общей предпосылки частнонаучного знания. Применение научной философии даёт глубоко разработанную и хорошо структурированную картину мира и его познания, трактовку сущности мира и человека, чем обуславливает глубокое и органическое влияние философской мысли в структуру частнонаучного знания. Философские положения, сформулированные и обоснованные на базе материала частных наук, выступают в качестве аргументов теоретических положений частных наук. Каждая наука, имея дело с определенной выделенной областью реального мира, нуждается в более широком обосновании своих положений. Это возможно осуществить с помощью категорий или законов, которое даёт научная философия, обобщающая данные всей системы наук. В этом всё и заключается зависимость науки от философии.

Как итог, важно отметить, что, с точки зрения В.В. Орлова, изложенным выше представлениям о научной философии в наибольшей мере отвечает философия марксизма – диалектический материализм. Он отмечал, что серьезное научное содержание в определенных пределах имеется также в современной зарубежной философии. Однако нельзя не заметить, что все течения современной зарубежной философии чрезмерно сужают предмет философии, сводят его, например, к логике и языку науки (неопозитивизм), особенностям научного знания и научных революций (постпозитивизм) и т.д. Диалектический материализм же, являясь результатом развития естествознания и общества, на данный момент наиболее точно отражает современную нам действительность. Что не говорит о его законченности. Диалектический материализм, как и любая наука, требует развития и углубления в соответствии с развитием частных наук. И результатом этого

развития станет не господствующее положение научной философии над остальными науками, и не положение служанки для науки, а взаимозависимое сосуществование научной философии и частных наук.

Список использованных источников:

1. Основы философии. Общая философия / В.В Орлов. – Пермь: 5-е изд. 2012.
2. Полное собрание сочинений том 18 / В.И. Ульянов – Москва: Издательство политической литературы, 5-е изд. 1980. Стр. 363.
3. Диалектика природы / Ф. Энгельс - Москва: политиздат, 1969. Стр. 20.
4. Hawking, S. *Why Are We Here?* / S., Hawking – Google Zeitgeist conference, 2011. – Mode of access: https://www.youtube.com/watch?v=r4TO1iLZmcw&ab_channel=GoogleZeitgeist. – Date of access: 20.02.2023.

ОСОБЕННОСТИ И РАЗНООБРАЗИЕ СУБКУЛЬТУРЫ ГЕЙМЕРОВ В ИНФОРМАЦИОННОМ ОБЩЕСТВЕ

Слижёва А. В.

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Бархатков А. И. – канд. филос. наук, доцент

Аннотация. С постоянным расширением аудитории видеоигр растет и субкультура геймеров. В условиях формирования информационного общества геймеры оказывают на него значительное влияние, стимулируя развитие новых технологий и формируя новые типы сетевых сообществ. При этом субкультура геймеров внутренне неоднородна, и в ее рамках можно выделить несколько относительно устойчивых типов и разновидностей.

Гейминг стал неотъемлемой частью индустрии развлечений и значительно повлиял на информационное общество. Субкультура геймеров развилась в разнообразную группу людей, которые разделяют страсть к играм. В этом выступлении мне бы хотелось рассмотреть различные типы геймеров, существующих в этой субкультуре, и как они соотносятся с более крупным контекстом информационного общества

Субкультура геймеров имеет богатую историю, которую можно проследить до ранних дней видеоигр. Игры начали свою историю как хобби в 1970-х и 1980-х годах с появлением первых игровых консолей, таких как Atari 2600 и Nintendo Entertainment System (NES). Однако сама субкультура начала формироваться в 1990-х годах с появлением онлайн-игр и LAN-вечеринок.

С появлением онлайн-игр в 1990-х и 2000-х годах геймеры могли соединяться и соревноваться с другими игроками со всего мира. Онлайн-сообщества геймеров, такие, как форумы и социальные сети, предоставляли место, где геймеры могли обсуждать свое хобби и делиться советами и хитростями. Эти сообщества также помогли создать чувство идентичности и принадлежности для геймеров.

Одним из заметных аспектов геймерской субкультуры является ее фокус на навыках и соревновании. Геймеры часто гордятся своими игровыми способностями и стремятся улучшить их путем практики и тренировки. Этот упор на навыки привел к созданию соревновательных игровых мероприятий, таких, как турниры по киберспорту, которые становятся все более популярными в последние годы.

Что же касается информационного общества, то под ним понимается общество, которое развивается в основном благодаря широкому использованию информационных и коммуникационных технологий (ИКТ) для создания, обмена и доступа к информации. В информационном обществе знания и информация рассматриваются как ценные ресурсы, а способность к их эффективному использованию существенна для успеха в различных областях, таких, как образование, бизнес и научные исследования.

Субкультура игроков оказала значительное влияние на информационное общество в нескольких аспектах.

1. Субкультура игроков создала спрос на новые технологии и платформы, что стимулировало инновации и развитие новых цифровых технологий.
2. Игры помогли формированию онлайн-сообществ.
3. Игры повлияли на развитие новых технологий, таких, как виртуальная и дополненная реальность, которые активно применяются в других отраслях (здравоохранение, образование, тренинг).

Теперь, имея общее представление о геймерах и их основном вкладе в информационное общество, рассмотрим, на какие виды они делятся.

Первый тип геймеров - это любительские геймеры (casual gamer). Этот тип геймеров играет в игры для развлечения и не инвестирует значительное количество времени или денег в свой игровой опыт. Любительские геймеры могут наслаждаться мобильными играми или простыми онлайн-играми, которые не требуют большого уровня мастерства или знаний. Согласно опросу, проведенному Entertainment Software Association, 65% взрослых американцев играют в видеоигры, при этом 60% из них играют на своих смартфонах (ESA, 2021). Любительские геймеры являются важной частью игровой индустрии, поскольку представляют собой большую часть потребительской базы.

Второй тип геймеров - это hardcore геймеры (hardcore gamer). Этот тип геймеров инвестирует значительное количество времени и денег в свой игровой опыт и обладает высоким уровнем мастерства и знаний в области игр. Hardcore геймеры являются ведущей силой в игровой индустрии и способствуют развитию новых игр и технологий, предоставляя обратную связь и предложения разработчикам игр.

Третий тип игроков - профессиональные игроки (professional gamer). Этот тип игроков зарабатывает на жизнь, играя в игры на соревновательной основе или транслируя свою игру в онлайн-режиме. Профессиональные игроки высококвалифицированы и часто имеют большое количество подписчиков на социальных платформах, таких как Twitch и YouTube. Согласно Forbes, самым высокооплачиваемым игроком в e-sports в 2020 году был Йохан Сундштейн, который заработал \$6,9 млн, играя в Dota 2 на соревновательной основе (Forbes, 2020). Профессиональные игроки являются свидетельством потенциальных возможностей карьеры, которые существуют в игровой индустрии.

Четвертый тип игроков - ретро-игроки. Этот тип игроков любит играть в старые игры, которые были популярны в прошлом. Ретро-игроки часто коллекционируют старые консоли и игры и могут участвовать в онлайн-форумах или сообществах, посвященных ретро-играм. Ретро-гейминг становится все более популярным в последние годы благодаря выпуску обновленных версий классических игр и наличию эмуляторов, которые позволяют игрокам на новых консолях испытать старые игры.

Последний тип игроков - социальные игроки. Этот тип игроков любит играть в игры с друзьями или же на других мероприятиях, таких как вечеринки. Социальные игры становятся более доступными в последние годы благодаря возрастанию количества вечеринок и многопользовательских игр, которые можно играть в локальной сети или в Интернете. Социальные игры становятся важным аспектом игровой индустрии, и многие игры включают социальные функции, такие как чат внутри игры и многопользовательские режимы.

Субкультура геймеров является разнообразной и сложной группой людей, которые разделяют страсть к играм. Различные типы геймеров, существующие в этой субкультуре, играют важную роль в большем контексте информационного общества. Независимо от того, является ли это развлечением, соревнованием или карьерой, игры стали неотъемлемой частью нашего общества и будут продолжать формировать то, как мы взаимодействуем с технологиями и друг с другом.

БУДДИЗМ В СОВРЕМЕННОМ МИРЕ

Сончик Д.А.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ратникова И.М. – канд. филос. наук, доцент

Аннотация. Данная работа посвящена выяснению места буддизма в разных сферах жизни современного человека.

В данный момент человек сталкивается с новыми проблемами, которых раньше не существовало. Многие из них носят глобальный, всеобъемлющий характер. Во многом, основная причина их возникновения имеет социальную природу. Несмотря на популяризацию атеизма в XX в., религиозное возрождение является реакцией на глобализацию и вызванные ею кардинальные трансформации фактически во всех областях культуры. На данный момент буддизм перестал являться чисто восточной религией и относится к числу одной из мировых религий. Основная причина этому – ускоренный культурный обмен как одна из основных черт глобализации [1].

В нашей стране наибольшую популярность получили медитативные практики. При этом практикуются они не только в специальных местах, но и дома на ежедневной основе с помощью специальных приложений. На русскоязычном рынке получили широкое распространение школы медитации и курсы по духовному росту. Причина такой популярности буддизма – толерантность и демократизм: людям необязательно менять вероисповедание, чтобы перенимать отдельные элементы философии буддизма.

Важно отметить, что медитации – не единственный аспект, который получил распространение. Принципы буддизма используются во многих сферах жизни общества. К примеру, в сфере бизнеса, поскольку в условиях современных кризисных процессов в экономике духовные принципы буддизма оказываются как никогда актуальными. Так, Сигеру Кобаяси, топ менеджер корпорации «Сони», отмечает, что эффективное управление этой компанией основано на строгом следовании дзен-буддистскому принципу «Му». Ши Йонгшин, настоятель монастыря «Шаолин», стремится управлять монастырем как предприятием и в настоящее время владеет правами на 43 наименования товаров. Стив Джобс проявлял серьезный интерес к дзен-буддизму. Многие считают, что именно буддизм помог обрести Стиву Джобсу тот духовный стержень, благодаря которому ему удалось стойко пережить все те взлеты и невзгоды, с которыми он встретился на своем жизненном пути [2].

Причиной для пересмотра существующих ценностей послужил и глобальный экологический кризис, во многом связанный с идеологией потребительства, лежащей в основании иерархии доминирующей системы ценностей современного общества. Философия буддизма акцентирует внимание на важности проявления уважения к любым живым существам, а не только к человеку. В основе учения буддизма лежат такие понятия, как «карма», сопряженное с выражением экологического сознания, «ахимса» – принцип ненасилия, «бодхичитта» – доброта и сострадание ко всем живым существам. Ещё одним принципом, крайне важным для улучшения экологической ситуации, является принцип терпимости. Этот принцип можно применить к постоянно растущим потребностям человека. Следование духовным устремлениям буддизма, осознание собственной ответственности, сохранение чистого разума может положительно повлиять на человека, способствовать формированию стратегии осознанного потребления.

Кроме того, современный темп и образ жизни сопровождается увеличением числа психологических проблем. По данным ВОЗ, за последние несколько десятилетий заболеваемость неврозами выросла в 24 раза. В духовной же практике буддизма накоплены ценные методики, которые дают возможность адекватного восприятия мира и выбора позитивного образа жизни [3].

Не чужда буддизму и научная деятельность. Так, духовный лидер буддистов Далай-Лама XIV принимает участие во многих научных конференциях, посвященных когнитивным исследованиям. Современные ученые, в свою очередь, также проявляют большой интерес к буддийской психологии. Если говорить про развитие науки и образования в целом, то между западным и восточным подходами всегда существовали различия. Западная традиция ориентирована на познание и изменение внешнего мира. Буддийский же подход к науке и образованию основывается на взаимосвязи разных форм познания, гармонии человека и мира. Наука здесь не противопоставляется религии. Таким образом, духовные принципы буддизма делают его привлекательным в глазах студенчества, культурной и научно-технической интеллигенции.

Список использованных источников:

1. Уланов М. С. «Срединный путь»: буддийский ответ на вызовы глобализации» [Электронный ресурс] / М. Уланов. – Режим доступа: <https://cyberleninka.ru/article/n/sredinnyy-put-buddiyskiy-otvet-na-vyzovy-globalizatsii> . – Дата доступа: 02.04.23.

Конференция «Компьютерные системы и сети»

2. Уланов М. С. «Буддизм и современный бизнес» [Электронный ресурс] / М. Уланов. – Режим доступа: <https://cyberleninka.ru/article/n/buddizm-i-sovremennyy-biznes>. – Дата доступа: 02.04.23.

3. Доржигушаева О. В. «Экологическая этика буддизма» [Электронный ресурс] / О. Доржигушаева. – Режим доступа: <https://cyberleninka.ru/article/n/ekologicheskaya-etika-buddizma>. – Дата доступа: 02.04.23.

КИНЕМАТОГРАФ КАК ФАКТОР ПРИОБЩЕНИЯ К ИСТОРИКО-КУЛЬТУРНОМУ НАСЛЕДИЮ БЕЛАРУСИ

Трубач К.И.

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Ратникова И.М. – канд. филос. наук, доцент

Аннотация. Данная работа посвящена рассмотрению роли кинематографа в деле приобщения к историко-культурному наследию нашей страны.

Кинематограф является весьма популярным и влиятельным видом искусства. В настоящем исследовании раскрывается образовательное значение кинематографа. Фильмы могут быть эффективным инструментом передачи информации научного характера и привлечения широкой аудитории к актуальным вопросам современности, поскольку являются как визуально, так и эмоционально привлекательными.

На данный момент в нашей стране осуществляется политика развития и продвижения национальной культуры различными способами: это и проведение тематических фестивалей и иных мероприятий, и изучение историко-культурного наследия Беларуси в текущих программах среднего и высшего образования [1]. Однако не все меры одинаково эффективны [2]. Настоящая работа посвящена раскрытию значимой роли кинематографа в вопросах поднятия уровня национальной культуры. Посредством кинематографа можно доносить до широкой аудитории не только исторические факты, но и достаточно сложные концепции и идеи, например, политической или экономической направленности. Успех таких мероприятий подтверждается исследованием, опубликованным в издании *Eurasia J. Math. Sci. Technol* [2].

Кроме того, исследование, опубликованное в издании *The Journal of Effective Teaching*, показало, что просмотр исторического фильма может улучшить понимание и усвоение материала, особенно если фильм при просмотре вызывает положительные эмоции [3]. Это подтверждается и опросом, проведенным среди студентов Белорусского государственного университета информатики и радиоэлектроники. Было опрошено 227 респондентов. Из данного числа респондентов 60,1% ответили положительно на вопрос: «Помогло ли историческое кино разобраться глубже в теме, раскрытой его сюжетом?».

Вместе с тем, 77,2% из них сообщили о проблемах с изучением историко-культурного наследия нашей страны, как в школе, так и на первом курсе университета. Как представляется, существует несколько причин, которые могут объяснить вышеупомянутые сложности. К ним можно отнести и информационную перегрузку в учебных программах, и стиль преподавания дисциплин, ставящих перед собой такую задачу, и отсутствие у обучающихся понимания связи между прошлым и проблемами сегодняшнего дня. Весомой помощью в решении этих вопросов и успешном приобщении к истории и культуре нашей страны может стать кинематограф. Согласно вышеупомянутому опросу, из общего числа респондентов, 61,9% применяли полученные из кино исторические знания в повседневной жизни, при этом 49,1% использовали их также во время учебных занятий.

Вместе с тем, белорусский кинематограф не вызывает широкого интереса со стороны современного зрителя [4]. Это подтверждается и некупаемостью картин киностудии «Беларусьфильм», и вышеуказанным опросом (только 12,9% респондентов предпочитают белорусское кино). Однако у этого направления всё же есть потенциал, который можно и нужно развивать. К примеру, как показал опрос, история белорусского средневековья богата на занимающие современного зрителя эпизоды.

Таким образом, кинематограф является значимым фактором приобщения к историко-культурному наследию нашей страны, эффективным способом привлечения внимания и подъёма интереса к национальной культуре. Активное развитие отечественного кинематографа позволило бы поднять уровень национального самосознания и развить национальную культуру быстрым и приемлемым для современной молодёжи способом.

Список использованных источников:

1. Brehony, K. J. (2011). *Cinema in Education: A Critical Review*. *Journal of Educational Multimedia and Hypermedia*, 20(2), 143-163.
2. *Послание Президента Республики Беларусь Александра Лукашенко белорусскому народу и Национальному собранию. 21 апреля 2010 года [Электронный ресурс]. – Режим доступа: http://president.gov.by/ru/news_ru/view/poslanie-prezidenta-respubliki-belarusaleksandra-lukashenko-beloruskomu-narodu-i-natsionalnomu-sobraniju-5887/. – Дата доступа: 27.03.2022.*
3. Chen, J. J. (2015). *Using Films as a Tool for Active Learning in Teaching Sociology*. *Journal of Effective Teaching*, 15(1), 5-20.

Конференция «Компьютерные системы и сети»

4. Культура Беларуси | *Официальный интернет-портал Президента Республики Беларусь [Электронный ресурс].
Режим доступа: <https://president.gov.by/ru/belarus/social/culture>. – Дата доступа: 27.03.2023*

НЕЧЕТКАЯ ЛОГИКА В ИСКУССТВЕННЫХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМАХ

Цыркунов А.С., студент гр.221703

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Лагунова Е.Н. – преподаватель

Аннотация. Принципы нечеткой логики позволяют применять ее к анализу и моделированию искусственных интеллектуальных систем, способных к рациональному выбору в ситуации неясного знания или недостатка информации.

Искусственный интеллект (ИИ) – это технология создания различных компьютерных систем и машин, а также область науки, которая устанавливает принципы их работы. При этом предполагается, что ИИ может и должен выполнять задачи, свойственные человеческому мышлению: сбор и анализ информации, способность к обучению и пр. Сегодня ИИ используются практически во всех областях нашей жизни, особенно крупными IT-компаниями производящими, главным образом, информационные продукты. Широкая область применения искусственного интеллекта делает его одной из самых актуальных и перспективных областей науки и технологии.

ИИ тесно связан с логикой как наукой, поскольку его алгоритмы и методы основаны на рационально-логических принципах. В классической бинарной логике суждения как мысленные формы могут быть либо истинными (и принимать значение единицы), либо ложными (принимать значение нуля), но в реальном мире некоторые явления не могут быть описаны с помощью такого подхода, поскольку суждения о них могут иметь различные степени истинности. В таких случаях более подходящим способом моделирования логических процессов выступает нечеткая логика.

Основными принципами нечеткой логики можно назвать следующие: суждения могут иметь различные степени истинности; множественность; интерпретативность; интуиция. Это позволяет моделировать системы, способные функционировать в состоянии неопределенности. Нечеткая логика может использоваться в совершенно различных областях человеческой деятельности: создании систем управления, философских рассуждениях, теории вероятностей и т.д.

Базовое понятие нечеткой логики – нечеткое множество. Нечеткое множество – это класс объектов с континуумом степеней принадлежности, характеризующихся функцией принадлежности, которая придает каждому объекту степень принадлежности, принимающую значение между нулем и единицей, где 0 означает полное отсутствие принадлежности, а 1 – полную принадлежность. Например, понятие «высокая температура», – нечеткое множество, где каждая определенная температура имеет свою степень принадлежности к этому множеству. В нечеткой логике введены понятия операций объединения, пересечения, дополнения композиции, разности и наложения нечетких множеств. Некоторые из этих операций, введенные для классических множеств, аналогичны, но учитывают степени принадлежности элементов к нечетким множествам.

Примеры прикладных задач, для решения которых применима нечеткая логика:

1. Автоматизированное принятие решений. Нечеткая логика может использоваться для создания экспертных систем, которые принимают решения в условиях неопределенности и неполноты информации. Например, экспертная система для оценки платежеспособности клиента банка может использовать нечеткую логику для анализа множества факторов, таких как возраст, пол, заработная плата, наличие текущих займов и пр.

2. Управление электроникой. Нечеткая логика может использоваться для управления сложными системами, например, управление температурой в помещении, учитывая множество факторов, таких как температура на улице, количество людей в помещении и пр.

3. Моделирование поведения людей. Нечеткая логика может использоваться для моделирования поведения людей в различных жизненных ситуациях. Например, для моделирования поведения водителей на дороге нужно учитывать множество факторов, таких как скорость движения, расстояние до других автомобилей, погодные условия и пр.

4. Нейронные сети. Нечеткая логика применяется для построения классификаторов, которые способны обрабатывать нечеткие категории и метки. В связи с активным ростом популярности нейронной сети «Chat GPT» хочется отметить, что и она использует нечеткую логику для создания экспертных систем, которые могут обрабатывать нечеткие или неопределенные вопросы и выдавать соответствующие ответы.

Таким образом, нечеткая логика выступает одним из эффективных инструментов моделирования ИИ в ситуации неопределенности и многозначности информационных процессов.

Список использованных источников:

1. Горбаченко, В. И. *Интеллектуальные системы: нечеткие системы и сети* / В. И. Горбаченко, Б. С. Ахметов, О. Ю. Кузнецова. — Москва : Издательство Юрайт, 2023. — 105 с.

ФИЛОСОФСКИЕ ПРОБЛЕМЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Чечко В.В.

*Белорусский государственный университет информатики и радиоэлектроники г. Минск,
Республика Беларусь*

Шепетюк В.В. – канд. филос. наук, доцент кафедры философии БГУИР

Аннотация. В работе представлено философское осмысление сложной междисциплинарной проблемы «искусственного интеллекта». Рассматриваются теоретико-методологические и философские вопросы. Характеризуются интеллект человека и «искусственный интеллект» компьютера. Описывается существенное их различие, приводятся позитивные и негативные последствия вторжения «искусственного разума» в повседневную жизнь.

Проблема «искусственного интеллекта» возникла достаточно давно. Однако чрезвычайно актуализировалась в 21 веке, стала по-настоящему горячей и судьбоносной. Вторжение «искусственного интеллекта» во многие сферы общества, возникновение цифровой экономики, усиливающаяся роль виртуальной реальности в нашем существовании в значительной мере определяют развитие человеческого рода. В связи с этим некоторые исследователи говорят о наступлении постантропологической эпохи[1]. Данная проблема вызывает исключительный интерес. Её широко и оживлённо обсуждают кибернетики, психологи, математики, логики и лингвисты. Однако её философские аспекты исследуются недостаточно. В данной работе осуществляется попытка философского осмысления, теоретико-методологического анализа проблемы «искусственного интеллекта». Анализ зарубежной литературы, посвященной проблеме «искусственного интеллекта», свидетельствует о том, что в ней господствуют натуралистический, естественно-научный, механистический, физикалистский и кибернетический подходы. Происходит неправомерное отождествление человеческого и «искусственного интеллекта», осуществляется «расчеловечивание» человека, его «машинизация» и оглушение, с одной стороны, а с другой стороны – «очеловечивание», антропоморфизация ЭВМ, то есть наделение её сугубо человеческими свойствами, в частности такими, как сознание, разум, знание, творчество, воля, чувства, эмоции.

В двадцатом столетии на волне достижений в области кибернетики, физиологии, нейробиологии и математической логики в западной науке и философии появилось течение, получившее название «Научный материализм». Это неоднородное течение в аналитической философии, в рамках которого в основном рассматриваются проблемы определения природы сознания, причем делается это с опорой на данные естествознания[2]. Его основные представители: Д. Армстронг, Н.П. Бехтерева, М. Бунге, Дж. Марголис, Дж. Смит, Р. Петроус, Дж. Сёрл, А.И. Яковлев. В их работах приводятся доводы о том, что сознание, разум и интеллект человека имеют материальную и энергетическую природу. Однако в современной англо-американской философии сознания ведущим течением является физикализм, который представляет собой стремление перевести всё знание на язык физики. Данное течение исходит из утверждения, что «всё есть физика», то есть всё подчинено жёстким физическим законам. В связи с этим психические, умственные, психофизиологические и биологические процессы мозга являются физическими феноменами. Главными его представителями являются Д. Дэвидсон, Дж. Ким и Д. Папино.

Понятие интеллекта однозначно и чётко не определено. В частности, в разных источниках можно обнаружить различные трактовки: «[лат. intellectus – познание, понимание, рассудок] – способность мыслить, совершать разумные действия, познавать»; «[лат. intellectus – ум, рассудок] – в истории философии понятие «интеллект» встречается главным образом для обозначения «чистой», активной силы мышления, принципиально отличающейся своим творческим характером от пассивных чувственных форм познания. Эта сила, или способность, трактовалась как исключительная особенность, характеризующая разумное существо, человека» [3]. В связи с этим, можно сказать, что интеллект человека – это общая характеристика его умственного потенциала, которая включает в себя способность познания и понимания действительности, самого себя и других людей. Интеллект представляет собой совокупность всех умственных способностей, навыков и умений человека, использование им в своей деятельности всех познавательных средств, идеальных образов, форм абстрактного, логического, креативного и интуитивного мышления.

Идея создания разумных машин по образу и подобию человека имеет долгую историю. Первые попытки их разработки приписываются ещё античным изобретателям. В 17 веке Б. Паскаль создал первую работающую машину, которая выполняла сложение двух чисел. В 18 веке французский изобретатель Ж. де Вокансон изготовил антропоморфного механического флейтиста, который мог исполнить 12 мелодий, производя механическое воздействие на клавиши инструмента и пропуская воздушную струю в мундштук. Данные примеры характерны для механистического подхода к проблеме создания «искусственного интеллекта». Этот подход имеет ряд недостатков, главным из которых является полное игнорирование особенностей человеческого сознания и

разума. В связи с этим необходимо обратиться к характеристике «искусственного интеллекта». «Искусственный интеллект» – моделирование и имитация человеческого интеллекта, его психических и мыслительных процессов в компьютере, а также создание «электронного мозга», который представляет собой техническое устройство, превосходящее человеческий интеллект. В литературе обычно указываются следующие характерные черты компьютерной системы, «искусственного интеллекта»:

- способность решать сложные задачи, хранить большие объёмы информации и работать с ними;
- наличие возможности пополнять имеющиеся знания;
- способность к генерации информации, которая в явном виде не содержится в системе (это качество позволяет системе конструировать информационную структуру с новой семантикой и практической направленностью);
- умение оперировать в ситуациях, связанных с различными аспектами нечёткости, включая «понимание» естественного языка;
- способность к адаптации.

«Разумные» устройства являются серьёзными помощниками человека, так как значительно расширяют его вычислительные и информационные способности. В связи с этим необходимо более эффективно использовать интеллектуальный потенциал системы «человек–машина» в познавательной, проектной и прогностической деятельности, разумно использовать «искусственный интеллект» в экономике, управлении и других областях, где необходимо оперировать большими массивами информации. Однако важно учесть то, что усиливающиеся процессы информатизации и роботизации, внедрение высоких технологий и повсеместного использования «искусственного интеллекта» существенно обостряют технико-технологическое отчуждение. Машина, созданная человеком, превращается во враждебную ему силу, начинает поработать его, угрожает сохранению его биосоциальной природы и духовно-нравственной сущности[4, с. 159 – 160].

Наиболее актуальной угрозой является применение недостаточно обученного ИИ в жизненно важных сферах, а также попытки его использования для контроля жизни человека. Одной из проблем является способность искусственного интеллекта создавать и публиковать в социальных сетях тексты, которые достаточно трудно отличить от написанных человеком. В качестве примера можно привести бота, созданный одной из версий программы GPT, который выдавал себя за человека в течение нескольких недель.

Иногда тексты ИИ невозможно распознать как искусственные. Исходя из этого, можно сказать, что машина может управлять даже эмоциями своего собеседника. Данные боты могут писать комментарии, создавать тот или иной информационный фон и подталкивать своего собеседника к тем или иным действиям, способствовать формированию определённого мнения, «выгодного» третьим лицам

Серьёзной проблемой может являться передача принятия части решений искусственному интеллекту как помощнику без адекватной оценки его возможностей. Если, к примеру, лечащий врач полностью доверит ИИ расшифровку рентгеновских снимков, ошибочные выводы могут иметь серьёзные последствия. Установка ИИ в качестве фильтра на вызовах «скорой помощи» также является небезопасной. Это может создать угрозу для жизни человека, находящегося в стрессе, речь которого недостаточно обученный искусственный интеллект может неверно интерпретировать. Однако всё это не значит, что ИИ бесперспективно использовать в принципе.

Приведённые выше аргументы позволяют сделать следующие выводы: машинная, компьютерная система «искусственного интеллекта» не способна рассуждать, конструировать действительность и прогнозировать события. Она не имеет творческого, рефлексивного, критического мышления и самосознания. «Искусственный интеллект» не владеет мудростью, мировоззренческими универсалиями культуры и не имеет своего собственного внутреннего духовного мира, идеологических установок, ценностных ориентаций и идеалов. Его работа запрограммирована человеком и осуществляется по принципам «да или нет», «или-или», «либо-либо», то есть односторонним и примитивным образом. В связи с этим можно сказать, что человек является уникальной «единицей» мироздания, на плечи которого ложится полный контроль за развитием «умной» машины.

Список использованных источников:

1. Фукуяма, Ф. *Наше постчеловеческое будущее: Последствия биотехнологической революции* / Ф. Фукуяма. – М.: АСТ, 2004. – 352 с.;
2. Марков, Б.В. *Философия. Стандарт третьего поколения: учеб пособие* / Б.В. Марков. – СПб.: Питер, 2017. – 464 с.
3. Кутырев, В.А. *Естественное и искусственное. Борьба миров* / В.А. Кутырев. – Изд-во «Нижний Новгород», 1994. – 200 с.
4. Канке, В.А. *Философские науки: краткий энциклопедический словарь* / В.А. Канке. – М.: Омега-Л, 2008. – 359 с.

Научное электронное издание

Материалы 59-й научной конференции аспирантов,
магистрантов и студентов БГУИР

по направлению

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

(Республика Беларусь, г. Минск, 17–21 апреля 2023 г.)

Дата размещения на сайте 17.04.2023.

Объем издания: 19,90 Мбайт.

Производитель электронного издания:
учреждение образования «Белорусский
государственный университет
информатики и радиоэлектроники».

Ул. П. Бровки, 6, 220013, г. Минск.

Сайт: www.bsuir.by