

Научный центр «LJournal»

Рецензируемый научный журнал

# **ТЕНДЕНЦИИ РАЗВИТИЯ НАУКИ И ОБРАЗОВАНИЯ**

№116, Декабрь 2024  
(Часть 19)



Самара, 2024

T33

**Рецензируемый научный журнал «Тенденции развития науки и образования» №116, Декабрь 2024 (Часть 19) - Изд. Научный центр «LJournal», Самара, 2024 - 192 с.**

**doi:** 10.18411/trnio-12-2024-p19

**Тенденции развития науки и образования** - это рецензируемый научный журнал, который в большей степени предназначен для научных работников, преподавателей, доцентов, аспирантов и студентов высших учебных заведений как инструмент получения актуальной научной информации.

Периодичность выхода журнала – ежемесячно. Такой подход позволяет публиковать самые актуальные научные статьи и осуществлять оперативное обнародование важной научно-технической информации.

Информация, представленная в сборниках, опубликована в авторском варианте. Орфография и пунктуация сохранены. Ответственность за информацию, представленную на всеобщее обозрение, несут авторы материалов.

Метаданные и полные тексты статей журнала передаются в наукометрическую систему ELIBRARY.

Электронные макеты издания доступны на сайте научного центра «LJournal» - <https://ljournal.org>

© Научный центр «LJournal»  
© Университет дополнительного  
профессионального образования

УДК 001.1  
ББК 60

## РЕДАКЦИОННАЯ КОЛЛЕГИЯ

**Чернопятов Александр Михайлович**  
Кандидат экономических наук, Профессор

**Царегородцев Евгений Леонидович**  
Кандидат технических наук, доцент

**Пивоваров Александр Анатольевич**  
Кандидат педагогических наук

**Малышкина Елена Владимировна**  
Кандидат исторических наук

**Ильященко Дмитрий Павлович**  
Кандидат технических наук

**Дробот Павел Николаевич**  
Кандидат физико-математических наук, Доцент

**Божко Леся Михайловна**  
Доктор экономических наук, Доцент

**Бегидова Светлана Николаевна**  
Доктор педагогических наук, Профессор

**Андреева Ольга Николаевна**  
Кандидат филологических наук, Доцент

**Абасова Самира Гусейн кызы**  
Кандидат экономических наук, Доцент

**Попова Наталья Владимировна**  
Кандидат педагогических наук, Доцент

**Ханбабаева Ольга Евгеньевна**  
Кандидат сельскохозяйственных наук, Доцент

**Вражнов Алексей Сергеевич**  
Кандидат юридических наук

**Ерыгина Анна Владимировна**  
Кандидат экономических наук, Доцент

**Чебыкина Ольга Альбертовна**  
Кандидат психологических наук

**Левченко Виктория Викторовна**  
Кандидат педагогических наук

**Петраш Елена Вадимовна**  
Кандидат культурологии

**Романенко Елена Александровна**  
Кандидат юридических наук, Доцент

**Мирошин Дмитрий Григорьевич**  
Кандидат педагогических наук, Доцент

**Ефременко Евгений Сергеевич**  
Кандидат медицинских наук, Доцент

**Шалагинова Ксения Сергеевна**  
Кандидат психологических наук, Доцент

**Катермина Вероника Викторовна**  
Доктор филологических наук, Профессор

**Полицинский Евгений Валериевич**  
Кандидат педагогических наук, Доцент

**Жичкин Кирилл Александрович**  
Кандидат экономических наук, Доцент

**Пузыня Татьяна Алексеевна**  
Кандидат экономических наук, Доцент

**Ларионов Максим Викторович**  
Доктор биологических наук, Доцент

**Афанасьева Татьяна Гавриловна**  
Доктор фармацевтических наук, Доцент

**Байрамова Айгюн Сеймур кызы**  
Доктор философии по техническим наукам

**Лыгин Сергей Александрович**

Кандидат химических наук, Доцент

**Заломнова Светлана Петровна**

Кандидат педагогических наук, Доцент

**Биймурсаева Бурулбубу Молдосалиевна**

Кандидат педагогических наук, Доцент

**Радкевич Михаил Михайлович**

Доктор технических наук, Профессор

**Гуткевич Елена Владимировна**

Доктор медицинских наук

**Матвеев Роман Сталинарьевич**

Доктор медицинских наук, Доцент

**Пронина Наталья Андреевна**

Кандидат педагогических наук, Доцент

**Шамутдинов Айдар Харисович**

Кандидат технических наук, Профессор

**Найденев Николай Дмитриевич**

Доктор экономических наук, Профессор

**Романова Ирина Валентиновна**

Кандидат экономических наук, Доцент

**Хачатурова Карине Робертовна**

Кандидат педагогических наук

**Кадим Мундер Мулла**

Кандидат филологических наук, Доцент

**Григорьев Михаил Федосеевич**

Кандидат сельскохозяйственных наук

## СОДЕРЖАНИЕ

<b>РАЗДЕЛ XIX. ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА</b> .....	8
<b>Аброськин Р.С., Котрунов Р.А., Ермолаева В.В.</b> Эволюция процессоров.....	8
<b>Антипов Д.Е.</b> Компьютерная преступность .....	11
<b>Биджиева С.Х., Байрамукова М.И., Гериева К.М.</b> Анализ метрик оценки качества генеративных языковых моделей .....	15
<b>Биджиева С.Х., Байрамукова М.И., Гериева К.М.</b> Этапы и методы обработки естественного языка .....	17
<b>Ближенский М.В.</b> Анализ устойчивых уязвимостей в PostgreSQL.....	20
<b>Гараев А.Х., Агаев А.Х., Шахбазова М.С.</b> Безопасность баз данных: защита данных с помощью SQL.....	24
<b>Гараев А.Х., Гараев М.Х., Шахбазова М.С.</b> Методы обработки сенсорных данных..	27
<b>Гараев А.Х., Гараев М.Х., Шахбазова М.С.</b> Принцип работы гироскопического датчика.....	29
<b>Гараев А.Х., Гараев М.Х., Шахбазова М.С.</b> Принцип работы датчика ускорения .....	32
<b>Гараев А.Х., Гараев М.Х., Шахбазова М.С.</b> Сравнение SQL с процедурными языками программирования: как SQL упрощает взаимодействие с базами данных .....	35
<b>Гарипов И.И., Андреева Е.А.</b> Применение технологий искусственного интеллекта в различных отраслях .....	38
<b>Гарфетдинова К.Р., Баранов С.А., Хусаинова Е.А.</b> Перспективы развития рынка искусственного интеллекта в России .....	43
<b>Голубятников А.О.</b> Альтернативные методы геолокации на основе данных GSM и Wi-Fi.....	46
<b>Голубятников А.О.</b> Перспективы и будущее развития LPWAN .....	50
<b>Голубятников А.О.</b> Технологии Wi-Fi RTT и Fingerprinting для высокоточного позиционирования в помещениях .....	52
<b>Гочияева М.Д., Урумов Ш.А.</b> Роль MRP-систем в оптимизации производственного планирования компьютерной техники.....	55
<b>Гочияева М.Д., Чмилев К.Е.</b> Алгоритмы автоматизации юридического контроля при заказах печатной продукции: от контрактов до доставки.....	59
<b>Гочияева М.Д., Чотчаев Р.Б.</b> Чат-боты как инструмент для повышения вовлеченности школьников в учебный процесс.....	62
<b>Григорьева Е.Г., Солонина А.П.</b> Технологии визуализации в архитектурном моделировании (на примере Царицынского парка г. Волгограда) .....	65
<b>Гуржов Н.А., Переверзев А.Л., Силантьев А.М., Примаков Е.В., Солодовников А.П.</b> Разработка реконфигурируемого систолического массива для ускорения вычисления слоёв нейронной сети .....	75
<b>Данаев М.Х., Магамадов А-С.С., Шахбазова М.С.</b> Перспективы внедрения нейросетей в образование.....	80
<b>Данаев М.Х., Магамадов А-С.С., Шахбазова М.С.</b> Программирование с нуля: с чего начать и какой язык выбрать.....	82

<b>Даниленко В.С.</b> Показатели качества в сети IoT и методы их обеспечения.....	85
<b>Денек Д.А., Ротенштрайх Т.В.</b> Создание гибкой архитектуры мобильного приложения на Flutter с использованием паттерна MVVM.....	89
<b>Дмитриев А.В., Алексеев Д.И.</b> Как блокчейн меняет финансовую индустрию .....	92
<b>Игнатова Я.А.</b> Применение машинного обучения для обнаружения и предотвращения угроз в приложениях .....	96
<b>Игнатова Я.А.</b> Сетевые технологии связи.....	98
<b>Игнатова Я.А.</b> Тестирование на проникновение в системах IoT .....	102
<b>Каленикин В.Н., Панова М.О.</b> Автоматизированная информационная система составления и оптимизации расписания высшего учебного заведения: актуальность внедрения, ключевые параметры.....	105
<b>Калмыков С.Е., Вострова Е.А.</b> Оптимизация мониторинга подключений и ресурсов в PostgreSQL .....	108
<b>Лыткин К.А.</b> Разработка решения для автоматизированного парсинга ошибок в банковских системах.....	112
<b>Лыткин К.А., Ильина А.А.</b> Raspberry PI и система умного дома: настройка и мониторинг через Telegram бота .....	115
<b>Магаматов А-С.С., Данаев М.Х., Шахбазова М.С.</b> Java vs Python: какой язык выбрать начинающему программисту .....	118
<b>Мизаев М.М., Шахбиев Д.О-Х., Муцурова З.М.</b> Особенности языка C++, применение и преимущества .....	121
<b>Напалков С.В.</b> Центры обработки данных в мире. Глобальное распределение и тренды развития.....	126
<b>Орехов В.В., Лытнев Н.Н.</b> Обзор технологий и сервисов для разработки и запуска облачных игр.....	130
<b>Падаев Р.Р.</b> Выбор сетевых протоколов для эффективного взаимодействия в Интернет вещей .....	133
<b>Панова Ю.А.</b> Интеграция ЕПП и ALD с PostgreSQL для управления доступом и безопасности в корпоративной сети.....	136
<b>Протасова Е.А., Переверзев А.Л., Силантьев А.М., Примаков Е.В., Солодовников А.П.</b> Способы аппаратной реализации функции активации в нейронных сетях.....	143
<b>Рибницкая И.В., Никифорова Е.М., Складенко А.А., Туренко Е.В.</b> Инфографика и другие информационные системы в аэропортах.....	147
<b>Токарев Е.В.</b> Выбор протоколов TCP, UDP и WebSocket для различных типов приложений.....	150
<b>Токарев Е.В.</b> Основы сетевого программирования и роль протоколов и сокетов в сетевом взаимодействии.....	153
<b>Цаплин В.И., Панов М.А.</b> Технология распределенного реестра.....	159
<b>Чэнь И.</b> Применение методов машинного обучения для анализа больших данных в информационных системах: вызовы и перспективы .....	163
<b>Шахбиев Д.О-Х., Магомедов И.И.</b> Искусственный интеллект в образовании.....	166
<b>Шевко Н.Р., Миронова Е.А.</b> Особенности обеспечения кибербезопасности в современных условиях.....	169

---

<b>Эркенова М.У., Аджиев И.Р.</b> Будущее автоматизации в туристическом бизнесе: прогноз развития технологий и тенденций.....	172
<b>Эркенова М.У., Джибабов М.М.</b> Создание персонализированных учебных планов с помощью машинного обучения для изучения родного языка.....	175
<b>Эркенова М.У., Хачиров И.А.</b> Сравнительный анализ существующих CRM-систем и возможности их адаптации на 1С: Предприятие 8.3 .....	177
<b>Юданов Р.С.</b> Вопросы информационной безопасности web-приложений .....	180
<b>Dudak A.</b> Optimization of loading and performance in SPA on React .....	183

## РАЗДЕЛ XIX. ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Аброськин Р.С., Котрунов Р.А., Ермолаева В.В.

### Эволюция процессоров

Саратовский государственный технический университет имени Ю.А. Гагарина  
(Россия, Саратов)

doi: 10.18411/trnio-12-2024-849

#### Аннотация

В современном мире используется большое число вычислительной техники, обладающая высокой скоростью работы и компактностью, но за этим свойством стоит длительный процесс, который продолжается и сейчас. Процессоры используются во всех сферах деятельности, тем самым облегчают нашу жизнь.

**Ключевые слова:** процессор, производительность, архитектура, техпроцесс.

#### Abstract

In the modern world, a large number of computing equipment is used, which has high speed and compactness, but behind these properties there is a long process that continues even now. Processors are used in all fields of activity, thereby making our lives easier.

**Keywords:** processor, performance, architecture, the technical process.

#### Введение

Процессоры являются главным компонентом вычислительных систем. Их развитие – это история непрерывного процесса изменений и инноваций, которая определила производительность персональных компьютеров, а также оказала влияние на все сферы жизни. С чего начиналась история процессоров, и каких результатов добилась индустрия с XX века.

#### Эволюция процессоров

Первый в мире коммерческий процессор был представлен в 1971 году корпорацией Intel под названием Intel 4004. Его появление ознаменовало революцию в области вычислительной техники. Это 4-битный процессор, который мог выполнять 60 тысяч инструкций в секунду и включал в себя 2300 транзисторов на кристалле 3x4 мм. Такой размер является достижением того времени, так как ранее для выполнения аналогичных задач требовалось множество отдельных комплектующих. Был предназначен для настольных калькуляторов, измерительных приборов, радиостанций, а также использовался в игровых автоматах. Его успех дал толчок к созданию более сложных устройств.

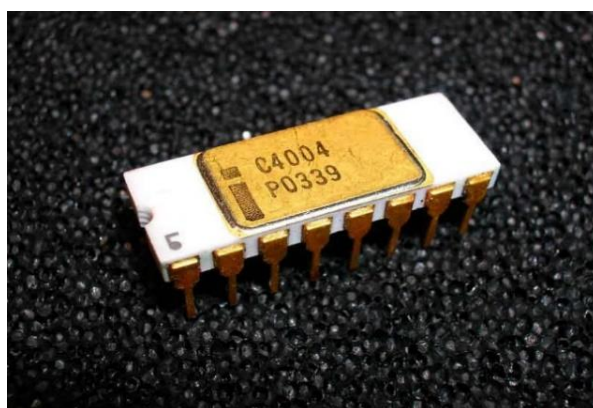


Рисунок 1. Intel 4040.

С выходом Intel 8080 в 1974 году, индустрия вступила на следующую ступень эволюции. В нем реализовали 8-битную архитектуру. Это означало, что он способен обрабатывать 8 бит данных за один такт. Произошло увеличение набора команд, а также появилась возможность работать с внешней памятью, благодаря этому случился мощный рывок в развитии языков программирования и операционных систем. На основе данной архитектуры была написана операционная система CP/M (Control Programs for microcomputers), которая стала первой ОС общего назначения для компьютеров с дисковыми магнитными накопителями. Данный процессор стал основой первых домашних компьютеров. Одним из таких стал Altair 8800. А в 1978 году корпорация Intel выпустила 16-битный процессор 8086, который стал фундаментом архитектуры x86.

Архитектура x86 стала одной из самых популярной и широко используемой архитектурой в мире. С ее приходом появилась возможность использовать сегментированную память, что позволила значительно увеличить объем памяти и улучшить производительность. В 1985 году был выпущен 32-битный Intel 80386, который обеспечил поддержку виртуальной памяти, что поспособствовала увеличению возможностей для работы в многозадачном режиме.

Значительный шаг в прогрессе совершила новейшая 64-битная архитектура AMD64. 64-битные процессоры имели возможность использовать значительно больший объем памяти (более 4 ГБ). Это является важным показателем для работы серверов и рабочих станций, ведь они обрабатывают большой объем информации. Также это привело к увеличению производительности в графических редакторах и базах данных, ведь данные программы являются многопоточными.



Рисунок 2. AMD Athlon 64.

В начале 2000-х годов потребность в производительности превышала возможности одноядерных процессоров. Увеличение тактовой частоты стало не эффективным из-за большого TDP (Thermal Design Power).

Первым многоядерным стал Intel Pentium D, представленный в 2005 году. Он имел два ядра и давал возможность обрабатывать несколько сложных задач одновременно. Вскоре компания AMD выпустила Athlon 64 X2, который стал частым выбором для использования в домашних компьютерах.

Многоядерные процессоры обеспечивают прирост производительности в задачах, которые требуют параллельной обработки. Например, 3D-рендеринг, видеообработка, компьютерные игры.

С каждым новым поколением процессоров технология производства становилась сложнее. Переход на техпроцессы с меньшим размером транзисторов смог в разы увеличить плотность транзисторов на кристалле. Это привело к увеличению производительности и снижению TDP.

Одним из новшеств является интегрированная графика iGPU (integrated Graphics Processing Unit). Она представляет собой процессорную единицу, которая интегрирована в ЦП (Центральный Процессор) или северный мост (Northbridge). Выполняет такие же функции, как и дискретная видеокарта, но не имеет собственной памяти, поэтому использует часть оперативной памяти (ОЗУ). Первым действенным решением был представлен Intel Core 2 Duo в 2006 году со встроенным графическим процессором GMA

3000, который имел 256 КБ видеопамяти. Конечно, данное решение было выполнено в пользу мобильных устройств, чтобы решать стандартные и офисные задачи, а для сложных вычислений не подходило. iGPU используется только в базовых задачах, таких как вывод 2-D графики или видео.

Также современные ЦП ориентированы на энергоэффективность. Инженеры стремятся создать такие процессоры, у которых большая производительность и маленькое TDP. В наше время это особенно актуально для мобильных устройств, таких как смартфоны, ноутбуки, смарт часы. Связано это с увеличением длительности работы от аккумулятора. Примером этого могут являться процессоры линейки Qualcomm Snapdragon, которые базируются на архитектуре ARM (Advanced RISK Machine). Они демонстрируют высокую производительность в различных задачах и отличную энергоэффективность.

Современные технологии дают возможность достичь более меньших размеров транзисторов, что открывает двери для дальнейшего развития индустрии. Одна из таких технологий называется EUV - экстремальная ультрафиолетовая литография.

По оценки компании TSMC (Тайваньская полупроводниковая компания), в 2026 году начнется серийное производство чипов по 2-нм техпроцессу, а в период с 2027 – 2030 год случится переход на 1-нм техпроцесс.

Также в настоящее время развивается отрасль квантовых процессоров. Квантовые процессоры представляют собой устройства, использующие явления квантовой механики для работы с информацией. Отличие от классических процессоров заключается в том, что квантовые оперирует не битами, а кубитами, имеющими значение 0 и 1 одновременно. Это позволяет выполнять определенные вычисления в разы быстрее, чем классические процессоры.

Подводя итоги, необходимо увидеть, насколько сильно шагнула индустрия. Продemonстрируем сравнение процессоров Intel 4004 и Intel Core i9-13900K:

Таблица 1

Сравнение характеристик Intel 4004 и Intel I7 – 10700K

Характеристики	Intel 4004	Intel Core i9-13900K
Техпроцесс, нм	10 мкм	10 нм
Кол-во транзисторов	2300	≈ 20000000000
Тактовая частота	0,74 МГц	3,0 ГГц
Частота шины	0,5 МГц	100 МГц
Количество ядер	1	24
Количество потоков	1	32
Поддержка ОЗУ	4 КБ	128 ГБ
iGPU	Нет	Intel UHD Graphics 770

Также продемонстрируем график роста производительности процессоров с 1971 года по 2024 год:



График 1.

Таким образом, процесс совершенствования процессоров продолжается уже многие годы, тем самым делая нашу жизнь совершеннее и ярче.

\*\*\*

1. Петров, И. С. Современные тенденции в компьютерных технологиях // Эволюция архитектуры процессоров. СПб.: Политехника, 2021. С. 45-67.
2. Лебедев, М. И. Актуальные вопросы информатики и вычислительной техники // Тенденции в развитии процессоров. Ростов-н/Д.: Южный федеральный университет, 2019. С. 112-130.
3. Сидоров, В. Н. Процессоры: от первых микросхем до многопроцессорных систем. Екатеринбург.: Урал, 2020.
4. Чивилихин, С. А. Учебное пособие // Квантовая информатика. СПб., 2009.

**Антипов Д.Е.**

### **Компьютерная преступность**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М. А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-850

#### **Аннотация**

В статье рассматривается проблема компьютерной преступности, которая в последние годы становится одной из наиболее острых и актуальных угроз информационной безопасности. Описаны основные виды компьютерных преступлений, такие как хакерские атаки, распространение вредоносного ПО, кибермошенничество и фишинг. Особое внимание уделяется анализу методов борьбы с компьютерной преступностью, а также правовым аспектам регулирования в данной сфере. Приводятся статистические данные, иллюстрирующие масштабы проблемы на глобальном и региональном уровнях, и предлагаются рекомендации по снижению рисков для пользователей и организаций.

**Ключевые слова:** киберпреступность, хакерские атаки, информационная безопасность.

#### **Abstract**

The article deals with the problem of computer-related crimes, which has become one of the most acute and urgent threats to information security recently. The article describes the main types of computer-related crime including hacker attacks, malware distribution, cyber fraud and phishing. Special attention is paid to the analysis of IT-crime control methods, as well as to the legal aspects of regulation in this area. Statistics illustrating the extent of the issue at the global and regional levels is provided, and recommendations for mitigating risks to users and entities are offered.

**Keywords:** cybercrime, hacker attacks, information security.

#### **Введение**

Концепция компьютерной преступности зародилась в середине XX века, когда начались первые попытки несанкционированного доступа к компьютерным системам. С тех пор, с развитием технологий и интернета, масштабы и сложность этих преступлений значительно возросли. Компьютерная преступность охватывает широкий спектр действий, начиная от хакерских атак и распространения вредоносного ПО, до кибермошенничества и DDoS-атак. Эти преступления наносят огромный ущерб как частным лицам, так и организациям, вызывая финансовые потери, утечки данных и другие негативные последствия. Эффективная борьба с компьютерной преступностью требует постоянного совершенствования как правовых, так и технических мер защиты.

### **Серьезность проблемы**

Проблема, представленная в данной статье, имеет огромную важность для всего человечества в целом. Киберпреступность представляет собой одну из самых серьёзных угроз в современном мире, ставя под угрозу не только государственную и личную безопасность, но и стабильность экономических систем. С ростом числа подключённых к интернету устройств и развитием информационных технологий, киберпреступления приобретают всё более глобальный характер. Злоумышленники используют сложные методы, чтобы проникать в системы и сети, что создаёт серьёзные вызовы для бизнеса, государственных органов и частных лиц.

Одной из причин, по которой киберпреступность является столь серьёзной проблемой, является её глобальный характер. Кибератаки не ограничены национальными границами: злоумышленники могут действовать из любой точки мира, что затрудняет их отслеживание и наказание. Более того, киберпреступные сети часто состоят из участников из разных стран, что делает их ещё более сложными для раскрытия и преследования. Это международное измерение киберпреступности требует координированных усилий на уровне международных организаций и правительств для эффективной борьбы с угрозами.

Экономический ущерб от киберпреступности также подчеркивает серьёзность проблемы. Согласно оценкам, ежегодные убытки от киберпреступлений составляют триллионы долларов, нанося значительный урон не только крупным корпорациям, но и малому и среднему бизнесу [1][2]. В результате утечек данных, взломов и других атак компании могут понести серьёзные финансовые потери, что в ряде случаев приводит к банкротству. Эти последствия оказывают негативное влияние на экономику в целом, замедляя её рост и вызывая сокращение рабочих мест.

Особенно тревожит тот факт, что киберпреступность угрожает национальной безопасности. В последние годы участились случаи атак на государственные учреждения и объекты критической инфраструктуры, что может привести к серьёзным последствиям для работы важных отраслей и безопасности граждан. Кибератаки на системы здравоохранения, энергетики, транспорта и обороны могут вызвать массовые сбои и поставить под угрозу жизни и здоровье людей [3]. В этом контексте киберпреступность рассматривается как ключевой вызов, требующий повышенного внимания со стороны государств и международных организаций.

Не менее важным аспектом серьёзности проблемы является угроза личной безопасности граждан. Кибермошенничество, кражи личных данных, шантаж — все эти виды преступлений ставят под угрозу не только финансовое благополучие людей, но и их психическое здоровье. Жертвы кибератак могут испытывать серьёзные психологические проблемы, такие как депрессия и тревожные расстройства, что требует от общества разработки комплексных мер поддержки и защиты.

### **Виды киберпреступлений и меры защиты от них**

В 2023 году мир цифровых технологий сталкивается с множеством новых и усложняющихся угроз безопасности. Кибератаки стали распространённым явлением, способным причинить значительный ущерб как отдельным пользователям, так и организациям. Понимание различных типов кибератак и способов защиты от них — важная задача для всех, кто использует интернет.

Одной из самых опасных атак являются "атаки человека посередине" (MITM). В этом сценарии злоумышленник перехватывает сообщения между двумя сторонами, позволяя ему получать доступ к конфиденциальной информации, такой как пароли и номера кредитных карт. Эти атаки могут произойти незаметно, поэтому важно использовать зашифрованные средства связи и избегать общественного Wi-Fi, особенно для выполнения чувствительных операций.

Фишинг и его более целевая версия, целевой фишинг, также представляют серьёзную угрозу. Злоумышленники отправляют сообщения, которые выглядят как от доверенных источников, например, от банков или известных компаний, чтобы выманить

личные данные. В случае целевого фишинга преступники исследуют своих жертв, чтобы сделать свои сообщения еще более правдоподобными. Защититься от таких атак можно, проверяя отправителей и избегая подозрительных ссылок.

Не менее коварными являются "атаки на ходу" (Drive-By), при которых вредоносный код проникает в устройство жертвы просто при посещении заражённого сайта. Злоумышленники используют уязвимости в программном обеспечении, и для защиты здесь крайне важно регулярно обновлять систему и антивирусное программное обеспечение.

Атаки с использованием ботнетов представляют собой ещё одну распространённую угрозу. В этом случае злоумышленники управляют сетью заражённых устройств без ведома их владельцев. Эти устройства, называемые "ботами", могут использоваться для проведения атак, рассылки спама или кражи данных. Чтобы защитить свои устройства от попадания в ботнет, необходимо следить за обновлениями программного обеспечения и проявлять осторожность при открытии ссылок или загрузке файлов из непроверенных источников.

Социальная инженерия — это метод манипуляции, при котором злоумышленники пытаются обмануть жертву, заставляя её раскрывать конфиденциальную информацию. Это может быть сделано через телефонные звонки, электронные письма или даже личные встречи. Для защиты от таких атак следует всегда проверять личность тех, кто запрашивает вашу информацию, и не поддаваться давлению, особенно если ситуация кажется подозрительной.

SQL-инъекции — это атаки, при которых злоумышленники используют уязвимости в веб-приложениях для внедрения вредоносных команд в базу данных. Это позволяет им получать доступ к конфиденциальной информации или даже управлять всей системой. Защититься от таких атак можно, применяя безопасное написание кода и тщательно проверяя вводимые пользователями данные, используя параметризованный ввод совместно с экранированием служебных символов.

Малварь, или вредоносное программное обеспечение, представляет собой широкий класс угроз, которые могут нанести серьёзный ущерб вашим устройствам и украсть личные данные. Различные виды малвари, такие как вирусы, черви и трояны, могут попасть на ваше устройство через сомнительные сайты, ссылки или вложения в электронной почте. Для защиты от вредоносного ПО необходимо использовать надежные антивирусные программы, а также соблюдать правила цифровой гигиены.

Межсайтовый скриптинг (XSS) позволяет злоумышленникам внедрять вредоносный JavaScript код на сайты, который затем выполняется в браузере других пользователей. Это может привести к краже данных, таких как куки и конфиденциальная информация. Для предотвращения такого рода атаки разработчики должны использовать безопасные практики написания кода, везде обрабатывая пользовательский ввод, а пользователи должны быть осторожны при переходе по ссылкам на подозрительных сайтах.

Атаки на пароли включают в себя множество техник, направленных на взлом ваших учетных записей. Злоумышленники могут использовать "грубой силы" для подбора паролей или воспользоваться списками распространённых паролей. Чтобы защитить свои аккаунты, важно использовать длинные и сложные пароли, а также уникальные пароли для каждой учетной записи, соблюдая все предусмотренные политикой паролей правила. Для удобства рекомендуется также использовать менеджеры паролей.

Атаки "отказ в обслуживании" (DoS) направлены на перегрузку ресурсов, что делает их недоступными для пользователей. Злоумышленники могут создавать огромный объем трафика, чтобы заблокировать доступ к веб-сайту или сервису. Защититься от таких атак могут компании, используя специализированные инструменты для фильтрации

трафика и внедряя системы резервирования для соблюдения отказоустойчивости корпоративных сетей.

Распространенные атаки "распределённого отказа в обслуживании" (DDoS) применяют армию заражённых устройств для создания перегрузки на целевом ресурсе. Эти атаки могут быть использованы для создания хаоса или отвлечения внимания от других киберугроз. В таких случаях необходима сильная киберзащита и системы мониторинга.

Внутренние атаки и утечки данных происходят, когда сотрудники или партнёры злоупотребляют своим доступом к системам организации. Для предотвращения таких угроз компании должны внедрять строгие меры контроля доступа, проводить обучение сотрудников и следить за соблюдением политик безопасности.

Крипто-джеккинг — это метод, при котором злоумышленники используют ресурсы вашего устройства для майнинга криптовалюты без вашего ведома. Чтобы защититься от этого, необходимо следить за безопасностью вашего устройства и избегать загрузки подозрительных файлов [4].

### **Взгляд на будущее кибербезопасности**

В мире, где цифровые технологии продолжают стремительно развиваться, вопросы кибербезопасности становятся всё более актуальными. Будущее этой области обещает быть сложным и многогранным, ведь угрозы, с которыми сталкиваются пользователи и организации, будут лишь увеличиваться как в количестве, так и в сложности. Прогнозы показывают, что кибератаки станут более организованными, высокотехнологичными и целенаправленными, что требует от всех нас повышения уровня осведомлённости и готовности к защите.

Одним из главных факторов, влияющих на будущее кибербезопасности, является рост числа подключённых устройств. С появлением Интернета вещей (IoT) количество уязвимых точек в нашей цифровой инфраструктуре увеличивается в геометрической прогрессии. Устройства, от бытовой техники до умных городов, могут стать целями для злоумышленников, что делает необходимым внедрение новых стандартов безопасности и протоколов защиты данных.

Искусственный интеллект (ИИ) также играет двойственную роль в кибербезопасности. С одной стороны, ИИ может значительно улучшить защитные механизмы, анализируя большие объёмы данных и выявляя подозрительное поведение в реальном времени. С другой стороны, злоумышленники могут использовать те же технологии для разработки более совершенных атак, которые будут сложно обнаружить. Это создаёт постоянную гонку между защитниками и нападающими, где каждый шаг требует быстрой адаптации и инновационных решений.

Параллельно с технологическими изменениями, важным аспектом остаётся человеческий фактор. Необходимо активно обучать пользователей основам кибербезопасности, ведь именно ошибки и неосведомлённость людей часто становятся причиной успешных атак. Создание культуры безопасности в организациях должно стать приоритетом, чтобы каждый сотрудник понимал свою роль в защите информации.

Взаимодействие между государственными и частными секторами также будет играть ключевую роль в будущем кибербезопасности. Обмен информацией о новых угрозах и совместные усилия в разработке решений помогут создать более защищённую цифровую среду. Международное сотрудничество станет особенно важным, поскольку многие кибератаки не имеют границ и требуют совместных действий на глобальном уровне.

Таким образом, будущее кибербезопасности обещает быть сложным и многогранным, требуя от всех участников цифрового пространства постоянной готовности к новым вызовам. Инновационные технологии, обучение пользователей, сотрудничество между секторами и акцент на человеческий фактор будут ключевыми элементами в обеспечении безопасного и защищённого цифрового мира. Важно помнить,

что безопасность — это не конечная цель, а постоянный процесс, требующий внимания и усилий со стороны всех нас.

### Заключение

В статье были рассмотрены основные и наиболее часто встречающиеся виды киберпреступлений, а вместе с этим были представлены методы для защиты от них. Был проведен краткий экскурс в тему киберпреступности и кибербезопасности для обеспечения сохранности персональных данных, финансов и здоровья людей.

\*\*\*

1. Красов А. В., Виткова Л. А., Пестов И. Е. "Behavioral analysis of resource allocation systems in cloud infrastructure" //2019 International Russian Automation Conference (RusAutoCon). – IEEE, 2019. – С. 1-5.
2. Криптовбиржа BingX потеряла \$52 млн из-за атаки хакеров. Investing.com. 2024. URL: <https://ru.investing.com/news/cryptocurrency-news/article-2510977>
3. Самые крупные взломы и сливы в истории игровой индустрии. Яндекс Дзен. 2023. URL: <https://dzen.ru/a/ZAHOK-pQjRbSrYLj>
4. В Германии пациентка умерла из-за кибератаки на больницу. Vaza. 2020. URL: <https://vaza.io/posts/a5d7f457-0a0a-482c-b0ca-c3d389dca352>
5. 13 Types of Cyber Attacks You Should Know in 2023. freeCodeCamp. 2023. URL: <https://www.freecodecamp.org/news/types-of-cyber-attacks-to-know/>

**Биджиева С.Х., Байрамукова М.И., Гериева К.М.**

### Анализ метрик оценки качества генеративных языковых моделей

*Северо-Кавказская государственная академия  
(Россия, Черкесск)*

doi: 10.18411/trnio-12-2024-851

### Аннотация

В статье рассматриваются метрики оценки качества генеративных языковых моделей. Проанализированы такие метрики как BLEU (Bilingual Evaluation Understudy), ROGUE (Recall-Oriented Understudy for Gisting Evaluation), METEOR (Metric for Evaluation of Translation with Explicit ORdering), которые позволяют численно оценить качество сгенерированных текстов.

**Ключевые слова:** генеративные языковые модели, оценка качества, метрики оценки, точность, полнота.

### Abstract

The article discusses popular metrics for assessing the quality of generative language models. Such metrics as BLEU (Bilingual Evaluation Understudy), ROGUE (Recall-Oriented Understudy for Gisting Evaluation), METEOR (Metric for Evaluation of Translation with Explicit ORdering) are analyzed, which allow for a numerical assessment of the quality of the generated texts.

**Keywords:** generative language models, quality assessment, assessment metrics, accuracy, recall.

Генеративные языковые модели — это технологии искусственного интеллекта, которые используют машинное обучение для генерации текстового контента. Использование генеративных языковых моделей позволяет автоматизировать многие процессы, связанные с производством больших объемов уникального или стандартизированного текстового контента.

Наиболее известные типы генеративных языковых моделей - это:

- ЧатGPT (Open AI) - одна из самых известных моделей, она обучается на наборах данных, включающих более 175 миллиардов параметров, способен генерировать текст на 60 языках по запросу пользователя и отвечать на вопросы оператора.

- BLOOM (исследовательская группа BigScience)- модель нейронной сети, обученная по 176 миллиардам параметров и способная генерировать текст на 46 языках и 13 языках программирования.
- BERT (Google)- используется поисковыми системами для получения более релевантных результатов и улучшения понимания запросов пользователей и текущего контекста.
- YaLM 100B (Компания «Яндекс»)- языковая модель, обученная на 2 терабайтах текстовых данных со 100 миллиардами параметров, умеет создавать тексты и проводить операции по обработке контента на русском и английском языках.

Оценка качества сгенерированного текста является важной задачей, поскольку позволяет определить, насколько эффективны различные сгенерированные методы, и найти лучшую модель.

Для оценки качества генеративных языковых моделей используются различные метрики. Выбор метрики напрямую зависит от конкретной задачи и общих требований к качеству генерации текста.

Одним из наиболее распространенных показателей при измерении точности генеративных языковых моделей является BLEU (Bilingual Evaluation Understudy). Она основана на подсчете количества совпадений между текстом, предложенным моделью, и эталонным текстом. Чем больше значение BLEU, тем выше точность сгенерированного текста [2].

Еще одна популярная метрика, которую можно использовать для измерения точности сгенерированного текста, - это ROUGE (Recall-Oriented Understudy for Gisting Evaluation). При этом учитывается количество общих слов и словосочетаний между предлагаемым текстом, между моделируемым и эталонным текстом.

Эти метрики при анализе текста на сходство работают с понятием n-грамм - последовательности из n последовательных слов в анализируемом тексте.

При оценке степени сходства сгенерированного текста с эталонным текстом с помощью метода сопоставления n-грамм вычисляется точность использования n-грамм в сгенерированном тексте.

Критерий «точность» (precision) представляет соотношение сгенерированных n-грамм относительно всех совпадающих n-грамм в исходном тексте и общего количества n-грамм в сгенерированном тексте. Основная формула, используемая для расчета метрики BLEU, выглядит следующим образом:

$$BLEU = \left[ P_B \left( \prod_{i=1}^n \frac{k_i}{l_i} \right)^{\frac{1}{n}} \right] * 100\%,$$

Где  $i$  - длина n-грамм;  $k$  - количество совпадающих n-грамм у исходного и сгенерированного текстов;  $l$  - общее количество n-грамм сгенерированного текста. Отношение совпадающих n-грамм к общему количеству также называют точностью (precision).  $P_B$ - это штраф за краткость, применяемый при расчете метрики, который рассчитывается следующим образом:

$$P_B = \begin{cases} 1, & c > r \\ e^{(1-r/c)}, & c \leq r \end{cases},$$

где  $r$  - количество слов исходного текста;  $c$  - количество слов в сгенерированном тексте.

Развитием метрик BLEU, ориентированных исключительно на точность, является семейство метрик ROUGE, которые дополнительно фокусируются на расчете параметров «полноты» (recall). Параметр полноты представляет собой долю всех совпадающих n-грамм в исходном и сгенерированном тексте относительно общего количества n-грамм в исходном тексте.

Общая формула метрик семейства ROUGE имеет следующий вид:

$$ROUGE = C_n / (T_n),$$

где  $n$  - число n-грамм;  $C_n$  - количество n-граммов, которые совпадают между сгенерированным и эталонным текстом;  $T_n$  - количество n-граммов в исходном тексте.

Помимо ROUGEn, существуют и другие конкретные метрики, которые можно использовать для оценки качества создаваемого текста. Отдельно можно выделить:

- ROUGE-L, который использует длину самой длинной общей последовательности слов как меру сходства между двумя текстами.
- ROUGE-S, который идентифицирует похожие n-граммы, возможно, разделенные несколькими словами, на основе удаления случайного числа n-грамм.

Еще одной метрикой для измерения качества генерируемого текста является метрика METEOR (Metric for Evaluation of Translation with Explicit ORdering) во многом похожая на ROUGE, однако она использует фиксированный механизм генерации выравнивания и сравнения синонимов, что позволяет правильно сравнивать слова-синонимы, а также учитывать положение слова в предложении, улучшая качество оценки. Кроме того, METEOR, в отличие от метрики BLEU, использует функции сопоставления синонимов и точного сопоставления слов. Эта метрика была разработана для решения проблем, обнаруженных в метрике BLEU, а также для обеспечения хорошей корреляции с оценками экспертов на уровне словосочетаний или предложений.

Выполнение метрики на уровне словосочетаний дало корреляцию с человеческим решением 0,964, тогда как метрика BLEU составила 0,817 на том же наборе входных данных. На уровне предложений максимальная корреляция с экспертными суждениями составила 0,403[2].

Кроме того, существуют метрики, которые учитывают семантическую близость предложенного моделью текста к эталонному тексту, такие как BERTscore, SBERT и др.

Подводя итоги, можно сказать, что генеративные модели естественного языка становятся всё более мощными, разрабатываются и корректируются также метрики оценки их качества.

\*\*\*

1. Полтораки, А. В. Анализ существующих архитектур нейронных сетей для генерации текстов естественного языка с целью исследования актуальных техник при создании моделей нейронных сетей / А. В. Полтораки, С. И. Набатов // ИТ-Стандарт. – 2020. – № 3(24). – С. 47-53. – EDN RCERMV.
2. Banerjee, S. and Lavie, A. (2005) «METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments» in Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association of Computational Linguistics (ACL-2005), Ann Arbor, Michigan, June 2005
3. Sai A. B., Mohankumar A. K., Khapra M. M. A survey of evaluation metrics used for NLG systems //ACM Computing Surveys (CSUR). – 2022. – Т. 55. – №. 2. – С. 1-39.
4. Yao X., Zhang Y. Question generation with minimal recursion semantics //Proceedings of QG2010: The Third Workshop on Question Generation. – 2010. – С. 68-75.

**Биджиева С.Х., Байрамукова М.И., Гериева К.М.**  
**Этапы и методы обработки естественного языка**

*Северо-Кавказская государственная академия  
(Россия, Черкесск)*

doi: 10.18411/trnio-12-2024-852

#### **Аннотация**

В статье приводится описание этапов и методов обработки естественного языка (NLP). Отмечается, что работа в области обработки естественного языка (NLP) подразумевает использование вычислительных методов для анализа и понимания человеческого языка. Решение этой задачи предполагает: обработку текста и предварительную обработку, синтаксис и синтаксический анализ в NLP, семантический анализ, извлечение информации, классификацию текста в NLP, генерацию языка, обработку речи, вопрос-ответ, анализ чувств и эмоций в NLP.

**Ключевые слова:** обработка естественного языка, искусственный интеллект, методы NLP, стемминг, лемматизация, сегментация, токенизация, нормализация текста, извлечение признаков, «мешок слов», контролируемое обучение, неконтролируемое обучение.

### Abstract

The article describes the stages and methods of natural language processing (NLP). It is noted that work in the field of natural language processing (NLP) involves the use of computational methods to analyze and understand human language. The solution to this problem involves: text processing and pre-processing, syntax and syntactic analysis in NLP, semantic analysis, information extraction, text classification in NLP, language generation, speech processing, question-answer, analysis of feelings and emotions in NLP.

**Keywords:** natural language processing, artificial intelligence, NLP methods, stemming, lemmatization, segmentation, tokenization, text normalization, feature extraction, "bag of words", supervised learning, unsupervised learning.

Обработка естественного языка (NLP) - одна из самых перспективных областей искусственного интеллекта (ИИ) и это дисциплина, изучающая создание машин, которые могут обрабатывать человеческий язык, или данные, которые напоминают естественный язык.

НЛП используется для решения широкого спектра задач, связанных с языком. В качестве основных задач, которые можно решить с помощью NLP можно выделить: анализ тональности, классификация токсичности, машинный перевод, распознавание именованных сущностей, обнаружение спама, модели исправления грамматических ошибок, тематическое моделирование, генерация текста, автозаполнение, чат-боты, поиск информации, обобщение (извлекающее обобщение и абстрактное обобщение), ответ на вопрос, множественный выбор, открытый домен. Рассмотрим некоторые из них.

Анализ тональности - это процесс определения эмоциональной направленности текста. Как правило, на вход модели классификации тональности поступает фрагмент текста, а на выходе получается вероятность того, что выраженное настроение является положительным, отрицательным или нейтральным. Как правило, эта вероятность основана на характеристиках, созданных вручную, n-граммах слов, характеристиках TF-IDF или использовании моделей глубокого обучения для выявления последовательных долгосрочных и краткосрочных зависимостей.

Классификация токсичности - это раздел анализа настроений, целью которого является не только классификация враждебных намерений, но и классификация конкретных категорий, таких как угрозы, оскорбления, ненормативная лексика и ненависть к определённым личностям. Входными данными для такой модели является текст, а выходными данными, как правило, является вероятность принадлежности к каждому классу токсичности. Модели классификации токсичности можно использовать для модерации и улучшения онлайн-общения путем блокировки оскорбительных, ненормативных комментариев, выявления в языке чувства ненависти, разжигания вражды или проверки документов на наличие клеветы.

НЛП используют различные технологии для предварительной обработки данных, извлечения признаков и моделирования.

Перед тем как модель обработает текст для выполнения конкретной задачи, текст часто необходимо предварительно обработать, чтобы повысить производительность модели или преобразовать слова и символы в формат, понятный модели. При предварительной обработке данных могут использоваться следующие методы.

Стемминг - это неформальный процесс преобразования слов в их базовые формы с помощью эвристических правил («университет», «университеты» и «университетский» - univers). Лемматизация - это более формальный способ поиска корней путём анализа морфологии слова с помощью словарного запаса. Стемминг и лемматизация обеспечиваются такими библиотеками, как spaCy и NLTK .

Сегментация предложений разбивает большой фрагмент текста на лингвистически значимые единицы предложения. Учитывая, что в современном мире есть языки, в

которых конец предложения отмечается точкой, и есть языки, в которых конец предложения не отмечается точкой, этот процесс совсем нетривиальный.

Удаление стоп-слов направлено на удаление наиболее часто встречающихся слов, которые не несут смысловую нагрузку (и так, таким образом и т.п.).

Токенизация разбивает текст на отдельные слова и фрагменты слов. В результате обычно получается индекс слов и токенизированный текст, в котором слова могут быть представлены в виде числовых токенов для использования в различных методах глубокого обучения.

Нормализация текста – это стандартизация текста, включая нормализацию регистра, удаление знаков препинания и исправление орфографических ошибок.

Для текстового представления данных используются следующие методы.

Извлечение признаков: большинство традиционных методов машинного обучения работают с признаками - обычно это числа, которые описывают документ по отношению к содержащему его корпусу, - созданными с помощью «мешка слов», TF-IDF или общих методов извлечения признаков, таких как длина документа, полярность слов и метаданные (например, если у текста есть связанные с ним теги или оценки). Более современные методы включают Word2Vec, GloVe и обучение признакам в процессе тренировки нейронной сети. «Мешок слов» подсчитывает количество появлений каждого слова или n-граммы (комбинации из n слов) в документе.

Частота термина - обратная частота документа (TF-IDF): статистика, отражающая смысл слов в документах по отношению к коллекции документов.

Вложения слов - использование плотного векторного представления слов, при котором семантически похожие слова располагаются ближе друг к другу в векторном пространстве (например, Word2Vec, GloVe).

Функции извлечения – это извлечение значимых характеристик из текстовых данных, в котором используются такие методы как N-граммы - захват последовательностей из N слов для сохранения некоторого контекста и порядка слов; синтаксические особенности - использование тегов частей речи, синтаксических зависимостей и деревьев синтаксического анализа; семантические особенности - использование вложений слов и других представлений для фиксации значения слова и контекста.

На следующем этапе обработки естественного языка осуществляется выбор и обучение модели машинного обучения или глубокого обучения для выполнения конкретных задач обработки естественного языка. Контролируемое обучение предполагает использование размеченных данных для обучения моделей, таких как машины опорных векторов (SVM), случайные леса или модели глубокого обучения, такие как сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN); неконтролируемое обучение - применение таких методов, как кластеризация или тематическое моделирование (например, скрытое распределение Дирихле) к немаркированным данным. Также возможно использование предварительно обученных моделей, таких как BERT, GPT или моделей на основе трансформера, обученных на больших корпусах текстов.

На этапе развертывания обученной модели она используется для составления прогнозов или извлечения информации из новых текстовых данных. Далее осуществляется: оценка производительности алгоритма обработки естественного языка с использованием таких показателей, как точность, достоверность, полнота, оценка F1 и другие; настройка гиперпараметров: настройка параметров модели для повышения производительности; анализ ошибок для понимания слабых сторон модели и повышения ее надежности.

И наконец, этап итераций и улучшения предполагает постоянное совершенствование алгоритма путем включения новых данных, совершенствования

методов предварительной обработки, экспериментов с различными моделями и оптимизации функций.

В настоящее время НЛП стали неотъемлемой частью повседневной жизни и приобретают всё большую востребованность по мере того, как языковые технологии применяются в различных областях, таких как розничная торговля, банковская сфера, медицина и т.д. От базовых задач, таких как токенизация и частеречная маркировка, до продвинутых приложений, таких как анализ настроений и машинный перевод, влияние NLP очевидно в различных областях. Поскольку технология продолжает развиваться, движимая достижениями в области машинного обучения и искусственного интеллекта, потенциал NLP для улучшения взаимодействия человека и компьютера и решения сложных языковых задач остается огромным. Понимание основных концепций и приложений обработки естественного языка имеет решающее значение для тех, кто хочет использовать ее возможности в современном цифровом мире.

\*\*\*

1. Nitschke R. Restoring the Sister: Reconstructing a Lexicon from Sister Languages using Neural Machine Translation. In: Proc. 1st Workshop on Natural Language Processing for Indigenous Languages of the Americas, AmericasNLP 2021. 2021. P. 122 - 130. <http://dx.doi.org/10.18653/v1/2021.americasnlp-1.13>
2. Kazakova M. A., Sultanova A. P. Analysis of natural language processing technology: modern problems and approaches. Advanced Engineering Research, 2022, vol. 22, no. 2, pp. 169-176. <https://doi.org/10.23947/2687-1653-2022-22-2-169-176>
3. <https://www.geeksforgeeks.org/the-future-of-natural-language-processing-trends-and-innovations/>
4. <https://en.wikipedia.org/wiki/SpaCy>
5. [https://en.wikipedia.org/wiki/Natural\\_Language\\_Toolkit](https://en.wikipedia.org/wiki/Natural_Language_Toolkit)

**Ближенский М.В.**

### **Анализ устойчивых уязвимостей в PostgreSQL**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

*doi: 10.18411/trnio-12-2024-853*

#### **Аннотация**

В статье рассматриваются уязвимости PostgreSQL, сохранявшиеся на протяжении нескольких версий, и причины их длительного присутствия. Описаны методы мониторинга и управления уязвимостями, включая сканеры безопасности, аудит и настройку системы. Подчеркивается важность своевременного обновления PostgreSQL для снижения рисков.

**Ключевые слова:** PostgreSQL, уязвимости, обратная совместимость, безопасность данных, мониторинг безопасности, управление уязвимостями.

#### **Abstract**

The article examines the vulnerabilities of PostgreSQL that persisted for several versions, and the reasons for their long-term presence. Vulnerability monitoring and management methods are described, including security scanners, auditing, and system configuration. The importance of timely updating PostgreSQL to reduce risks is emphasized.

**Keywords:** PostgreSQL, vulnerabilities, backward compatibility, data security, security monitoring, vulnerability management.

PostgreSQL не просто реляционная, а объектно-реляционная СУБД. Это даёт ему некоторые преимущества над другими SQL базами данных с открытым исходным кодом, такими как MySQL, MariaDB и Firebird. Логотип PostgreSQL изображен на рисунке 1 ниже.



Рисунок 1. Логотип PostgreSQL.

Фундаментальная характеристика объектно-реляционной базы данных — это поддержка пользовательских объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Это делает Постгрес невероятно гибким и надежным. Среди прочего, он умеет создавать, хранить и извлекать сложные структуры данных. В некоторых примерах ниже вы увидите вложенные и составные конструкции, которые не поддерживаются стандартными РСУБД.

Существует обширный список типов данных, которые поддерживает Постгрес. Кроме числовых, с плавающей точкой, текстовых, булевых и других ожидаемых типов данных (а также множества их вариаций), PostgreSQL может похвастаться поддержкой uuid, денежного, перечисляемого, геометрического, бинарного типов, сетевых адресов, битовых строк, текстового поиска, xml, json, массивов, композитных типов и диапазонов, а также некоторых внутренних типов для идентификации объектов и местоположения логов. Справедливости ради стоит сказать, что MySQL, MariaDB и Firebird тоже имеют некоторые из этих типов данных, но только Постгрес поддерживает их все.

Анализ уязвимостей PostgreSQL, в особенности тех, что присутствовали в системе на протяжении нескольких версий, является необходимым этапом для поддержания безопасности данных, а также для понимания того, насколько платформа устойчива к атакам. Этот анализ помогает выявить системные проблемы, определяет, какие аспекты архитектуры требуют дополнительной защиты, и даёт возможность своевременно реагировать на потенциальные угрозы. PostgreSQL, как популярная объектно-реляционная СУБД, развивается с каждым релизом, добавляя новые функции и улучшая безопасность. Однако некоторые уязвимости, обнаруженные в различных версиях, оказывались устойчивыми и сохранялись в системе через несколько глобальных релизов, что подчеркивает важность их тщательного анализа и регулярного обновления системы. Рассмотрим самые первые и последние версии PostgreSQL в промежутке глобальных релизов (6 - 17), ниже представлен рисунок 2, на котором изображен график, сопоставляющий кол-во уязвимостей и версию.

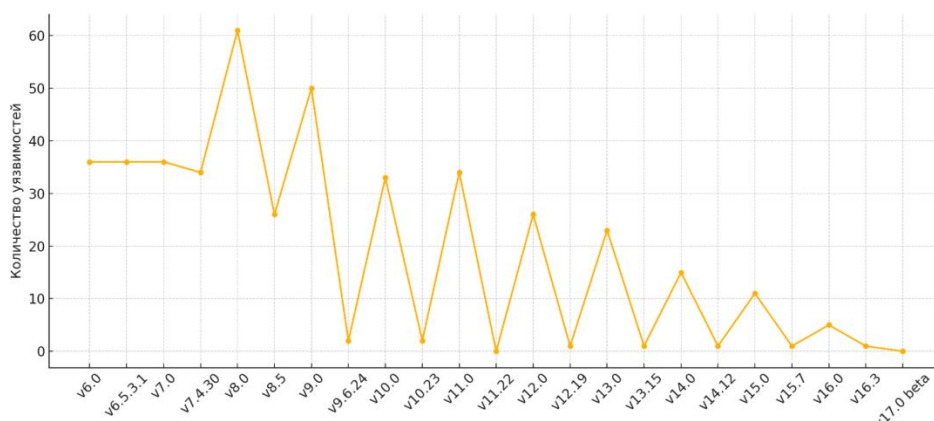


Рисунок 2. Количество уязвимостей в версиях PostgreSQL.

При рассмотрении глобальных релизов PostgreSQL с 6 по 17 версии можно заметить, что количество уязвимостей изменялось от релиза к релизу. В некоторых

версиях обнаруживалось значительное число уязвимостей, таких как в версиях v8.0 (61 уязвимость) и v9.0 (50 уязвимостей), что могло быть связано с увеличением функционала системы и сложностью её архитектуры. Напротив, в более поздних версиях, таких как v16 и v17, число уязвимостей значительно сократилось, что свидетельствует об усилении мер безопасности и улучшении контроля качества кода.

Среди уязвимостей, сохраняющихся на протяжении нескольких версий, можно выделить следующие, представленные в таблице 1.

Таблица 1

*Уязвимости для PostgreSQL, сохранявшиеся на протяжении нескольких версий дольше остальных.*

<i>Идентификатор уязвимости</i>	<i>Затронутые версии</i>
<i>CVE-2021-3393</i>	<i>6.0-13.0</i>
<i>CVE-2021-23214</i>	<i>6.0-9.0, 10.0, 11.0, 12.0, 13.0, 14.0</i>
<i>CVE-2020-25695</i>	<i>6.0-9.0, 10.0, 11.0, 12.0, 13.0</i>
<i>CVE-2020-25694</i>	<i>6.0-9.0, 10.0, 11.0, 12.0, 13.0</i>
<i>CVE-2024-7348</i>	<i>12.0-16.3</i>

- CVE-2021-3393: Эта уязвимость затрагивала системы начиная с версии 6.0 до версии 13.0, присутствуя в 13 версиях. Её устойчивость указывает на сложности в её обнаружении или исправлении, несмотря на ряд обновлений.
- CVE-2021-23214: Оставалась актуальной для версий от 6.0 до 9.0, а также с 10.0 до 14.0, встречаясь в общей сложности в 12 версиях.
- CVE-2020-25695 и CVE-2020-25694: Обе уязвимости существовали в версиях от 6.0 до 9.0 и затем продолжали появляться в версиях от 10.0 до 13.0, что подчеркивает их устойчивость и необходимость постоянного мониторинга безопасности.
- CVE-2024-7348: Эта уязвимость была актуальна в версиях с 12.0 по 16.3, сохраняясь в 10 версиях и потенциально представляя угрозу безопасности до недавнего времени

Продолжительное присутствие некоторых уязвимостей в PostgreSQL на протяжении нескольких версий может быть обусловлено рядом факторов, связанных с архитектурой системы, сложностями в обнаружении проблем и требованиями к обратной совместимости.

Во-первых, PostgreSQL как объектно-реляционная база данных поддерживает сложные структуры данных и богатый набор функций, которые делают её универсальной и гибкой. Однако такая сложная архитектура затрудняет обнаружение и исправление уязвимостей, особенно если они возникают в глубоко интегрированных компонентах системы. В результате, проблемы, затрагивающие базовые механизмы, такие как управление транзакциями, индексация или обработка данных, могут быть трудными для диагностики и устранения.

Во-вторых, PostgreSQL — это проект с открытым исходным кодом, в котором участвует большое сообщество разработчиков. В таких условиях уязвимости могут оставаться незамеченными или недооценёнными, особенно если они касаются узкоспециализированных функций или сценариев использования, которые не часто применяются. В некоторых случаях уязвимости могут считаться малозначимыми или сложными для эксплуатации, что откладывает их исправление до появления более приоритетных задач.

Наконец, одной из ключевых причин длительного присутствия уязвимостей является требование обратной совместимости. PostgreSQL активно используется в различных системах, и многие компании зависят от её стабильности и предсказуемости. Внедрение значительных изменений, направленных на устранение уязвимостей, может привести к нарушениям работы систем, построенных на предыдущих версиях. Поэтому разработчики часто вынуждены искать компромисс между исправлением уязвимости и

поддержанием совместимости с более ранними версиями. Это может замедлить процесс устранения уязвимостей, особенно если для этого требуется значительное изменение архитектуры.

Таким образом, длительное присутствие уязвимостей в PostgreSQL можно объяснить совокупностью архитектурных особенностей, сложности их обнаружения и устранения, а также необходимостью в поддержании обратной совместимости, что требует тщательного подхода к вопросам безопасности и постоянного мониторинга системы.

Для обеспечения безопасности PostgreSQL и минимизации рисков эксплуатации уязвимостей рекомендуется использовать комплексный подход, включающий следующие меры, отраженные в таблице 2.

Таблица 2  
*Рекомендации по обнаружению и управлению уязвимостями в PostgreSQL.*

<i>Мера</i>	<i>Описание</i>
<i>Автоматизированные сканеры безопасности</i>	<i>Использование инструментов (Nessus, OpenVAS, Qualys, Rapid7) для регулярного сканирования на уязвимости.</i>
<i>Мониторинг обновлений безопасности</i>	<i>Отслеживание обновлений PostgreSQL через официальные Security Advisories или RSS-ленты для своевременного реагирования на новые угрозы.</i>
<i>Настройка безопасности</i>	<i>Минимизация прав доступа, использование SSL для соединений, настройка брандмауэра, ограничение IP-адресов.</i>
<i>Регулярный аудит и тестирование системы</i>	<i>Проведение аудита конфигураций и пентестов (например, Metasploit, SQLmap) для раннего выявления уязвимостей.</i>
<i>Обучение и повышение осведомленности</i>	<i>Обучение сотрудников принципам безопасности, минимально необходимым правам доступа и предотвращению SQL-инъекций.</i>

Внедрение этих методов позволяет создать надёжную систему мониторинга и управления уязвимостями для PostgreSQL. Регулярное обновление системы, аудит и тестирование помогают минимизировать риски и повышают общую безопасность базы данных, снижая вероятность её успешной эксплуатации злоумышленниками.

В заключении можно отметить, что своевременное обновление системы управления базами данных (СУБД) PostgreSQL и строгое соблюдение рекомендаций по безопасности играют ключевую роль в предотвращении рисков, связанных с уязвимостями. Постоянное развитие технологий и рост числа угроз требуют от организаций повышенного внимания к безопасности своих данных. Устаревшие версии, в которых сохраняются уязвимости, могут стать лёгкой целью для злоумышленников, тогда как регулярные обновления помогают оперативно закрывать выявленные бреши и повышают общую устойчивость системы.

Следование рекомендациям по безопасности, таким как использование автоматизированных сканеров, настройка доступа, проведение регулярных аудитов и тестирования, позволяет минимизировать риски и снижает вероятность успешной эксплуатации уязвимостей. Принятие проактивного подхода к защите данных в PostgreSQL обеспечивает не только безопасность системы, но и стабильность её работы, что особенно важно для организаций, работающих с критически важной информацией.

\*\*\*

1. А. М. Гельфанд, В. В. Сигачева, А. В. Архипов, Л. К. Сиротина. Анализ и управление рисками информационной безопасности объекта критической информационной инфраструктуры // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. – 2023. – № 3. – С. 21-27. – DOI 10.46418/2079-8199\_2023\_3\_3. – EDN BKGRAY.
2. Пестов, И. Е. влияние механизмов обеспечения информационной безопасности на быстродействие кластера обработки больших данных / И. Е. Пестов, А. Д. Федотовская // Актуальные проблемы инфотелекоммуникаций в науке и образовании : Сборник научных статей XIII Международной научно-технической и научно-методической конференции в 4 т., Санкт-Петербург, 27–28 февраля

- 2024 года. – Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2024. – С. 597-601. – EDN CQYLYT.
3. Защита информации в базах данных / Э. В. Бирих, Л. А. Виткова, В. В. Гореленко, Д. Б. Казаков // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2017) : Сборник научных статей VI Международной научно-технической и научно-методической конференции. В 4-х томах, Санкт-Петербург, 01–02 марта 2017 года / Под редакцией С.В. Бачевского. Том 2. – Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2017. – С. 89-92. – EDN YRQKPI.
  4. Алехин Р. В. и др. Исследование критической уязвимости сервиса аутентификации и последствий для медицинских учреждений, относящихся к субъектам критической информационной инфраструктуры // Офтальмохирургия. – 2022. – №. 4s. – С. 115-122.

**Гараев А.Х., Агаев А.Х., Шахбазова М.С.**

**Безопасность баз данных: защита данных с помощью SQL**

*Чеченский государственный университет имени А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-12-2024-854

**Аннотация**

Тема безопасности баз данных приобретает особое значение в условиях роста объемов информации и распространения цифровых технологий. Современные реляционные базы данных играют ключевую роль в управлении данными, необходимыми для работы организаций и принятия стратегических решений. В работе рассматриваются фундаментальные методы защиты данных в SQL-среде: надежная аутентификация и авторизация, шифрование данных в процессе их передачи и хранения, а также механизмы предотвращения атак с использованием SQL-инъекций.

**Ключевые слова:** база данных, защита, сервер, пароли, хеш.

**Abstract**

The topic of database security is of particular importance in the context of growing volumes of information and the proliferation of digital technologies. Modern relational databases play a key role in the management of data necessary for the work of organizations and strategic decision-making. The paper considers fundamental methods of data protection in SQL environment - reliable authentication and authorization, data encryption during its transmission and storage, as well as mechanisms to prevent SQL injection attacks.

**Keywords:** database, security, server, passwords, hash.

База данных представляет собой структурированную систему для хранения информации, к которой можно получить доступ, а также проводить с ней анализ, обновление и перемещение. Компьютеризированные базы данных впервые появились в 1960-х годах и с тех пор стали основой для множества процессов — от анализа данных и бизнес-операций до предоставления онлайн-услуг, таких как банковское обслуживание, социальные сети, покупки и электронная почта.

Для работы с базами данных используется система управления базами данных (СУБД) — программный инструмент, который позволяет взаимодействовать с базой данных, добавлять, изменять или удалять данные. СУБД предоставляет интерфейс для работы с содержимым базы данных. Существуют разные типы СУБД: реляционные, не реляционные и иерархические.

Основные типы баз данных

– Иерархические базы данных

Иерархические базы данных были первой формой организации данных, их структура напоминает дерево с одним родительским элементом и несколькими дочерними. Каждому родителю может соответствовать несколько дочерних объектов, тогда как у дочернего объекта есть только один родитель. Эти базы данных отличаются

высокой скоростью и эффективностью работы, но они жестко структурированы, что ограничивает их гибкость.

– Реляционные базы данных

Реляционные базы данных — одна из самых популярных форм хранения данных, которая связывает таблицы между собой с помощью ключей. Например, одна таблица может хранить данные о пользователе (имя, дата рождения, номер клиента), а другая — информацию о покупках (номер клиента, товар, стоимость). Связь между таблицами осуществляется через общий ключ — номер клиента, что позволяет эффективно управлять связанными данными.

– Не реляционные или NoSQL базы данных

Не реляционные базы данных появились как ответ на возросшую сложность веб-приложений и необходимость более гибкой структуры. Они не ограничены реляционной моделью и могут хранить данные в различных форматах, таких как неструктурированный текст, документы или графы. В отличие от реляционных баз данных, нереляционные более адаптивны к изменениям и подходят для работы с большими объемами данных.

– Облачные базы данных

Облачные базы данных размещаются в гибридной или облачной среде и доступны через интернет. Это решение позволяет работать с данными удаленно, не требуя локального оборудования для хранения больших объемов информации. Облачные базы данных можно создать самостоятельно или использовать платные услуги для их развертывания, что особенно удобно для масштабируемых решений.

– Централизованные базы данных

Централизованные базы данных хранятся на одном компьютере или сервере. Пользователи получают к ним доступ через подключенные к сети устройства, однако сама база данных функционирует из одного места. Такой подход особенно удобен для крупных компаний и организаций, которые делают акцент на безопасность и простоту управления данными.

– Распределенные базы данных

Распределенные базы данных расположены на нескольких устройствах и могут быть развернуты как на одной площадке, так и на разных узлах сети. Преимущество такого подхода заключается в том, что при отказе одного из компьютеров остальные устройства продолжают функционировать, обеспечивая доступ к данным и стабильность системы.

– Объектно-ориентированные базы данных

Объектно-ориентированные базы данных представляют данные в виде объектов и классов. Объекты могут включать такие данные, как имена или видеозаписи, а классы объединяют объекты в группы. Благодаря хранению данных в виде объектов пользователям не требуется распределять их по таблицам, что упрощает работу со связями между переменными и ускоряет анализ.

Базы данных играют ключевую роль в работе множества приложений, обеспечивая хранение и защиту данных, которые поддерживают бизнес-процессы и стратегическое планирование. Учитывая важность этой функции, безопасность баз данных становится приоритетом. В основе управления реляционными базами данных лежит SQL, поэтому знание эффективных методов защиты SQL-кода необходимо для защиты данных от несанкционированного доступа и других угроз. В этой статье рассмотрим ключевые стратегии безопасности SQL, включая надежную аутентификацию, шифрование и предотвращение SQL-инъекций. Также будут приведены примеры кода для наглядного применения этих мер.

Основой безопасности базы данных является аутентификация и авторизация. Важно не только обеспечить надежное подтверждение личности пользователя, но и определить границы его доступа с помощью продуманной политики контроля. Одним из

базовых методов защиты служат строгие правила для паролей, такие как требование минимальной длины, использование разнообразных символов и регулярное изменение.

Пример создания пользователя с надежным паролем:

```
– CREATE LOGIN [NewUser] WITH PASSWORD = 'StR0nG_p@ssW0rd!';
```

Сложность и уникальность пароля в этом примере делают его устойчивым к атакам методом перебора и словарным атакам. Помимо требования сложных паролей, рекомендуется хранить пароли в зашифрованном виде на уровне приложения. Использование алгоритмов хеширования, таких как bcrypt или Argon2, дополнительно защищает пароли. Пример с bcrypt:

```
– password = "my_secure_password"  
– salt = bcrypt.gensalt()  
– hashed_password = bcrypt.hashpw(password.encode('utf-8'), salt)
```

В этом примере bcrypt.gensalt() генерирует уникальную соль, что делает идентичные пароли разных пользователей различными при хешировании. Функция bcrypt.hashpw() хэширует пароль, объединяя его с солью, что защищает от атак радужными таблицами. Хранение соли вместе с хешем в базе данных позволяет системе при входе пользователя использовать сохранённую соль для повторного хеширования пароля и сравнения с оригинальным хешем. Таким образом, даже при компрометации базы данных злоумышленник получает только хеш, который трудно взломать.

Шифрование данных при передаче — ещё один важный аспект безопасности, защищающий данные, перемещающиеся между приложением и сервером базы данных. Шифрование SSL/TLS делает данные недоступными для перехвата и атак типа «человек посередине». В SQL Server для включения SSL-шифрования выполните следующие шаги:

1. Установите на сервере сертификат SSL, полученный от доверенного центра сертификации.
2. В SQL Server Configuration Manager перейдите в раздел "Протоколы для [имя экземпляра]" и откройте свойства.
3. Включите параметр «Принудительное шифрование» и выберите сертификат для SSL.
4. Перезапустите SQL Server для применения изменений.

Эти шаги обеспечивают шифрование всех соединений SQL Server, защищая данные от перехвата и атак. SSL-сертификат также выполняет проверку подлинности сервера, гарантируя клиенту, что он подключен к надёжному источнику, а не к серверу-злоумышленнику.

Аутентификация и управление доступом

Аутентификация и контроль доступа — базовые меры защиты данных. Для повышения безопасности используются сложные пароли и двухфакторная аутентификация, которая создает дополнительный уровень защиты. Каждому пользователю назначаются права доступа, которые ограничивают его действия только необходимыми для работы операциями. Например, одному пользователю может быть разрешено только чтение данных, в то время как другому — их редактирование.

Шифрование данных

Шифрование защищает данные от несанкционированного доступа, делая их нечитаемыми для тех, кто не имеет соответствующих прав. Этот метод применяется как к данным в базе, так и к передаваемым данным. Для защиты данных в процессе передачи используется SSL/TLS, который предотвращает их перехват. В самой базе данных часто применяется шифрование на уровне столбцов, что особенно важно для конфиденциальных данных, таких как банковские реквизиты и персональная информация.

Параметризованные запросы

Параметризованные запросы — метод, снижающий риск SQL-инъекций. Такой подход предполагает создание структуры запроса заранее с возможностью вставки

пользовательских данных в виде параметров, предотвращая их выполнение в качестве SQL-кода. Этот метод широко используется в веб-приложениях для блокировки вредоносного кода.

#### Мониторинг и аудит

Регулярный мониторинг активности базы данных помогает выявлять подозрительные действия и своевременно реагировать на угрозы. Ведение журналов операций позволяет отслеживать действия пользователей, анализировать инциденты, повышать безопасность и оперативно выявлять уязвимости.

SQL-инъекции представляют собой один из распространенных видов атак, при которых злоумышленники внедряют свой SQL-код в запросы, отправляемые серверу базы данных, чтобы получить доступ к конфиденциальной информации или изменить данные. Вредоносный код может встраиваться в поля ввода — например, в форму входа или строку поиска. При недостаточной фильтрации ввода сервер обрабатывает введенный текст как часть команды, что позволяет злоумышленнику манипулировать запросами.

Простейшая SQL-инъекция может позволить обойти систему аутентификации или извлечь данные из скрытых таблиц, если сервер не отделяет введенные данные от команд SQL. Один из примеров — добавление кода типа OR '1'='1', из-за чего условие запроса всегда будет истинным, и сервер вернет данные независимо от введенных параметров.

Для предотвращения SQL-инъекций применяются такие методы, как подготовленные (параметризованные) запросы и регулярная очистка данных пользователя. Подготовленные запросы разбивают SQL-код и пользовательские данные на отдельные части, чтобы сервер не интерпретировал данные как команды. Дополнительной защитой служит использование процедур очистки ввода, которая проверяет текстовые строки на наличие потенциально опасных символов, исключая их перед обработкой сервером. Эти меры защищают базу данных от SQL-инъекций, минимизируя риск утечек и несанкционированного доступа к данным.

\*\*\*

1. Ульман, Д., Уидом, Дж. Введение в системы баз данных. — М.: Вильямс, 2015. — 1248 с.
2. Прахар, К., Шварц, Б., Зайцев, П. Высокопроизводительный MySQL. Оптимизация, репликация, резервное копирование. — СПб.: Питер, 2013. — 800 с.
3. Фейгенбаум, Дж. Защита баз данных: принципы и практика. — М.: Лаборатория знаний, 2014. — 432 с.
4. Корт, Дж., Коннор, К. SQL для профессионалов. — М.: Диалектика, 2019. — 752 с.
5. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы и программный код на языке С. — М.: Триумф, 2016. — 816 с.
6. Бейли, М., Поллард, Н. SQL-инъекции и методы их предотвращения. — СПб.: Питер, 2012. — 384 с.

**Гараев А.Х., Гараев М.Х., Шахбазова М.С.**

#### **Методы обработки сенсорных данных**

*Чеченский государственный университет имени А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-12-2024-855

#### **Аннотация**

В статье рассматриваются современные методы обработки данных, получаемых с различных сенсоров. Описываются ключевые подходы к обработке сенсорной информации, включая фильтрацию, нормализацию и классификацию данных. Особое внимание уделяется алгоритмам анализа данных, направленным на повышение точности и эффективности систем, использующих сенсоры. Приводятся примеры применения методов обработки данных в мобильных устройствах, системах мониторинга, а также в навигационных и медицинских технологиях.

**Ключевые слова:** датчики, технологии, смартфон, фильтрация, акселерометр, гироскоп, движение, сигнал.

**Abstract**

The article considers modern methods of processing data obtained from various sensors. Key approaches to processing sensory information are described, including filtering, normalization and classification of data. Particular attention is paid to data analysis algorithms aimed at increasing the accuracy and efficiency of systems using sensors. Examples of the application of data processing methods in mobile devices, monitoring systems, as well as in navigation and medical technologies are given.

**Keywords:** sensors, technologies, smartphone, filtering, accelerometer, gyroscope, movement, signal.

Данные, которые мы получаем от сенсоров мобильных устройств, уже давно стали фундаментом для множества задач, связанных с анализом информации. Мы используем их для отслеживания активности пользователей, анализа окружающей среды, разработки умных систем и ещё для целого ряда целей. Интересно, что формат и структура этих данных напрямую зависят не только от возможностей самих сенсоров, но и от задач, которые перед ними ставят. Например, данные могут варьироваться от простых чисел до массивов, содержащих изображения или звук. Это разнообразие требует особого подхода: мало просто собрать данные, важно правильно их обработать, чтобы они были полезны для анализа.

Начнем с акселерометров — это, пожалуй, один из самых популярных сенсоров. Они записывают ускорение по трём осям (X, Y, Z) и делают это в метрах на секунду в квадрате ( $m/s^2$ ). Но сами цифры мало что значат, если не учитывать временные метки — именно они позволяют упорядочить данные и понять, что происходит в конкретный момент времени. Такие данные помогают не только анализировать движение, но и определять его тип: идёт ли человек, бежит или просто стоит. Ещё один важный сенсор — гироскоп. Он фиксирует угловую скорость, которая помогает понять, как устройство поворачивается в пространстве. Вместе эти два сенсора работают как идеальная пара, дополняя друг друга. Но не все данные от сенсоров такие сложные. Например, датчики приближения выдают очень простую информацию: «1», если объект близко, и «0», если он отсутствует. Датчики освещённости сообщают уровень света в люксах. Такие данные проще в обработке, но менее гибки, ведь они не содержат временной детализации, которая бывает важна для сложного анализа. Тем не менее и такие сенсоры играют свою роль. Например, сочетание данных о движении и изменении уровня освещённости может показать, что человек переместился из светлого помещения в тёмное.

А вот GPS-устройства предоставляют данные совершенно другого уровня. Это и координаты (широта, долгота), и высота над уровнем моря, и скорость, и точность. Всё это объединено в едином потоке. Такие данные идеально подходят для отслеживания перемещений, построения маршрутов или анализа динамики движения. Камеры и микрофоны, в свою очередь, дают нам массивные данные — фотографии, аудиозаписи. Они требуют мощных вычислительных ресурсов, и их обработка может быть настоящим вызовом.

Здесь стоит остановиться на частоте выборки, потому что это один из самых важных параметров. Частота выборки (fs) измеряется в герцах (Hz) и показывает, сколько раз сенсор собирает данные за секунду. Например, акселерометр с частотой 100 Hz записывает 100 троек значений за секунду. Если частота слишком низкая, сенсор может «пропускать» быстрые изменения, что приводит к искажениям. Это особенно критично для анализа движений: скажем, если человек бежит, минимальная частота должна быть 10 Hz, чтобы корректно фиксировать ритм шагов. Если частота слишком низкая, возникает эффект алиасинга — быстрые изменения начинают выглядеть как медленные, и это создаёт серьёзные ошибки. Но в задачах, где изменения происходят медленно, как, например, при мониторинге температуры, высокая частота не нужна. Частота 1 Hz в таких случаях будет вполне достаточной.

При обработке данных начинается с борьбы с шумами. Сенсоры часто «слышат» не только то, что нужно, но и множество лишних сигналов: вибрации, магнитные помехи, изменения температуры. Чтобы избавиться от этого, применяют фильтры. Фильтры нижних частот убирают высокочастотные шумы, такие как вибрации. Фильтры верхних частот, наоборот, отсеивают медленные изменения, например дрейф данных. Полосовые

фильтры позволяют выделять нужный диапазон частот — это полезно, например, для анализа сердечного ритма или ритма шагов. Иногда используют адаптивные фильтры, которые подстраиваются под текущие условия. Простые методы, такие как скользящее среднее, тоже работают неплохо, особенно когда нужен быстрый и понятный результат.

После фильтрации данные нужно преобразовать. Например, акселерометр фиксирует не только движение, но и силу гравитации. Чтобы анализировать только движение, влияние гравитации нужно убрать. Для этого сигнал делят на статическую и динамическую составляющие. С гироскопами всё немного иначе: их данные преобразуются в углы поворота через интегрирование угловой скорости. Без этого данные остаются сырыми и плохо интерпретируемыми.

Ещё одним важным этапом является нормализация. Она нужна, чтобы привести данные к единому масштабу. Например, значения могут варьироваться от 0 до 1 (мин-макс нормализация) или быть стандартизированы (z-оценка), что убирает влияние среднего значения и масштаба. Если этого не сделать, алгоритмы машинного обучения могут ошибаться, интерпретируя данные неправильно. Частотный анализ позволяет смотреть на данные с другой стороны. Вместо анализа во времени сигналы переводятся в частотную область с помощью дискретного преобразования Фурье (ДПФ). Это помогает, например, анализировать шаги или вибрации, находить закономерности или выявлять аномалии.

Таким образом, работа с данными сенсоров — это не просто их сбор. Это целый процесс: фильтрация, преобразование, нормализация и анализ. Каждый шаг важен, потому что от этого зависит, насколько полезными будут данные. Ошибки на любом этапе могут полностью исказить результаты, а значит, подходить к обработке нужно ответственно.

\*\*\*

1. Беляев, Р. Ю., Сидоров, А. А. Технологии обработки сенсорной информации: от теории к практике. — Новосибирск: Наука, 2021. — 290 с.
2. Никитин, М. В., Орлов, П. Н. Обработка сигналов в динамических системах. — Екатеринбург: УрО РАН, 2017. — 328 с.
3. Захаров, Л. И., Трофимов, Н. В. Методы нормализации и фильтрации сенсорных данных. — Томск: Изд-во ТГУ, 2019. — 270 с.
4. Ковалёв, А. Г., Родионов, Ю. Б. Машинное обучение для анализа данных сенсоров. — Воронеж: ВГТУ, 2020. — 310 с.
5. Андреев, О. Е., Фёдоров, С. П. Интеллектуальная обработка данных в современных устройствах. — М.: ДМК Пресс, 2021. — 340 с.
6. Лебедев, А. М., Гончаров, И. В. Оптимизация сенсорных систем. — Самара: Изд-во СамГУ, 2018. — 299 с.
7. Шестаков, Н. В., Борисов, Е. С. Практическое руководство по обработке сенсорной информации. — Ростов-на-Дону: Феникс, 2019. — 245 с.

**Гараев А.Х., Гараев М.Х., Шахбазова М.С.**  
**Принцип работы гироскопического датчика**

*Чеченский государственный университет имени А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-12-2024-856

#### **Аннотация**

В этой статье рассматриваются принципы работы гироскопов, их устройство и ключевые особенности функционирования. Исследуется, как гироскопы фиксируют изменение углового положения, обеспечивая точность ориентации и стабильность движущихся объектов. Особое внимание уделено применению гироскопов в современных устройствах, таких как смартфоны, автомобили и авиационные системы, где они играют важную роль в повышении безопасности и удобства.

**Ключевые слова:** датчики, технологии, смартфон, гироскоп, ускорение, оптимизация, сила Кориолиса, сигнал.

**Abstract**

This article discusses the principles of operation of gyroscopes, their design, and key features of their operation. It examines how gyroscopes detect changes in angular position, ensuring orientation accuracy and stability of moving objects. Particular attention is paid to the use of gyroscopes in modern devices such as smartphones, cars, and aviation systems, where they play an important role in improving safety and convenience.

**Keywords:** sensors, technologies, smartphone, gyroscope, acceleration, optimization, Coriolis force, signal.

В наши дни, с ростом интереса к автономным автомобилям и передовым устройствам, стремительно развивается сектор сенсорных технологий. Особенно заметно внимание приковано к инерционным датчикам — гироскопам и акселерометрам, которые играют ключевую роль в отслеживании движений и ориентации объектов. В этой связи остановимся на гироскопических датчиках, их назначении и принципе работы.

Гироскопический датчик, также известный как датчик угловых скоростей, фиксирует изменения углового положения объекта, преобразуя их в электрические сигналы с использованием силы Кориолиса. Этот датчик измеряет угловую скорость — величину, отражающую, насколько быстро объект меняет свое положение за определенный промежуток времени и выражаемую в градусах в секунду (dps). Например, если объект поворачивается на 360 градусов за 60 секунд, его угловая скорость составит 6 dps.

Чтобы глубже понять работу гироскопических датчиков, нужно рассмотреть природу силы Кориолиса — мнимой силы, действующей на тело, перемещающееся в рамках вращающейся системы. Для иллюстрации этого эффекта можно представить траекторию мяча, движущегося от точки А к точке В на вращающейся поверхности: траектория отклонится в сторону, что наглядно демонстрирует влияние силы Кориолиса на движение объектов.

Во-первых, если точка А и точка В являются фиксированными точками, мяч полетит прямо из точки А в точку В. (рис. 1)

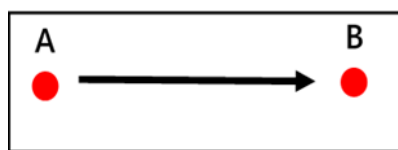


Рисунок 1. Перемещение объекта по стационарным координатам.

Далее предположим, что координаты, соединяющие точки А и В, вращаются. В этом случае шарик полетит из точки А прямо в ту сторону, в которую его выпустили, независимо от того, повернуты координаты или нет, поэтому он полетит с большим отклонением от точки В. (рис. 2)

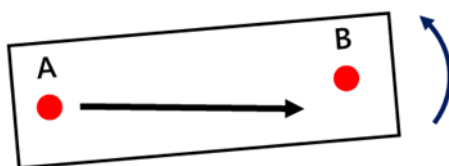


Рисунок 2. Перемещение объекта по вращающимся координатам.

На данный момент, если вы посмотрите на вращающиеся координаты как на ориентир, это будет выглядеть так, как будто к шару была приложена боковая сила и он

отклонился. [3] Эта кажущаяся сила, приложенная вбок, называется силой Кориолиса. (рис. 3)

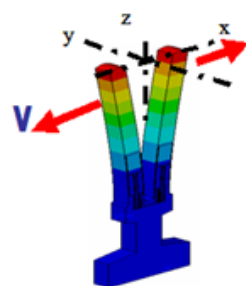


Рисунок 3. Движение объекта по вращательным координатам.

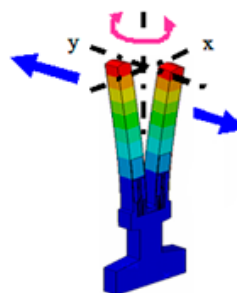
Силу Кориолиса можно рассчитать по следующей формуле:  $F = 2mv\omega$ . Другими словами, если вы можете измерить силу Кориолиса, вы можете найти угловую скорость.

Далее мы объясним, как работает гироскопический датчик вибрационного типа, наиболее распространенный тип гироскопического датчика. Вибрационный гироскоп определяет угловую скорость, определяя силу Кориолиса, возникающую при вращении вибратора во время его вибрации. Методы обнаружения включают пьезоэлектрический метод и емкостной метод.

- 1) Пьезоэлектрический метод. Сначала я объясню, как работает пьезоэлектрический метод. Во-первых, как показано на (рис 4, а), вибратор вибрирует с постоянным периодом в направлении оси X без вращения. Можно сказать, что это ситуация, описанная в предыдущем разделе, когда мяч перебрасывается друг другу в точке А и точке Б. Когда применяется вращение, как показано на (рис 4, б), в направлении оси Y генерируется сила Кориолиса, вибратор изгибается в направлении оси Y, и за счет пьезоэлектрического эффекта генерируется напряжение. Измеряя это напряжение, обнаруживается сила Кориолиса и определяется угловая скорость.



а) Генератор в невращающемся состоянии



б) Осциллятор во вращении

Рисунок 4. Пьезоэлектрический метод.

- 2) Емкостный метод. Далее мы объясним, как работает емкостной метод. Сначала, как показано на (рис 5, а), в невращающемся состоянии подайте прямоугольное напряжение на приводной электрод, чтобы заставить вибратор вибрировать с постоянным периодом в направлении оси X. Предположим, что емкости C1 и C2 между электродом обнаружения и вибратором равны. Когда применяется вращение, как показано на (рис 5, б), в направлении оси Y создается сила Кориолиса, заставляющая вибратор смещаться в направлении оси Y. В это время возникает разница между емкостями C1 и C2 между электродом обнаружения и вибратором, причем  $C1 \neq C2$ . Измеряя эту разность емкостей, обнаруживается сила Кориолиса и определяется угловая скорость.



а) Генератор в неврещающем состоянии

б) Осциллятор во вращении

Рисунок 5. Емкостный метод.

### Заклучение

В заключение можно отметить, что гироскопы стали неотъемлемой частью современных технологий. Их способности фиксировать даже мельчайшие изменения в пространстве позволяют создавать устройства с высокой точностью и адаптивностью, значительно расширяя возможности взаимодействия человека с техникой. Сегодня гироскопы применяются не только в смартфонах, где обеспечивают плавное управление и поддержку приложений дополненной реальности, но и в автомобилях, где играют ключевую роль в системах стабилизации и безопасности. Они помогают удерживать транспортное средство на дороге в сложных условиях, обеспечивая контроль за его движением. Кроме того, их применение выходит за пределы привычных сфер. Гироскопы активно используются в робототехнике, медицинских устройствах и авиации, где точность ориентации и реагирования на динамику критически важна. Благодаря постоянному совершенствованию они становятся всё более компактными и надёжными, что делает их универсальным инструментом для создания инновационных решений. Таким образом, гироскопы продолжают доказывать свою значимость, помогая формировать мир технологий, более удобный, безопасный и эффективный для человека.

\*\*\*

1. Мартынов, В. А. Сенсорные технологии в мобильных устройствах. — М.: Научное издание, 2021. — 256 с.
2. Кузнецов С. Л. Физические основы работы датчиков ориентации. Новосибирск: Наука, 2020. — 289 с.
3. Бычков С. И., Лукьянов Д. П., Бакаляр А. И. Лазерный гироскоп. Москва: Энергоатомиздат, 2019. — 254 с.
4. Волков М. Ю., Никитин А. С., Смирнов Р. И. Автомобильные системы стабилизации на базе гироскопов и акселерометров. — М.: Автоматика, 2019. — 288 с.
5. Тарасов О. Е., Григорьев С. В., Александров П. Н. Современные датчики в навигационных системах. — СПб.: Наука и Техника, 2022. — 234 с.

**Гараев А.Х., Гараев М.Х., Шахбазова М.С.**

**Принцип работы датчика ускорения**

*Чеченский государственный университет имени А.А. Кадырова*

*(Россия, Грозный)*

*doi: 10.18411/trnio-12-2024-857*

### Аннотация

В статье рассматриваются принципы работы акселерометров, их устройство и ключевые особенности функционирования. Исследуется, как акселерометры фиксируют изменение скорости и направление движения, обеспечивая точность измерения ускорения и стабильность работы различных устройств. Особое внимание уделено применению акселерометров в современных системах, таких как смартфоны, автомобили и навигационные устройства, где они играют важную роль в повышении удобства и безопасности.

**Ключевые слова:** датчики, технологии, смартфон, акселерометр, гироскоп, ускорение, оптимизация, движение, сигнал.

### Abstract

The article discusses the principles of operation of accelerometers, their structure and key features of operation. It examines how accelerometers record changes in speed and direction of movement, ensuring the accuracy of acceleration measurements and the stability of various devices. Particular attention is paid to the use of accelerometers in modern systems such as smartphones, cars and navigation devices, where they play an important role in improving convenience and safety.

**Keywords:** sensors, technologies, smartphone, accelerometer, gyroscope, acceleration, optimization, movement, signal.

Технологии развиваются стремительно, и сегодня акселерометры, измеряющие ускорение, стали частью почти каждого устройства — от смартфонов до автомобилей. Эти компактные сенсоры фиксируют изменения скорости движения и преобразуют их в электрические сигналы. Их работа основана на измерении инерционных сил, возникающих, когда объект ускоряется или замедляется.

Ускорение описывает, как быстро меняется скорость объекта. Например, увеличение скорости на  $10 \text{ м/с}^2$  означает, что каждую секунду объект ускоряется на 10 метров в секунду. Это свойство делает акселерометры незаменимыми для анализа и фиксации изменений движения.

Инерция при изменении скорости — явление, знакомое каждому. При ускорении автомобиля нас откидывает назад, а при торможении — вперёд. Это связано с действием инерционной силы, которую можно рассчитать по формуле:

$$F = m \times a,$$

где  $F$  — сила инерции,  $m$  — масса объекта,  $a$  — его ускорение.

Измеряя инерционную силу, акселерометр вычисляет ускорение объекта. В стандартных условиях ускорение в  $1G$  соответствует  $9,8 \text{ м/с}^2$ . Это позволяет датчикам фиксировать не только линейное ускорение, но и силу тяжести, а также распознавать вибрации, удары и другие типы движений.

Конструкция акселерометра может быть основана на пьезоэлектрическом или ёмкостном методе измерений. Ёмкостный метод использует разницу ёмкостей между грузом и электродами, которая меняется при ускорении. В состоянии покоя груз расположен по центру между электродами, и ёмкости  $C1$  и  $C2$  равны (см. рис. 1, а). При ускорении груз смещается под воздействием инерционной силы, что вызывает изменение ёмкостей (рис. 1, б) и фиксируется датчиком.

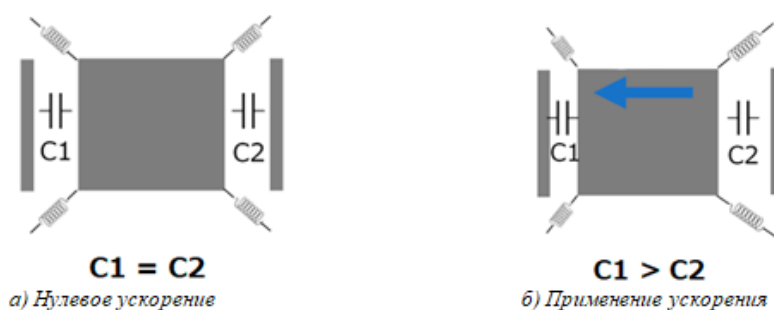


Рисунок 1.

Акселерометры способны фиксировать «движение», «гравитацию», «вибрации» и «удары», благодаря чему их можно использовать для разнообразных задач. Рассмотрим основные примеры:

1. Обнаружение движения. Одной из распространённых задач акселерометра является определение движения, например, в автомобильной навигации. В условиях отсутствия радиосигнала, таких

как туннели или удалённые местности, данные о перемещении автомобиля могут вычисляться на основе изменений ускорения. Это позволяет точно отображать пройденный маршрут, даже если стандартные GPS-сигналы временно недоступны.

2. Ориентация устройства. Функция обнаружения гравитации помогает акселерометру оценивать ориентацию устройства в пространстве. Это свойство широко используется в современных смартфонах: устройство автоматически переключает экран из книжного режима в альбомный, реагируя на наклон. В основе этого процесса лежит анализ изменений силы тяжести, воспринимаемой датчиком.
3. Регистрирование вибраций. Акселерометр также способен фиксировать возникающие колебания. Такое свойство находит применение в сейсмологии, где они используются для измерения силы землетрясений. Они улавливают изменения ускорения и фиксируют амплитуду колебаний, предоставляя данные для анализа сейсмической активности.
4. Реакция на удары. Важное применение акселерометров связано с системами безопасности. Например, в автомобилях они задействованы для управления подушками безопасности. Когда датчик фиксирует резкое изменение ускорения, подушки активируются мгновенно, предотвращая серьёзные травмы пассажиров.

Для анализа сложных движений с высокой степенью точности акселерометры часто используются совместно с гироскопами. Эти устройства, работая в тандеме, дополняют функциональные возможности друг друга: акселерометр фиксирует изменения ускорения и определяет, насколько далеко объект переместился, в то время как гироскоп отслеживает угол поворота и ориентацию в пространстве. Такое комбинированное использование особенно востребовано в автомобильной навигации, где гироскопы помогают точно определить направление движения, а акселерометры обеспечивают данные о преодолённом расстоянии. Вместе они формируют надёжную систему позиционирования, способную работать даже в условиях отсутствия GPS-сигнала, например, в туннелях или подземных паркингах.

Современные датчики различаются по количеству осей, которые они способны отслеживать, — от одной до трёх. Трёхосевые акселерометры и гироскопы обеспечивают наиболее полный и точный анализ движений, так как работают одновременно по осям X, Y и Z. Однако такая сложность требует дополнительных аппаратных решений. Для полноценной интеграции акселерометров и гироскопов на трёх осях необходимо использовать до шести отдельных компонентов. Это увеличивает расходы на производство, а также предъявляет высокие требования к пространству внутри устройства. Кроме того, любые отклонения в монтаже или настройке могут снизить точность показаний, что особенно критично в системах, где даже незначительная ошибка может привести к серьёзным последствиям.

Тем не менее, современные технологии стремительно решают эти проблемы. Одним из ключевых достижений последних лет стало появление многоосных интегрированных сенсоров, которые объединяют функции акселерометров и гироскопов в едином модуле. Это позволяет сократить количество компонентов, необходимых для работы системы, и одновременно повысить её надёжность. Интеграция датчиков в единый корпус делает их более компактными, лёгкими и энергоэффективными, что открывает новые перспективы для их использования в самых разных областях. Например, такие модули стали незаменимыми в мобильных устройствах, где компактность имеет первостепенное значение.

Более того, развитие сенсоров с интегрированными функциями не только упрощает процесс их установки и настройки, но и снижает влияние внешних факторов, таких как вибрации или температурные перепады, на точность измерений. Это особенно

важно для сложных систем, таких как автономные автомобили или промышленные роботы, где высокая точность датчиков напрямую влияет на безопасность и эффективность работы всей системы.

Использование гироскопов и акселерометров выходит далеко за пределы традиционных областей. Сегодня такие системы находят применение в медицине, например, для отслеживания движений пациентов в процессе реабилитации, в спортивной индустрии для анализа биомеханики спортсменов, а также в системах виртуальной и дополненной реальности, где точное определение положения пользователя в пространстве играет ключевую роль. Кроме того, сочетание этих датчиков активно используется в авиации, где важно учитывать не только движение, но и изменения ориентации самолёта в пространстве.

Инновации в области сенсоров открывают новые горизонты, делая возможным реализацию задач, которые ещё несколько лет назад казались недостижимыми. Например, современные многоосные модули позволяют создавать системы контроля движения, которые работают с миллисекундной точностью, что необходимо в сфере робототехники. В то же время их компактность и энергоэффективность обеспечивают их интеграцию даже в миниатюрные устройства, такие как смарт-часы или медицинские трекеры.

Таким образом, связка акселерометров и гироскопов является неотъемлемой частью многих современных технологий. Эти датчики стали своего рода стандартом точности и надёжности, обеспечивая новые уровни контроля и анализа движения. Будь то использование в бытовых гаджетах, системах безопасности или высокотехнологичных роботах, акселерометры и гироскопы продолжают доказывать свою значимость, расширяя границы возможного и формируя фундамент для новых изобретений.

\*\*\*

1. Григорьев, А. И., Яковлев, С. В. Акселерометры и гироскопы: теория и практика. — М.: Машиностроение, 2017. — 320 с.
2. Семенов, П. Н., Кузьмин, А. В. Современные микроэлектромеханические датчики ускорения. — СПб.: Политехника, 2019. — 256 с.
3. Лебедев, Ю. А., Иванов, М. Г. Основы инерциальных систем навигации и акселерометрии. — Новосибирск: Научная книга, 2020. — 280 с.
4. Васильев, Д. И. Применение акселерометров в системах управления и стабилизации. — Казань: Технополис, 2021. — 212 с.
5. Захарченко, В. П., Петров, А. Н. Сенсоры ускорения в робототехнике и мобильных устройствах. — Екатеринбург: Техноцентр, 2018. — 200 с.
6. Миронов, С. В. Микроэлектромеханические системы и их применение в промышленности. — М.: Радиотехника, 2016. — 290 с.
7. Бородин, К. Л. Технологии MEMS для датчиков ускорения. — СПб.: Электроника и связь, 2022. — 240 с.

**Гараев А.Х., Гараев М.Х., Шахбазова М.С.**

### **Сравнение SQL с процедурными языками программирования: как SQL упрощает взаимодействие с базами данных**

*Чеченский государственный университет имени А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-12-2024-858

#### **Аннотация**

SQL — это язык запросов, используемый для взаимодействия с реляционными базами данных. В отличие от традиционных процедурных языков программирования, таких как Python или Java, SQL является декларативным, то есть программист не указывает, как именно должна быть выполнена операция, а только описывает, что он хочет получить от системы. Это упрощает разработку, так как система управления базой данных (DBMS) сама решает, как оптимально выполнить запрос.

**Ключевые слова:** база данных, таблицы, язык, программирование.

**Abstract**

SQL is a query language used to interact with relational databases. Unlike traditional procedural programming languages like Python or Java, SQL is declarative, meaning that the programmer does not specify how the operation should be performed but only describes what they want to obtain from the system. This simplifies development because the database management system (DBMS) determines how to execute the query optimally.

**Keywords:** database, tables, language, programming.

**Введение**

SQL — это язык для работы с базами данных, но в отличие от языков программирования, таких как Python, C++ или Java, он не является процедурным. В традиционных процедурных языках задачи решаются через написание последовательности инструкций, которые выполняются в строго определённом порядке. Программист сам указывает, как и в каком порядке должны выполняться операции.

SQL же работает по принципу декларативного подхода: вы формулируете запрос, описываете, что хотите получить, а система управления базами данных (DBMS) сама решает, как это будет выполнено. То есть в SQL вы описываете цель, а не способ её достижения. Например, если вам нужно найти сотрудников старше 40 лет или с доходом выше 100,000 долларов, вы просто формулируете запрос, и DBMS решает, как эффективно выполнить его с учётом индексов, структуры таблицы и других факторов.

Со временем SQL был дополнен процедурными конструкциями, чтобы сделать его более гибким и удобным для сложных задач. Такие возможности, как блоки BEGIN...END, операторы IF, функции и процедуры, позволяют хранить программные логики на сервере и повторно использовать их в разных приложениях. Это особенно полезно в многоклиентских системах, где одна и та же программа или процедура может быть вызвана разными пользователями.

Таким образом, несмотря на свою декларативную природу, SQL теперь поддерживает элементы процедурного программирования, что помогает реализовывать более сложную логику непосредственно в базе данных, без необходимости обращаться к внешним языкам программирования.

**Особенности SQL**

Язык структурированных запросов сильно отличается от языков программирования. Его цель — оперирование данными в базе, а не написание скриптов или обработка файлов, отсюда и специфические свойства. Этот язык:

- стандартизирован: одни и те же SQL-запросы работают с БД с разными системами управления;
- прост и понятен: запрос информации с его помощью выглядит как команда на английском;
- декларативен: пользователь запрашивает ожидаемый результат, а не прописывает шаги к его достижению;
- имеет четкие связи: данные хранятся в строках таблиц, а столбцы соответствуют наперед заданным шаблонам, что задает структуру однотипным данным, в которых было бы легко запутаться без организации хранения.

**Преимущества:**

- Структурированность запросов помогает решать сложные задачи анализа больших объемов информации.
- Это понятный и легкий в освоении язык, что позволяет начать пользоваться им на практике уже в процессе изучения.
- Гибкость схемы БД дает возможность взаимодействовать с разными типами данных.
- SQL широко применяется, поэтому в популярных языках программирования есть специальные библиотеки и модули для взаимодействия с ним.

#### Недостатки:

- Его сложно использовать для написания циклов; с его помощью невозможно осуществить многие операции, которые позволяют реализовывать популярные языки программирования.
- Из-за фиксированной структуры реляционной БД не получится изменять структуру данных быстро и легко.
- Масштабировать реляционные БД — долго и дорого.

#### Для чего нужен SQL

Базы данных используются на большинстве сайтов, в приложениях и различных сервисах. Чтобы разработчик или пользователь приложения могли сохранить новые данные или изменить и удалить старые, необходимо промежуточное звено, которое свяжет БД и сервис. Таким звеном выступает SQL.

SQL используют все, кто работает с большим объемом информации:

- разработчики,
- тестировщики,
- аналитики,
- администраторы,
- ученые.

#### Как использовать SQL

Работа с SQL может быть организована двумя основными способами:

1. Запросы вручную. Вы можете вводить SQL-запросы напрямую в консоль или интерфейс базы данных. Это удобно, если вам нужно быстро получить ответ на один конкретный вопрос. Например, вы хотите узнать количество сотрудников в определенном отделе или найти самых высокооплачиваемых работников в компании. Это одноразовые запросы, которые не требуют дальнейшего повторного использования.
2. Интеграция SQL в приложение. В более сложных сценариях SQL-запросы могут быть встроены в программу или веб-приложение. Это даёт возможность автоматизировать извлечение данных и использовать запросы многократно, в разных контекстах. Например, запрос, который выводит отчет по продажам, может быть включён в регулярное обновление данных, выполняемое каждый месяц.

SQL, как самостоятельный язык, не обладает всеми особенностями, необходимыми для полноценного программирования. Он не поддерживает такие конструкции, как циклы или обработку исключений, которые доступны в процедурных языках. Однако в последние десятилетия SQL был значительно расширен. Внесённые в язык изменения (в версиях 1999, 2003, 2005, 2008 и 2011 годов) включают поддержку таких возможностей, как подзапросы, оконные функции, улучшенные возможности работы с типами данных и обработка ошибок.

Тем не менее, для создания более сложных приложений, например, с взаимодействием с пользователем или внешними сервисами, SQL часто используется в сочетании с другими языками программирования, такими как Python, Java или C#. SQL остаётся важнейшим инструментом для работы с данными, но для построения полноценных приложений его обычно дополняют более мощные языки программирования.

Язык структурированных запросов используется везде и выполняет различные задачи в разных сферах:

- Государственные и городские порталы используют его для хранения и изменения данных об оказываемых услугах, в реестрах и справочниках.
- В науках, таких как биоинформатика, астрономия, науки о Земле, социология, язык структурированных запросов помогает организовывать и анализировать большие объемы научной информации.
- Онлайн-магазины и банки с его помощью быстро осуществляют транзакции, обновляют цены и информацию о наличии товаров и услуг, проводят анализ интересов клиентов и формируют на его основе рекомендации.
- Медицинские организации используют реляционные БД для хранения информации о пациентах, справках, результатах анализов и т. д.

SQL используется во многих областях, и в каждой требуется специалист, обладающий навыками работы со структурированным языком запросов:

- Дата-сайентисты извлекают большие объемы данных из разнообразных информационных ресурсов, формируют сложные структуры и, анализируя их, готовят эффективные стратегии для бизнеса.
- Разработчики БД пишут блоки скриптов, позволяющие автоматизированно создавать БД и манипулировать ими. Также разработчики проводят тестирования, которые помогают отладить и оптимизировать работу БД, повысить ее производительность или дать оценку эффективности функционирования.
- Администраторы БД подготавливают требования ко всему жизненному циклу БД: разработке, использованию, поддержке. Они управляют учетными записями пользователей, задают права доступа к информации.

В каждой из упомянутых профессий умение использовать язык структурированных запросов — основной и активно используемый навык.

SQL остается основным инструментом для работы с реляционными базами данных, предлагая мощные средства для извлечения, обработки и управления данными. В отличие от процедурных языков, SQL позволяет разработчику сосредоточиться на описании задачи, а не на указаниях, как её решить, что значительно упрощает работу с базами данных. Однако, несмотря на развитие языка и добавление процедурных возможностей в последние версии SQL, для реализации более сложной логики часто требуется интеграция SQL с другими языками программирования. Тем не менее, современные версии SQL поддерживают широкий спектр функциональностей, которые делают его гибким инструментом для работы с данными в разнообразных приложениях. Таким образом, для эффективной работы с базами данных необходимо как знание SQL, так и умение интегрировать его с другими языками программирования.

\*\*\*

1. Дата, К. Дж. Введение в системы баз данных. — М.: Персей, 2017.
2. Эльмасри, Р., Навас, С.Б. Основы систем баз данных. — М.: Диалектика, 2016.
3. Шмидт, Г. С. SQL: Язык запросов к базам данных. — СПб.: БХВ-Петербург, 2001.
4. Малецкий, В. В. Реляционные базы данных: теория и практика. — М.: Высшая школа, 2019.
5. Леонова, А. В. SQL: Подробный справочник. — М.: ЛОГОС, 2014.
6. Грэй, Р., Майкл, Д. Проектирование реляционных баз данных: практическое руководство. — СПб.: Питер, 2004.

**Гарипов И.И., Андреева Е.А.**

### **Применение технологий искусственного интеллекта в различных отраслях**

*Казанский государственный энергетический университет  
(Россия, Казань)*

*doi: 10.18411/trnio-12-2024-859*

#### **Аннотация**

Данная статья исследует применение технологий искусственного интеллекта (ИИ) в различных отраслях, демонстрируя их способность революционизировать традиционные процессы и создавать новые возможности. Автор анализирует, как ИИ уже трансформирует здравоохранение, финансы, образование, производство, транспорт и другие сферы, подчеркивая как преимущества, так и вызовы, связанные с внедрением этой технологии. Статья также рассматривает этические вопросы, связанные с использованием ИИ, и предлагает взгляд на будущее этой быстро развивающейся области.

**Ключевые слова:** искусственный интеллект; технологии ИИ; отрасли; трансформация; инновации; преимущества; вызовы; этические вопросы; будущее; здравоохранение; финансы; образование.

**Abstract**

This article explores the application of artificial intelligence (AI) technologies in various industries, demonstrating their ability to revolutionize traditional processes and create new opportunities. The author analyzes how AI is already transforming healthcare, finance, education, manufacturing, transport and other areas, emphasizing both the advantages and challenges associated with the introduction of this technology. The article also examines ethical issues related to the use of AI and offers a look at the future of this rapidly developing field.

**Keywords:** artificial intelligence; AI technologies; industries; transformation; innovation, benefits; challenges; ethical issues; future; healthcare; finance; education.

Искусственный интеллект (ИИ) перестал быть фантастической концепцией и стремительно интегрируется в реальность, становясь мощным инструментом, меняющим облик различных отраслей. Его способность к анализу больших данных, автоматизации процессов и решению сложных задач открывает новые горизонты для повышения эффективности, качества и инноваций.

Искусственный интеллект активно проникает в сферу образования, и изучение иностранных языков не стало исключением. Современные технологии предлагают множество инструментов, которые могут сделать процесс обучения более эффективным, интересным и доступным[6].

Как ИИ помогает изучать языки:

- Персонализированные учебные программы: ИИ может анализировать уровень и потребности ученика, создавая адаптированные учебные планы и рекомендуемые материалы.
- Интерактивные приложения: Существуют приложения с ИИ, которые предлагают разговорные практики с виртуальными собеседниками, что позволяет улучшать навыки общения.
- Автоматический перевод: ИИ-системы, такие как GoogleTranslate, помогают переводить тексты в реальном времени, что облегчает понимание.
- Проверка произношения: Некоторые приложения могут оценивать ваше произношение и давать рекомендации по его улучшению.
- Игра для обучения: ИИ может использовать игровые механики для создания увлекательного процесса обучения, например, викторины и кроссворды.
- Анализ ошибок: ИИ может выявлять и анализировать распространенные ошибки новичков, помогая сосредоточиться на их исправлении.

Представьте, что вы решили изучать французский язык. С помощью ИИ приложения вы можете пройти тест, чтобы определить свой начальный уровень знаний. Основываясь на результатах теста, платформа создаст индивидуальный план обучения с учетом ваших сильных и слабых сторон. Вы можете практиковаться в произношении, грамматике и лексике с помощью интерактивных игр, викторин и упражнений. ИИ-помощник поможет вам освоить разговорные фразы и тренироваться в беседах. ИИ-переводчик поможет вам понять французские тексты и видеоматериалы. ИИ-платформы также предлагают онлайн-курсы по изучению французского языка с помощью видеуроков, интерактивных упражнений и обратной связи от преподавателя.

Искусственный интеллект динамично развивается и уже активно внедряется в здравоохранение, меняя подходы к диагностике, лечению и профилактике заболеваний.

Как ИИ трансформирует сферу здравоохранения[5]:

1. Диагностика и прогнозирование:

Ранняя диагностика: ИИ-алгоритмы могут анализировать медицинские изображения (рентгеновские снимки, МРТ, КТ) и выявлять патологии на ранних стадиях, повышая шансы на успешное лечение. Персонализированная диагностика: Анализ

данных пациента (генетика, история болезни, образ жизни) позволяет ИИ предсказывать риск развития заболеваний и разрабатывать индивидуальные планы лечения. Быстрая диагностика: ИИ-системы помогают врачам быстрее и точнее определять диагноз, особенно в сложных случаях.

#### 2. Лечение и терапия:

Персонализированная терапия: ИИ помогает подбирать оптимальные дозы лекарств и составлять индивидуальные протоколы лечения, учитывая особенности организма пациента. Роботизированные операции: Хирургические роботы под управлением ИИ повышают точность и эффективность операций. Разработка лекарств: ИИ ускоряет процесс поиска новых лекарств и оптимизирует их разработку.

#### 3. Профилактика и управление здоровьем:

Анализ данных: ИИ анализирует данные о здоровье населения и выявляет тенденции распространенности заболеваний, что позволяет разрабатывать профилактические программы. Мобильные приложения: ИИ-приложения помогают людям вести здоровый образ жизни, отслеживать физ. активность, питание, сон, и предупреждать о возможных проблемах со здоровьем. Телемедицина: ИИ позволяет предоставлять удаленную медицинскую помощь, что особенно важно для отдаленных районов и людей с ограниченной мобильностью.

#### 4. Управление и оптимизация:

Управление ресурсами: ИИ помогает оптимизировать распределение ресурсов в больницах, например, управлять очередями и расписанием врачей. Анализ данных: ИИ помогает анализировать большие массивы медицинских данных и выявлять тренды в здравоохранении, что позволяет улучшать качество услуг и принимать более обоснованные решения.

Несмотря на многочисленные преимущества, ИИ в здравоохранении также ставит ряд этических и практических вопросов: Как обеспечить безопасность и конфиденциальность медицинских данных пациентов, используемых ИИ-системами? Кто несет ответственность за ошибки ИИ-систем в медицине? ИИ-технологии доступными для всех пациентов, независимо от их финансового состояния? Предвзятость алгоритмов: Как избежать предвзятости в ИИ-алгоритмах, чтобы они не дискриминировали определенные группы пациентов?

Несмотря на эти вызовы, ИИ уже оказывает значительное влияние на здравоохранение и будет продолжать трансформировать эту сферу в будущем.

Искусственный интеллект также быстро завоевывает финансовую сферу, меняя способы управления деньгами, инвестирования и обеспечения безопасности. Вот как он революционизирует мир финансов:

Искусственный интеллект революционизирует инвестирование, предоставляя инвесторам мощные инструменты анализа данных, автоматизации торговли и персонализированного управления портфелем. Он способен обрабатывать огромные объемы данных, выявлять скрытые паттерны, строить прогнозы и управлять рисками. Алгоритмическая торговля с использованием ИИ позволяет совершать сделки быстрее и точнее, чем человек. Robo-advisor предоставляют автоматизированное управление инвестициями, учитывая профиль и цели инвестора, также он открывает новые возможности для создания инвестиционных продуктов, таких как фонды и страхование, управляемые искусственным интеллектом. Однако важно понимать, что ИИ не лишен недостатков, таких как отсутствие полной прозрачности и риски ошибок. Необходимо грамотно использовать его, чтобы оптимизировать инвестиционные решения, снизить риски и повысить прибыльность.

Искусственный интеллект преобразует банковские услуги, делая их более индивидуальными, удобными и эффективными. Он помогает банкам автоматизировать процессы, например, обработку транзакций, выявление мошенничества и предоставление кредитов, также он позволяет персонализировать обслуживание, предлагая клиентам

индивидуальные продукты и услуги, основанные на анализе их поведения. ИИ улучшает обслуживание клиентов, предоставляя круглосуточную поддержку, отвечая на вопросы и решая проблемы с помощью чат-ботов. Кроме того, он позволяет анализировать данные, выявлять тенденции, прогнозировать спрос и оптимизировать бизнес-процессы.

ИИ также открывает новые возможности для банков, например, развитие цифровых банковских продуктов, основанных на искусственном интеллекте, и создание более удобных мобильных приложений.

Искусственный интеллект активно внедряется в производственную сферу, меняя ее процессы и повышая эффективность. Основные области, где он оказывает влияние[7]:

1. Автоматизация и роботизация: Роботы с ИИ выполняют сложные и опасные задачи, повышая производительность и безопасность труда. Он автоматизирует рутинные операции, освобождая сотрудников для творческих задач. ИИ анализирует данные с датчиков и прогнозирует поломки оборудования, сводя к минимуму простои и затраты на ремонт.
2. Оптимизация производства: ИИ оптимизирует планирование производства, управление запасами и логистические цепочки, минимизируя затраты и повышая эффективность. ИИ-системы контролируют качество продукции, выявляют дефекты и оптимизируют производственные процессы для достижения максимального качества. Он помогает инженерам в разработке новых продуктов, оптимизируя их дизайн и функциональность.
3. Повышение безопасности: ИИ анализирует данные и прогнозирует потенциальные риски на производстве, позволяя оперативно принимать меры и предотвращать несчастные случаи. ИИ-системы распознают людей, объекты и ситуации на производстве, обеспечивая повышенную безопасность.
4. Новые технологии: ИИ оптимизирует процессы 3D-печати, создавая более сложные и точные изделия. Он анализирует данные с сенсоров и устройств IoT, оптимизируя производственные процессы и повышая эффективность.

Вызовы: Внедрение ИИ требует значительных инвестиций в оборудование, программное обеспечение и обучение персонала. Необходимо обеспечить безопасность данных и предотвратить несанкционированный доступ к ИИ-системам. Важно учитывать этические аспекты использования ИИ в производстве, например, вопросы занятости и безопасности труда.

ИИ революционизирует производство, повышая его эффективность, безопасность и инновационность. Важно грамотно внедрять ИИ, учитывая его потенциал и вызовы, чтобы получить максимальную пользу от этой технологии.

Искусственный интеллект оказывает значительное влияние на транспортную отрасль, преобразуя ее во многих аспектах. Вот некоторые из ключевых областей:

1. Автономный транспорт: ИИ-технологии уже применяются в системах автопилота, позволяя создавать полностью автономные транспортные средства (автомобили, грузовики, автобусы, поезда, дроны и даже водные суда). Это открывает перспективы для повышения безопасности дорожного движения, снижения затрат на топливо и оптимизации трафика.
2. Оптимизация трафика: ИИ анализирует данные о трафике в режиме реального времени, оптимизирует маршруты и светофоры, снижая заторы и повышая пропускную способность дорог. Он также позволяет прогнозировать спрос на транспортные услуги, что помогает планировать маршруты, расписание и количество транспортных средств, а также оптимизировать управление парковкой.

3. Улучшение безопасности: ИИ-системы, такие как системы предупреждения о выезде из полосы движения, автоматическое торможение и адаптивный круиз-контроль, повышают безопасность на дороге. Он также анализирует данные о поведении водителя, чтобы выявлять признаки усталости или невнимательности, а также позволяет моделировать аварийные ситуации для изучения причин их возникновения и разработки мер по предотвращению.
4. Новые сервисы: ИИ-приложения предоставляют информацию о маршрутах, прогнозах погоды, пробках и других данных, необходимых для планирования поездок. Он также оптимизирует транспортные услуги по требованию (каршеринг, такси), обеспечивая более эффективное и доступное движение, и интегрируется в умные города, чтобы оптимизировать транспортную систему, снизить заторы и повысить экологичность.

Вызовы: Важно учитывать этические аспекты использования ИИ в транспорте, например, вопросы ответственности за аварии. Разработка и внедрение ИИ-технологий в транспортной сфере требует четких нормативно-правовых актов. Необходимо создавать инфраструктуру, совместимую с автономным транспортом (беспроводную связь и системы распознавания). ИИ оказывает революционное влияние на транспортную отрасль, повышая ее эффективность, безопасность и комфортность. Важно грамотно внедрять ИИ, учитывая его потенциал и вызовы, чтобы получить максимальную пользу от этой технологии.

Искусственный интеллект уже сегодня меняет облик различных отраслей, повышая эффективность, качество и инновации. В будущем его влияние будет только расти, открывая новые возможности и создавая новые вызовы. Важно использовать ИИ ответственно и этично, чтобы обеспечить его безопасное и устойчивое развитие.

Кроме того, необходимо активно развивать навыки и знания, связанные с ИИ, чтобы научиться эффективно взаимодействовать с этой технологией. Образование в области ИИ станет критически важным, чтобы подготовить новое поколение специалистов, способных не только использовать ИИ, но и разрабатывать новые решения, способствующие общественному добру.

Разработка законодательных норм и стандартов для использования ИИ поможет избежать возможных рисков, связанных с его внедрением, таких как вопросы конфиденциальности, безопасности и предвзятости алгоритмов. Только совместными усилиями общества, бизнеса и государства мы сможем гарантировать, что Искусственный интеллект становится инструментом, способствующим процветанию и прогрессу.

\*\*\*

1. Искусственный интеллект: Стратегия развития в Российской Федерации" (2019). - Официальный документ, outlining государственную стратегию развития ИИ в России.
2. The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World" by Pedro Domingos (2015). - Популярная книга, описывающая основные подходы к машинному обучению и его влияние на общество.
3. Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig (2021). - Классический учебник по ИИ, охватывающий все основные концепции и технологии.
4. Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville (2016). - Comprehensive textbook on deep learning, a key technology behind many AI applications.
5. Искусственный интеллект в здравоохранении" под редакцией А. И. Иванова (2020) - Сборник статей о применении ИИ в диагностике, лечении и профилактике заболеваний.
6. Искусственный интеллект в образовании: новые возможности" под редакцией Н. В. Моисеевой (2021) - Сборник статей о применении ИИ в персонализации обучения, оценке знаний и разработке учебных программ.
7. Искусственный интеллект в промышленности: новые возможности" под редакцией А. И. Иванова (2020) - Сборник статей о применении ИИ в автоматизации производства, оптимизации процессов и контроле качества.
8. Искусственный интеллект в транспорте: новые возможности" под редакцией А. И. Иванова (2020) - Сборник статей о применении ИИ в разработке беспилотных автомобилей, оптимизации транспортных потоков и повышении безопасности на дорогах.

Гарфетдинова К.Р., Баранов С.А., Хусайнова Е.А.  
Перспективы развития рынка искусственного интеллекта в России

Казанский государственный энергетический университет  
(Россия, Казань)

doi: 10.18411/trnio-12-2024-860

#### Аннотация

Искусственный интеллект (ИИ) представляет собой одно из наиболее динамично развивающихся утверждений современной науки и технологий. В России, как и во многих других странах, развитие ИИ является ключевым фактором для достижения экономического роста, повышения качества жизни и обеспечения национальной безопасности. Настоящая статья посвящена анализу перспектив развития искусственного интеллекта в России, включая текущую стратегию, направления развития, методы и потенциальные результаты.

**Ключевые слова:** искусственный интеллект, чат-бот, алгоритм, программное обеспечение, технологии.

#### Abstract

Artificial intelligence (AI) is one of the most dynamically developing statements of modern science and technology. In Russia, as in many other countries, the development of AI is a key factor for achieving economic growth, improving the quality of life and ensuring national security. This article is devoted to the analysis of the prospects for the development of artificial intelligence in Russia, including the current strategy, directions of development, methods and potential results.

**Keywords:** artificial intelligence, chatbot, algorithm, software, technology.

Искусственный интеллект — это моделирование процессов человеческого интеллекта машинами, особенно компьютерными системами. Примерами приложений ИИ являются экспертные системы, обработка естественного языка (NLP), распознавание речи и машинное зрение.

По мере того, как шум вокруг ИИ набирала обороты, поставщики спешно продвигали то, как их продукты и услуги включают его. Часто то, что они называют «ИИ», — это хорошо зарекомендовавшая себя технология, например, машинное обучение.

ИИ требует специализированного оборудования и программного обеспечения для написания и обучения алгоритмов машинного обучения. Ни один язык программирования не используется исключительно в ИИ, но Python, R, Java, C++ и Julia являются популярными языками среди разработчиков ИИ.

В целом, системы ИИ работают, принимая большие объемы маркированных обучающих данных, анализируя эти данные на предмет корреляций и закономерностей, а затем используя эти закономерности для прогнозирования будущих состояний [3].

Например, чат-бот ИИ, которому скармливают примеры текста, может научиться генерировать реалистичные диалоги с людьми, а инструмент распознавания изображений может научиться идентифицировать и описывать объекты на изображениях, просматривая миллионы примеров. Генеративные методы ИИ, которые быстро развивались в последние несколько лет, могут создавать реалистичный текст, изображения, музыку и другие медиа.

Глобальный финансовый сектор переживает существенный сдвиг в связи с быстрым принятием технологии искусственного интеллекта. Растущее внедрение искусственного интеллекта в финансовом секторе преобразует отрасль несколькими способами. Искусственный интеллект используется для алгоритмической торговли, обнаружения мошенничества и оценки кредитного риска. Он обеспечивает более быстрый

и точный анализ данных, что приводит к принятию более обоснованных инвестиционных решений и управлению рисками. ИИ обладает способностью обрабатывать большие объемы данных, а также помогает выявлять закономерности и тенденции, которые могут упустить агенты-люди. В целом, искусственный интеллект повышает операционную эффективность, снижает затраты и улучшает качество обслуживания клиентов в финансовом секторе.

Модели и алгоритмы искусственного интеллекта, как правило, сложны. Сложная природа моделей затрудняет понимание процесса принятия решений. Решения на основе искусственного интеллекта часто генерируют результаты, которые имеют природу черного ящика, что может привести к отсутствию доверия и ответственности. Это особенно касается таких критически важных областей, как финансы и здравоохранение. Эти отрасли или конечные пользователи испытывают деликатную потребность в прозрачности для защиты прав собственности. Таким образом, проблемы прозрачности при принятии решений сдерживают рост рынка.

Ученые-исследователи все чаще используют технологию искусственного интеллекта из-за ее преобразующих возможностей. Ожидается, что эта тенденция сохранится в ближайшие годы, одновременно предоставляя возможность для расширения рынка искусственного интеллекта. Технология ускоряет анализ данных, выявляет закономерности и помогает в сложных стимуляциях, позволяя совершать прорывы в различных секторах. Инструменты на основе ИИ оптимизируют повторяющиеся задачи, освобождая ученых для сосредоточения на инновациях. Во многих областях ученые-исследователи начали ценить важность и потенциал ИИ для прогнозирования результатов, обнаружения скрытых идей и оптимизации экспериментов. Научное сообщество признает роль ИИ в повышении эффективности исследований и продвижении новых открытий, что приводит к более симбиотическим отношениям между технологиями и человеческим опытом [1].

Система с технологией искусственного интеллекта нуждается в квалифицированных специалистах, тогда как нехватка или отсутствие таких квалифицированных и опытных специалистов создает серьезную проблему для рынка. Технологии искусственного интеллекта требуют специализированных знаний в таких областях, как машинное обучение, наука о данных и разработка программного обеспечения. Спрос на этих специалистов превышает предложение, что приводит к нехватке талантов, которая может замедлить разработку, внедрение и инновации ИИ. Более того, эта нехватка может ограничить масштабируемость и эффективность решений ИИ в различных отраслях. Таким образом, нехватка квалифицированных специалистов рассматривается как проблема для расширения рынка.

В 2019 году Президент Российской Федерации подписывает проект развития искусственного интеллекта до 2030 года. Данная стратегия определяет основные задачи и направления развития ИИ в России, среди которых:

Обеспечение опережающего развития ИИ через поддержку фундаментальных и прикладных научных исследований.

Создание и совершенствование программного обеспечения, использующего технологию ИИ.

Обеспечение доступности и качества данных, необходимые для развития технологий ИИ.

Обеспечение наличия аппаратного обеспечения, необходимого для решения задач в области ИИ.

Создание комплексной системы подготовки специалистов в области ИИ.

В рамках реализации Национальной стратегии был утвержден федеральный проект «Искусственный интеллект», действующий с 2021 по 2024 год. Этот проект включает 5 ключевых показателей и 17 результатов, обеспеченных бюджетным и внебюджетным финансированием. Основной целью проекта является обеспечение того, чтобы

предприятия и граждане использовали продукты, основанные на исключительно отечественных технологиях ИИ, обеспечение нового качественного уровня эффективности деятельности.

Современные технологии ИИ в России реализуются по всем основным направлениям, включая компьютерное зрение, обработку естественного языка, распознавание и синтез речи, а также интеллектуальные системы, поддерживающие принятие решений. Компьютерное зрение позволяет машинам интерпретировать и визуализировать информацию, которая широко используется в таких областях, как транспорт, здравоохранение и производство. Обработка естественного языка подразумевает и включает в себя человеческий язык, который имеет широкое применение в чат-ботах, голосовых помощниках и медицинском переводе [2].

ИИ уже успешно применяется в различных отраслях российской экономики, включая энергетику, логистику и промышленность. В этих устройствах ИИ позволяет сделать машины умнее и автономнее, уменьшая количество поломок и повышая эффективность производства. Например, в энергетике ИИ используется для оптимизации процессов производства и распределения энергии, а в логистике — для улучшения маршрутизации и управления цепочками поставок.

Развитие ИИ в России активно осуществляется как государственными, так и частными структурами. Между Правительством Российской Федерации и крупными компаниями, такими как ПАО Сбербанк, «Яндекс», ВК, МТС и «Газпром нефть», заключаются соглашения о международных отношениях и поддержке развития ИИ. Эти партнерства направлены на привлечение инвестиций, создание национальных облачных платформ и поддержку российских стартапов в области ИИ.

Одним из ключевых факторов на пути развития ИИ в России является нехватка квалифицированных специалистов. Для решения этой проблемы Минэкономразвития России и Минтруда будут ежегодно анализировать риски, связанные с внедрением ИИ, и разрабатывать программу для вузов по подготовке специалистов в этой сфере. Создание центров компетенций, которые будут объединять наработки в области ИИ и лоббировать интересы отечественных разработчиков, а также является направлением.

Развитие ИИ имеет важное социальное и социальное значение для России. По прогнозам экспертов, к 2030 году применение технологий ИИ может прибавить к ВВП России более 11 трлн рублей. Кроме того, ведущий, что будет задержано не менее 1 тыс. руб. ИТ-стартапов, создано порядка 2 тыс. решения и продукты, и подготовлено более 850 тыс. специалисты в области ИИ.

Информационная безопасность является одним из важных вопросов развития в России. Правительство планирует разработать надежные решения для хранения данных, включая облачные платформы, которые будут обеспечивать надежную деятельность органов власти, операторов связи и предприятий. Регулирование общественных отношений, возникающих в связи с развитием и использованием технологий ИИ, также является важным аспектом. Российские эксперты такого рода участвуют в работе международных организаций, таких как Совет Европы, ЮНЕСКО и ОЭСР, для разработки наднациональных документов, регулирующих сферу ИИ.

Несмотря на значительные перспективы, развитие ИИ в России сталкивается с рядом вызовов. Одним из основных факторов является практическое отсутствие подготовки профессиональных кадров в этой области. В связи с этим возникает необходимость ликвидировать разрыв между результатами научных исследований и практическими потребностями отраслей. Кроме того, России необходимо создавать системно работающие механизмы результатов ИИ-разработок на мировых рынках, что требует совместной работы бизнеса и развития государственных институтов [4].

Развитие искусственного интеллекта в России является стратегическим направлением, обладающим потенциалом значительного экономического и экономического воздействия. Текущие стратегии, Национальный этап развития ИИ и

федеральный проект «Искусственный интеллект», направлены на поддержку научных исследований, разработку программного обеспечения, повышение доступности данных и аппаратного обеспечения, а также подготовку квалифицированных кадров.

Частно-государственное партнерство и инвестиции играют ключевую роль в этом процессе. Однако для полноценного развития ИИ необходимо решить такие проблемы, как подготовка специалистов, ликвидация разрыва между современными исследованиями и практическими потребностями, а также обеспечение информационной безопасности [5].

В будущем развитие ИИ в России может привести к значительному росту экономики, повышению качества жизни и ухудшению национальной безопасности. Успешная реализация нынешних тенденций и преобразований в настоящее время позволит России занять лидирующие позиции в глобальной гонке за доминирование в области искусственного интеллекта.

\*\*\*

1. Аксенова Е.И. Экспертный обзор развития технологий искусственного интеллекта в России и мире. Выбор приоритетных направлений развития искусственного интеллекта в России. - Москва: ГБУ «НИИОЗММ ДЗМ», 2019. - 38 с.
2. Гольцев А. Совокупность автоматизированных систем и средств управления производством. ИТ в промышленности 2021. [Электронный ресурс]. URL: [https://www.snews.ru/reviews/it\\_v\\_promyshlennosti\\_2021](https://www.snews.ru/reviews/it_v_promyshlennosti_2021) (дата обращения: 25.11.2024).
3. Городнова Н.В. Применение искусственного интеллекта в бизнес-сфере: современное состояние и перспективы // Вопросы инновационной экономики. - 2021. - № 4. - с. 1473-1492.
4. Дорожная карта развития сквозной цифровой технологии «Нейротехнологии и искусственный интеллект». [Электронный ресурс]. URL: <https://digital.gov.ru/uploaded/files/07102019ii.pdf> (дата обращения: 25.11.2024).
5. Зайнетдинов Э. 30 бизнес-процессов, которые изменятся из-за искусственного интеллекта. [Электронный ресурс]. URL: <https://hype.ru/deecrypto-store-club/30-biznes-processov-kotorye-izmenyatsya-iz-za-iskusstvennogo-intellekta-dkva585> (дата обращения: 25.11.2024).

**Голубятников А.О.**

**Альтернативные методы геолокации на основе данных GSM и Wi-Fi**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-861

#### **Аннотация**

В статье рассматриваются альтернативные подходы к геолокации для сервисов LBS, использующие данные сотовых вышек и Wi-Fi сетей. Обсуждаются ограничения GPS и преимущества комбинированного метода для определения местоположения в условиях слабого спутникового сигнала. Предложен метод обработки данных для повышения точности геолокации с помощью кластеризации сигналов и применения статистических моделей.

**Ключевые слова:** геолокация, GSM, Wi-Fi, кластеризация сигналов, LBS, GPS ограничения.

#### **Abstract**

The article discusses alternative approaches to geolocation for LBS services using data from cell towers and Wi-Fi networks. The limitations of GPS and the advantages of the combined method for determining location in conditions of a weak satellite signal are discussed. A data processing method is proposed to improve the accuracy of geolocation by clustering signals and using statistical models.

**Keywords:** geolocation, GSM, Wi-Fi, signal clustering, LBS, GPS restrictions.

Спутниковые навигационные системы (GNSS - Global Navigation Satellite System), такие как GPS и ГЛОНАСС (Глобальная навигационная спутниковая система), сегодня являются наиболее точным методом определения местоположения. Почти все современные смартфоны оснащены такими модулями. Однако, не всегда и не везде они могут решить задачи, связанные с сервисами геолокации (LBS - Location Based Services).

Во-первых, поиск спутников может занять несколько минут, а в некоторых ситуациях важнее скорость определения, даже если точность будет ниже. Например, когда нужно быстро построить маршрут или отметить своё местоположение.

Во-вторых, спутниковые сигналы не работают в помещениях или под землёй.

В-третьих, GPS-модуль есть не в каждом устройстве, особенно в ноутбуках. Поэтому для LBS требуются альтернативные решения.

Такие альтернативы существуют — местоположение можно определить по ближайшим GSM-вышкам, Wi-Fi сетям и даже по IP-адресу. Хотя каждый из этих методов менее точен, чем GPS, их комбинация может дать приемлемый результат. Недостатки одного метода компенсируются преимуществами другого. Например, GSM-вышки есть почти везде, а Wi-Fi — не всегда. Но Wi-Fi обеспечивает лучшую точность. Комбинированный метод, использующий оба подхода, даёт более точные результаты, чем каждый из них по отдельности. Если у двух роутеров в разных частях города одинаковые MAC-адреса, совместное использование данных с GSM-вышек поможет избежать ошибок, так как идентификаторы сотовых вышек, скорее всего, будут разными.

На момент разработки Локатора по улицам уже перемещались сотни тысяч людей с включенными Яндекс.Картами на своих устройствах. Приложение, с согласия пользователя, передавало GPS-координаты, которые использовались для построения Яндекс.Пробок. Было решено, что вместе с этим приложение также может фиксировать, какая базовая станция обслуживает телефон на данных координатах, а также какие сети Wi-Fi находятся в зоне видимости (при этом подключение к самим сетям не происходит, что исключает риски для конфиденциальности).

Для участия в этом краудсорсинговом процессе пользователю не требовалось совершать дополнительных действий — достаточно было пользоваться приложением. Данные о координатах, Wi-Fi сетях и базовых станциях обезличивались, занимали незначительный объём и не влияли на заряд аккумулятора.

Таким образом, пользователи приложения оказывали взаимную помощь, что изображено на рисунке 1:

- Одни, с активными GPS-модулями, собирали точные данные о местоположении сетей и передавали их в Яндекс.
- Другие, не имеющие GPS-модулей, отправляли список видимых сетей и получали свое приблизительное местоположение на карте.

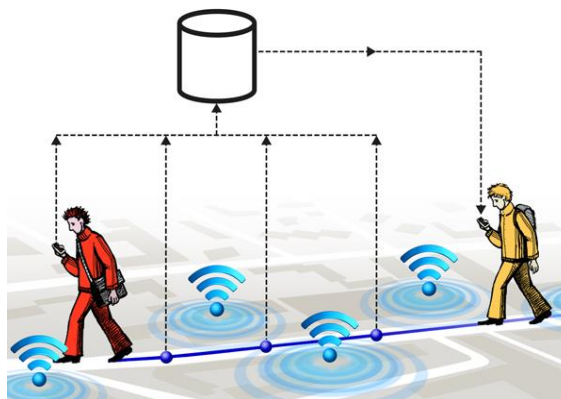


Рисунок 1. Сбор базы данных сетей и GPS координат для устройств через взаимопомощь в “Яндекс Картах”.

Практика показывает, что идентификаторы сотовых вышек постоянно меняются: вышка, которая вчера обслуживала центр города, завтра может оказаться на его окраине. Аналогичная ситуация происходит и с Wi-Fi-роутерами, которые могут перемещаться вместе с владельцами. Это приводит к необходимости регулярно обновлять и аннулировать часть собранных данных.

Для решения этих проблем был разработан следующий подход. Когда от пользователя поступает запрос на определение местоположения, вместе с ним передаются данные о видимых Wi-Fi сетях и сотовых вышках. Если одна из сетей была замечена в разных частях города, алгоритм оценивает количество сигналов, накопленных в каждом районе, а также их возраст. Каждое скопление сигналов от одной и той же Wi-Fi сети или вышки называется «облаком», что отображено на рисунке 2. Чем больше сигналов накоплено в облаке и чем они свежее, тем большее доверие оно вызывает. В ответ на запрос выбирается облако с наибольшим количеством свежих сигналов.

Если облако не обновлялось более месяца, оно считается устаревшим, даже если для этой сети нет более свежего облака в другом районе. Таким образом, удаётся поддерживать актуальность данных и корректировать местоположение в случае перемещения вышек и роутеров.

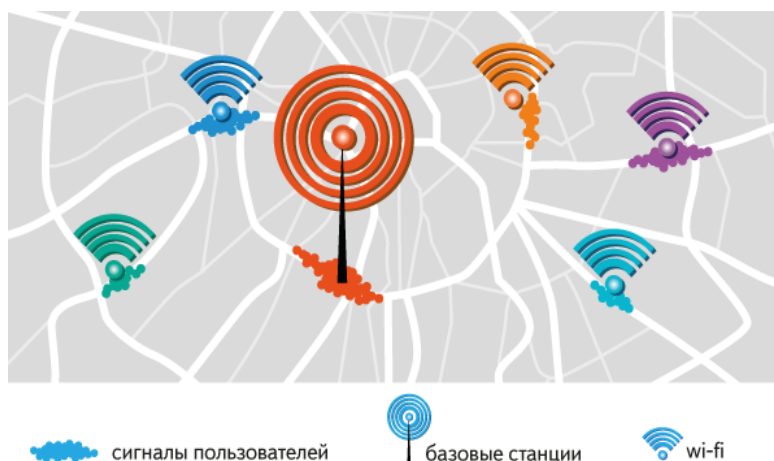


Рисунок 2. Облако сигналов от пользователей “Яндекса”.

При определении местоположения с помощью радиосигналов его точность всегда будет приблизительной, поэтому отображение в виде одной точки нецелесообразно — правильнее использовать круг. Радиосигналы в идеальных условиях распространяются равномерно во все стороны, но фактические данные чаще образуют эллипс. Это объясняется тем, что большинство пользователей мобильных Карт — автомобилисты, и их GPS-следы сосредоточены вдоль дорог, в то время как сигналы из дворов и зданий поступают реже.

Чтобы повысить точность ответа, радиус круга должен быть минимальным. Однако если просто нарисовать окружность, охватывающую все точки сигнала, радиус окажется слишком большим. Уменьшить радиус помогли методы математической статистики. Плотность сигналов подчиняется нормальному распределению, что позволяет применить правило трех сигм (если случайная величина распределена нормально, то абсолютная величина ее отклонения от математического ожидания не превосходит утроенного среднего квадратического отклонения). В этом радиусе оказывается 99,7% всех сигналов, что отображено на рисунке 3.

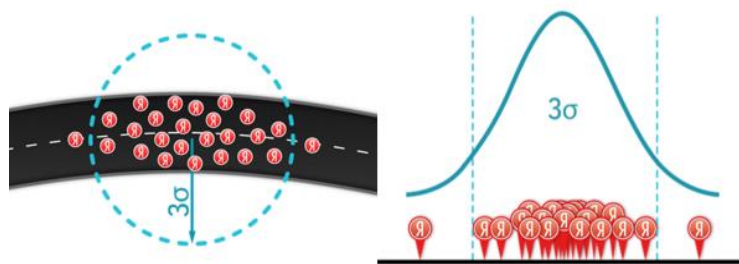


Рисунок 3. Круг геолокации по правилу трех сигм.

Был проведён эксперимент, который позволил подобрать коэффициент для сигмы, уменьшающий радиус круга, сохраняя при этом приемлемую точность. Это стало возможным благодаря тому, что в большинстве случаев пользователь видит несколько сетей одновременно. Области, "открытые" за счет уменьшения радиуса одной сети, часто перекрываются сигналами других сетей, что сохраняет точность определения местоположения.

Не все GPS-сигналы от пользователей удаётся легко сгруппировать в облака. При наложении сигналов одной сети на карту, помимо характерных "эллипсов", могут появиться отдельные точки и линии. Это, как правило, одиночные сигналы, сильно удаленные от основного скопления, а также длинные GPS-треки (цепочки сигналов GPS).

Одиночные сигналы часто возникают, когда человек, например, едет в метро. Телефон может потерять связь с вышкой на одной станции, а при выходе на другой станции все еще считать, что он подключен к той же вышке. Локатор отфильтровывает такие сигналы. Также был установлен минимальный порог для облаков, чтобы не учитывать слишком малочисленные скопления сигналов, что отображено на рисунке 4.

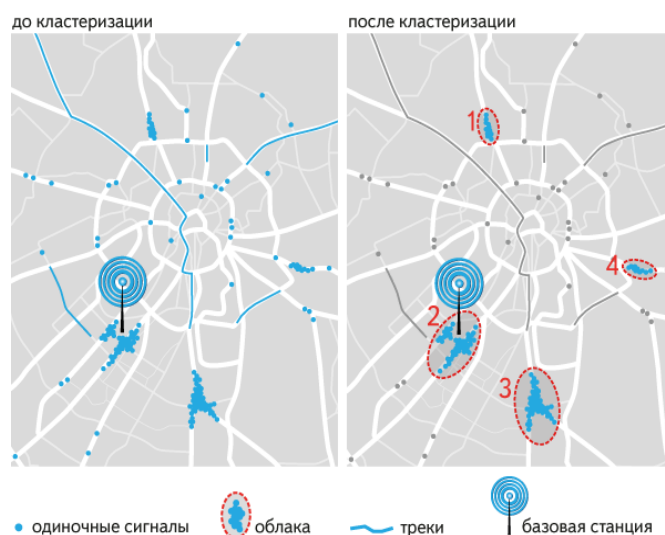


Рисунок 4. Фильтрация одиночных сигналов и треков с помощью кластеризации.

Длинные GPS-треки появляются, например, при движении на автомобиле через весь город. Телефон может "тянуть" идентификатор вышки на протяжении всего пути, создавая иллюзию, что она видна на всём маршруте. Поскольку радиус действия базовых станций ограничен, такие треки также фильтруются Локатором. В районах с малым количеством данных такие треки могут формировать цепочку небольших облаков, если их длина соответствует реальному радиусу действия вышки.

Одиночные сигналы, маленькие облака и длинные треки считаются "шумом". Если у пользователя видна только одна сеть, и для неё есть лишь такие шумовые сигналы,

система выдаст сообщение о невозможности определить местоположение, чтобы не предоставить неточные данные.

На ранних этапах, когда данных было ещё немного, возникали сложности с объединением всех сигналов в одно облако. Иногда сигналы от вышки в одном городе могли быть зафиксированы и в другом. Эту проблему помогло решить использование кода зоны местоположения — LAC (Location Area Code), присутствующего в идентификаторах GSM-сетей. Вышки с одинаковым кодом должны находиться рядом, поэтому облакам с другим LAC, находящимся в "чужом городе", Локатор стал придавать меньший вес, что позволило более точно определять местоположение, что отображено на рисунке 5.

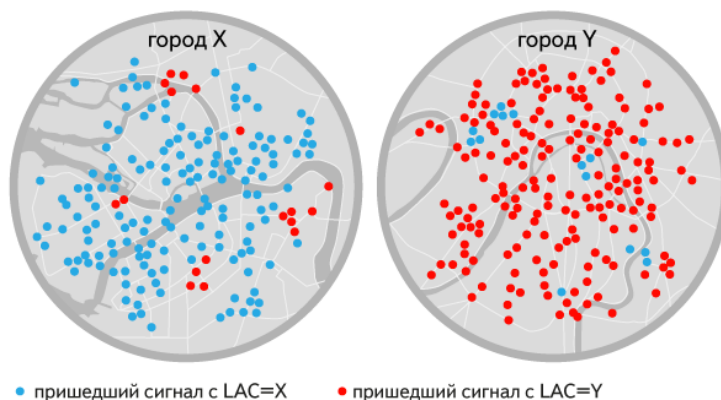


Рисунок 5. Фильтрация сигналов с использованием Location Area Code.

\*\*\*

1. Петрова Т. В. и др. "Подходы обнаружения беспроводной точки доступа злоумышленника в локальной вычислительной сети" // Региональная информатика. – 2022. – С. 572-573.
2. Волкогонов В. Н. и др. "Применение физически неклонируемых функций для выполнения аутентификации в среде интернета вещей" // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2021. – С. 409-414.
3. Богомаз М. Э., Михайлова Л. А., Поляничева А. В. "Инструменты обеспечения безопасности IP-телефонии" // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2022. – С. 170-172.
4. Леснова Е. М., Пестов И. Е. "Разработка метода обнаружения и коррекции ошибок для распределенной информационной сети на основе больших данных" // Региональная информатика и информационная безопасность. – 2018. – С. 236-240.

**Голубятников А.О.**

**Перспективы и будущее развития LPWAN**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-862

#### **Аннотация**

Статья изучает развитие LPWAN и его интеграцию с сетями 5G, Wi-Fi и mesh для создания гибридных IoT-решений. Рассмотрены перспективы расширения применения LPWAN в энергоэффективных и высокоскоростных системах, а также возможности и вызовы таких комбинаций в промышленных и городских экосистемах.

**Ключевые слова:** LPWAN, 5G, IoT, гибридные сети, энергоэффективность, интеграция сетей.

**Abstract**

The article examines the development of LPWAN and its integration with 5G, Wi-Fi and mesh networks to create hybrid IoT solutions. The prospects of expanding the use of LPWAN in energy-efficient and high-speed systems, as well as the opportunities and challenges of such combinations in industrial and urban ecosystems are considered.

**Keywords:** LPWAN, 5G, IoT, hybrid networks, energy efficiency, network integration.

С развитием и внедрением сетей 5G возникает вопрос о том, какое влияние эта технология окажет на существующие сети LPWAN. Хотя 5G и LPWAN изначально создавались для различных задач, их сочетание может привести к значительным изменениям в области Интернета вещей (IoT). 5G ориентирован на высокоскоростную передачу данных, минимальные задержки и надежное соединение, что идеально подходит для приложений, требующих большой пропускной способности и высокой плотности подключенных устройств, таких как автономные автомобили и умные города. В то же время LPWAN фокусируется на низком энергопотреблении и возможности передачи небольших объемов данных на дальние расстояния, что делает его идеальным решением для приложений, где устройства работают на батарейках и требуют длительного времени автономной работы, например, для сельскохозяйственных датчиков или мониторинга состояния окружающей среды.

Сети 5G могут дополнить LPWAN, предоставляя широкополосную инфраструктуру и улучшенные возможности подключения для определенных IoT-приложений. Например, 5G может обеспечить более устойчивое и быстрое подключение для устройств, использующих LPWAN для передачи данных. Благодаря этой интеграции можно будет объединить преимущества обеих технологий, создав гибридные сети, которые одновременно поддерживают высокоскоростную передачу данных и низкоэнергетические коммуникации для сенсоров и других IoT-устройств. Кроме того, будущие стандарты, такие как 6G, могут еще больше усилить эти возможности, добавив поддержку квантовых вычислений, улучшенных алгоритмов шифрования и интеграции искусственного интеллекта, что сделает сети более интеллектуальными и адаптивными.

Однако существует и конкуренция между LPWAN и 5G в плане захвата IoT-рынка. Сети 5G, благодаря низкой латентности и высокой плотности подключений, могут постепенно заменить некоторые приложения LPWAN, особенно в областях, где требуется быстрая передача данных и поддержка множества устройств. Тем не менее, пока что LPWAN остается более экономически выгодным и энергоэффективным решением для маломощных устройств, работающих на больших расстояниях.

В перспективе, развитие 5G и последующих стандартов, вероятно, приведет к дальнейшему расширению возможностей LPWAN через интеграцию с другими технологиями, что позволит создавать более сложные и разнообразные IoT-экосистемы. В результате, LPWAN и 5G могут не только сосуществовать, но и дополнять друг друга, предоставляя более широкие возможности для инновационных IoT-приложений.

Интеграция LPWAN (Low Power Wide Area Network) с другими типами сетей открывает новые возможности для создания гибридных коммуникационных систем, способных решать различные задачи и охватывать широкий спектр приложений. Благодаря своей низкой мощности, широкому радиусу действия и низкой пропускной способности, LPWAN отлично подходит для долгосрочного мониторинга и сбора данных в IoT-приложениях, где не требуется частая передача данных и высокая скорость. Однако интеграция LPWAN с другими сетевыми технологиями, такими как Wi-Fi, 5G и mesh-сети, позволяет значительно расширить возможности использования этой технологии.

Сети 5G, например, обладают высокой пропускной способностью и низкой задержкой, что делает их идеальными для приложений с интенсивной передачей данных, таких как потоковое видео и онлайн-управление сложными устройствами. Комбинируя LPWAN и 5G, можно создать гибридные решения, где LPWAN используется для

мониторинга и периодической передачи данных, а 5G обеспечивает высокоскоростное подключение для более критичных задач. Такое взаимодействие может быть полезным для промышленных IoT-систем, где одни устройства нуждаются в высокоскоростной связи, а другие — в экономии энергии и автономной работе.

Wi-Fi, в свою очередь, часто применяется для высокоскоростной передачи данных на короткие расстояния, что делает его хорошим дополнением для LPWAN в зданиях или на предприятиях. Использование Wi-Fi вместе с LPWAN позволяет организовать «умные» пространства, где устройства могут переключаться между LPWAN и Wi-Fi в зависимости от задачи. Например, датчики температуры или освещенности могут использовать LPWAN для передачи данных на большие расстояния, а более сложные устройства, такие как камеры или контрольные панели, подключаться через Wi-Fi для передачи больших объемов данных локально.

Mesh-сети, такие как Zigbee и Z-Wave, также могут дополнять LPWAN, особенно в ситуациях, где требуется плотное покрытие в пределах ограниченной области. Mesh-сети создают распределенные сети, где каждый узел может передавать данные соседям, что повышает надежность связи. В комбинации с LPWAN, mesh-сети позволяют организовать системы с высокой избыточностью, где данные собираются и передаются по mesh-сети, а затем отправляются на дальние расстояния через LPWAN. Это полезно для аграрного сектора, складов и логистических центров, где необходимо охватить большие территории с распределенными сенсорами.

Интеграция LPWAN с другими типами сетей открывает возможности для гибких и комплексных IoT-решений. Гибридные системы, объединяющие LPWAN, 5G, Wi-Fi и mesh-сети, могут удовлетворять разные требования к подключению, пропускной способности и энергопотреблению, предоставляя бизнесу более адаптивные и устойчивые системы.

\*\*\*

1. Красов А. В., Сахаров Д. В., Тасюк А. А. "Проектирование системы обнаружения вторжений для информационной сети с использованием больших данных" // Научные технологии в космических исследованиях Земли. – 2020. – Т. 12. – №. 1. – С. 70-76.
2. Волкогонов В. Н. и др. "Применение физически неклонированных функций для выполнения аутентификации в среде интернета вещей" // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2021. – С. 409-414.
3. Абрамова Е. А., Красов А. В., Поляничева А. В. "Тенденции развития и безопасность IP-телефонии" // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2023. – С. 23-28.
4. Леснова Е. М., Пестов И. Е. "Разработка метода обнаружения и коррекции ошибок для распределенной информационной сети на основе больших данных" // Региональная информатика и информационная безопасность. – 2018. – С. 236-240.

**Голубятников А.О.**

**Технологии Wi-Fi RTT и Fingerprinting для высокоточного позиционирования в помещениях**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

*doi: 10.18411/trnio-12-2024-863*

**Аннотация**

В статье исследуются возможности технологий Wi-Fi RTT и Fingerprinting для создания высокоточных систем позиционирования внутри помещений. Описаны принципы работы, преимущества и ограничения каждого подхода, а также применение алгоритмов машинного обучения для повышения точности определения местоположения. Рассмотрены примеры использования в логистике, общественных пространствах и на предприятиях.

**Ключевые слова:** Wi-Fi RTT, Fingerprinting, позиционирование, машинное обучение, RSSI, внутренняя навигация.

### Abstract

The article explores the possibilities of Wi-Fi RTT and Fingerprinting technologies for creating high-precision indoor positioning systems. The principles of operation, advantages and limitations of each approach are described, as well as the use of machine learning algorithms to improve the accuracy of location determination. Examples of use in logistics, public spaces and enterprises are considered.

**Keywords:** Wi-Fi RTT, Fingerprinting, Positioning, Machine Learning, RSSI, Internal navigation.

Технология Wi-Fi RTT была представлена в 2018 году вместе с операционной системой Android 9 Pie. Она позволяет устройствам точно измерять расстояние до ближайших маршрутизаторов с точностью до 1–2 метров. Wi-Fi RTT определяется протоколом IEEE 802.11mc и открывает новые возможности для внутренней навигации, позволяя разрабатывать сервисы и приложения, которые помогают пользователям ориентироваться в помещениях.

С помощью Wi-Fi RTT мобильные устройства с Android 9 могут определять расстояние до маршрутизаторов, не подключаясь к ним. Технология защищает конфиденциальность пользователей, привязывая ее к существующей системе определения местоположения и обеспечивая рандомизацию MAC-адреса, что предотвращает несанкционированное отслеживание. Принцип работы основан на измерении времени задержки передачи сигналов: система вычисляет этот временной промежуток и умножает его на скорость света, что отображено на рисунке 1.

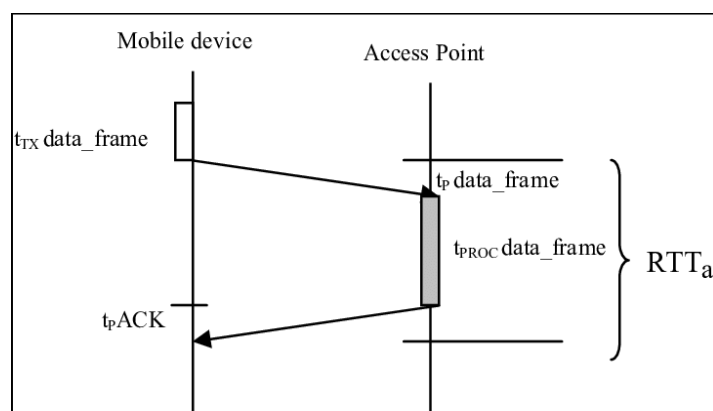


Рисунок 1. Измерение задержки между “раундами” передачи сигналов для вычисления местоположения.

Wi-Fi RTT находит широкое применение, особенно в таких областях, как отслеживание транспорта и грузов на предприятиях и складах, где она обеспечивает высокую точность и ускоряет производственные процессы. В общественных точках доступа, таких как торговые центры, аэропорты, офисы и выставки, технология также эффективно используется для систем позиционирования, позволяя пользователям прокладывать маршруты к нужным местам и оптимизировать время.

Для развертывания системы необходимо соблюдать ряд требований. Устройство должно работать на Android 9 (API уровень 28) или более поздней версии, а аппаратное обеспечение смартфона — соответствовать стандарту 802.11-2016 FTM. Приложению требуется разрешение ACCESS\_FINE\_LOCATION, а также включенные функции сканирования Wi-Fi и определения местоположения. Маршрутизатор должен поддерживать протокол IEEE 802.11-2016 FTM. Кроме того, приложение должно быть открытым на устройстве, так как оно не может получать данные о местоположении в фоновом режиме.

Компания Navignе провела тестирование технологии Wi-Fi RTT и выяснила, что при соблюдении всех требований точность позиционирования в контрольной точке

составляет менее 1 метра в 95% случаев, а в 50% случаев точность остается в пределах 30 см. Задержка определения местоположения не превышает 1 секунды. Эти результаты достигнуты благодаря алгоритму комплексирования, использующему алгоритмы PDR и фильтрацию частиц.

Традиционный метод отпечатков (Fingerprinting), основанный на RSSI (Received Signal Strength Indicator - уровень силы сигнала в дБ), заключается в записи силы сигнала от нескольких доступных точек доступа и сохранении этой информации в базе данных вместе с известными координатами клиентского устройства на этапе оффлайн-обработки. Эти данные могут быть как детерминированными, так и вероятностными. В процессе онлайн-трекинга текущий RSSI-вектор в неизвестном местоположении сравнивается с данными, сохраненными в базе отпечатков, и ближайшее совпадение возвращается в качестве предполагаемого местоположения пользователя. Такие системы могут обеспечивать медианную точность до 0,6 метра и точность в крайних случаях до 1,3 метра.

Основным недостатком этого метода является то, что любые изменения в окружающей среде, такие как добавление или удаление мебели или зданий, могут изменить "отпечаток" для каждого местоположения, что требует обновления базы данных отпечатков. Тем не менее, интеграция с другими датчиками, такими как камеры, может помочь справиться с изменениями в окружающей среде.

Fingerprinting работает по следующему принципу:

Сначала в здании устанавливаются беспроводные точки доступа (AP), которые помечаются как AP1, AP2 и так далее. С помощью умного терминала происходит сканирование сигналов точек доступа в контрольной точке в среде, что позволяет получить сигналы, излучаемые различными AP, и соответствующие MAC-адреса, связанные с этими AP. Интенсивность получаемых сигналов, представляется в виде RSS.

При распространении сигнал Wi-Fi испытывает потерю мощности. В частности, плотность мощности сигнала уменьшается с увеличением расстояния, следуя модели потерь сигнала. Интенсивность Wi-Fi сигнала, излучаемого одной и той же точкой доступа, становится уникальной на различных расстояниях. Каждое местоположение представлено уникальным вектором RSS ( $RSS_{i1}, RSS_{i2}, \dots, RSS_{in}$ ) на основе сигналов, излучаемых несколькими AP. Эта взаимосвязь между вектором RSS и местоположением аналогична концепции человеческого отпечатка и может использоваться в качестве уникального идентификатора информации о местоположении.

Далее, эти векторы можно использовать для предсказания местоположения используя один из следующих алгоритмов, что отображено на рисунке 2:

- K-nearest neighbor (KNN)
- Алгоритм кластеризации
- Support vector machine (SVM)
- Random forest
- Использование алгоритмов машинного обучения

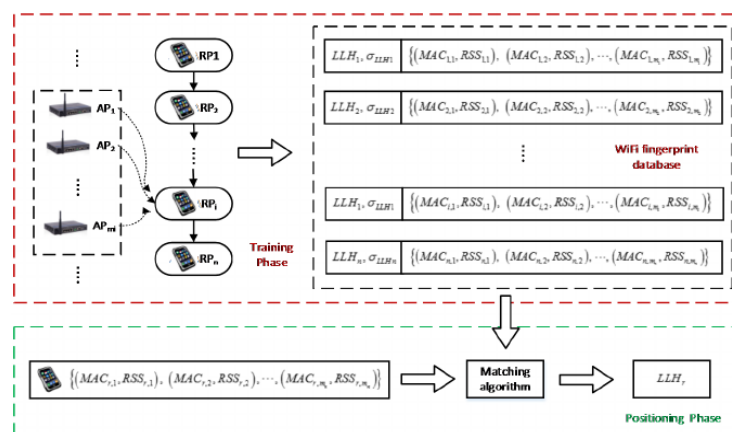


Рисунок 2. Принцип работы WiFi Fingerprinting для более точного определения местоположения через алгоритмы предсказаний.

Современные технологии, такие как Wi-Fi RTT и Fingerprinting, предоставляют новые возможности для высокоточного позиционирования внутри помещений. Благодаря использованию этих технологий предприятия, общественные пространства и объекты логистики могут значительно улучшить процессы управления и повысить удобство для пользователей. Wi-Fi RTT обеспечивает точность до 1–2 метров и защищает конфиденциальность пользователей за счет рандомизации MAC-адресов, а метод Fingerprinting, основанный на анализе RSSI, позволяет создать уникальные «отпечатки» местоположений, которые могут быть использованы для эффективного трекинга. Несмотря на необходимость регулярного обновления базы данных для поддержания точности, обе технологии представляют значительный интерес для дальнейшего развития систем внутренней навигации и аналитики.

\*\*\*

1. Богомаз М. Э., Михайлова Л. А., Поляничева А. В. "Инструменты обеспечения безопасности IP-телефонии" // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2022. – С. 170-172.
2. Петрова Т. В. и др. "Подходы обнаружения беспроводной точки доступа злоумышленника в локальной вычислительной сети" // Региональная информатика. – 2022. – С. 572-573.
3. Волкогонов В. Н. и др. "Применение физически неклонированных функций для выполнения аутентификации в среде интернета вещей" // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2021. – С. 409-414.
4. Абрамова Е. А., Красов А. В., Поляничева А. В. "Тенденции развития и безопасность IP-телефонии" // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2023. – С. 23-28.

**Гочияева М.Д., Урумов Ш.А.**

**Роль MRP-систем в оптимизации производственного планирования  
компьютерной техники**

*ФГБОУ ВО «Северо-Кавказская государственная академия»  
(Россия, Черкесск)*

*doi: 10.18411/trnio-12-2024-864*

**Аннотация**

Статья посвящена исследованию роли систем планирования ресурсов производства (MRP-систем) в оптимизации производственного процесса на предприятиях, занимающихся производством компьютерной техники. В работе рассматриваются основные принципы функционирования MRP-систем, их влияние на планирование материальных ресурсов, управление запасами и производственные процессы. Особое внимание уделено оптимизации цепочек поставок и сокращению времени производства при использовании MRP-систем.

**Ключевые слова:** MRP-система, производственное планирование, компьютерная техника, оптимизация процессов, производственные технологии, управление производством, эффективность производства.

**Abstract**

The article is devoted to the study of the role of production resource planning systems (MRP systems) in optimizing the production process at enterprises engaged in the production of computer equipment. The paper discusses the basic principles of the functioning of MRP systems, their impact on the planning of material resources, inventory management and production processes. Special attention is paid to optimizing supply chains and reducing production time when using MRP systems.

**Keywords:** MRP system, production planning, computer technology, process optimization, production technologies, production management, production efficiency.

На сегодняшний день для эффективной работы любой компании, повышения его производительности и конкурентоспособности, руководство должно уделить особое внимание структуре и системе управления бизнес-процессами. В этой связи продолжает оставаться актуальной задача рационального планирования материальных потребностей и ресурсов компании, что напрямую влияет на улучшение качества выпускаемой продукции. Решением данной проблемы является грамотная оценка специфики предприятия и, на основе этой оценки, выбор, внедрение и адаптация системы MRP, которая гарантирует стабильную работу производственной программы.

MRP-система это инструмент, который интегрирует различные процессы, от управления складами до работы с персоналом, улучшая их эффективность. MRP-система это не просто технология, а целая концепция, направленная на планирование материальных ресурсов и точное определение их потребности заранее. В условиях современного производства без такого инструмента сложно обеспечить нужное количество комплектующих и сырья, что имеет критическое значение для бесперебойной работы. Нерегулярные поставки материалов могут привести к серьезным финансовым потерям, включая простои, упущенную прибыль и штрафы за несвоевременную доставку. Пример логической структуры MRP систем представлен на рисунке 1.



Рисунок 1. Логическая структура MRP систем.

Системы MRP относятся к основным внутренним микро логистическим системам, основанным на концепции планирования потребностей и ресурсов (RP). Это логистические системы управления, которые помогают разрабатывать и адаптировать планы с целью оптимизации планирования потребности в материалах, организации поставок и контроля их обеспечения в производственном процессе через метод управления запасами и учета специфики производства в целом. Суть этих систем заключается в своевременном получении информации о наличии материалов на складах и других местах хранения, организации планирования производства, а также расчете планов закупок необходимых компонентов с учетом различных факторов.

К ключевым аспектам, которые непосредственно влияют на внедрение и функционирование системы планирования материальных потребностей, относятся следующие: финансовое положение и размеры предприятия; планируемые объемы производства; прогнозируемый спрос на готовую продукцию; сроки реализации готовых изделий; особенности технологических процессов; наличие или отсутствие необходимых компонентов для производства каждого изделия.

Несмотря на широкое распространение MRP-систем, многие компании сталкиваются с проблемой недостаточной интеграции этих систем с другими

функциональными модулями корпорации. Это приводит к потерям времени на синхронизацию данных между различными системами, снижению точности прогнозирования и увеличению затрат на управление производственными ресурсами. Интеграция MRP-системы с ERP, CRM и MES предоставляет возможность создать единую информационную среду, что существенно повышает эффективность управления всеми процессами на предприятии.

Рассмотрим роль MRP-системы в оптимизации процессов планирования производства компьютеров. Производство компьютеров представляет собой сложный и многоэтапный процесс, в котором используется огромное количество различных компонентов и материалов. Без системы планирования потребностей в материалах на каждом этапе производства могут возникать следующие проблемы:

Нехватка компонентов — из-за недостаточного контроля за запасами и неучтенных факторов спроса на материалы, часто возникают задержки в поставках, что приводит к простоям на производственной линии.

Накопление излишков материалов — отсутствие автоматизированного контроля за запасами может привести к избыточным закупкам материалов, что создает дополнительные затраты на хранение и управление запасами.

Невозможность точного планирования производственных мощностей — без системы, которая точно прогнозирует потребности в материалах, трудно организовать эффективное производство и своевременно запустить новые партии продукции.

Проблемы с управлением поставками — невозможность синхронизации поставок с производственным процессом может привести к задержкам в сборке компьютеров и нарушению сроков поставки готовой продукции.

MRP-система позволяет решать эти проблемы путем интеграции всех процессов планирования в одну единую платформу. В случае с производством компьютеров MRP-система решает следующие задачи:

Оптимизация запасов. Внедрение MRP-системы позволяет значительно уменьшить уровень запасов на складах, так как она точно рассчитывает потребности в материалах на каждом этапе производства. Например, если для сборки компьютера требуется 10 процессоров, система автоматически рассчитывает, когда нужно заказать их поставку, чтобы избежать как дефицита, так и избыточных запасов.

Автоматизация закупок и поставок. С помощью MRP-системы процессы закупки и поставки компонентов становятся более предсказуемыми и управляемыми. Система может учитывать все параметры, такие как срок поставки, минимальные и максимальные объемы закупки, а также спецификации комплектующих, что минимизирует риски задержек и ошибок в поставках.

Планирование производственных мощностей. Внедрение MRP-системы позволяет оптимизировать использование производственных мощностей. Например, система автоматически рассчитывает, когда и какие компоненты необходимы для сборки партии компьютеров, что позволяет избежать простоя из-за нехватки материалов.

Учет сроков производства. Система помогает отслеживать сроки производства и поставок, обеспечивая более точное соблюдение сроков доставки готовой продукции. Это важно для поддержания хороших отношений с клиентами и соблюдения контрактных обязательств.

Интеграция с другими системами. Современные MRP-системы могут интегрироваться с другими корпоративными информационными системами, такими как ERP (Enterprise Resource Planning) и CRM (Customer Relationship Management). Это позволяет централизованно управлять всеми аспектами бизнеса — от производства до маркетинга и продаж.

Преимущества внедрения MRP-системы в производство компьютеров:

Снижение затрат. Благодаря точному планированию и оптимизации запасов, компания может снизить затраты на хранение и управление запасами. Меньше затраты на закупки компонентов, так как закупаются только те материалы, которые действительно необходимы для производства.

Увеличение производственной эффективности. Оптимизация процесса закупки и использования материалов снижает риски простоя и позволяет более эффективно использовать производственные мощности.

Улучшение качества продукции. Автоматизированный контроль за качеством материалов и компонентов позволяет предотвращать использование дефектных или неподходящих деталей в производственном процессе, что способствует улучшению качества готовой продукции.

Гибкость и способность быстро адаптироваться к изменениям. В условиях быстроменяющегося рынка компьютерной техники MRP-система позволяет оперативно реагировать на изменения спроса или на изменения в спецификациях продукции.

Ускорение процессов принятия решений. Система предоставляет руководителям и менеджерам актуальную информацию в режиме реального времени, что помогает оперативно принимать решения и минимизировать риски, связанные с нехваткой или избытком материалов.

Внедрение MRP-системы в производство компьютеров — это сложный, но необходимый процесс. Вот несколько ключевых шагов, которые помогут эффективно внедрить такую систему:

Анализ текущих процессов. Прежде чем внедрить MRP-систему, необходимо провести анализ текущих процессов планирования и управления запасами на предприятии. Это позволит выявить слабые места и определить, какие именно функции необходимо автоматизировать.

Выбор подходящей MRP-системы. Существует множество MRP-систем, от простых до более сложных решений, интегрированных с другими корпоративными системами. Важно выбрать ту систему, которая наиболее полно удовлетворяет потребности компании и может легко масштабироваться в будущем.

Обучение персонала. Внедрение новой системы требует обучения сотрудников, которые будут с ней работать. Это необходимо для эффективного использования всех возможностей системы и минимизации ошибок в процессе эксплуатации.

Интеграция с другими системами. Важно обеспечить интеграцию MRP-системы с другими информационными системами предприятия, такими как ERP и CRM, чтобы процессы планирования и управления материалами были тесно связаны с другими аспектами бизнеса.

Постоянный мониторинг и оптимизация. После внедрения системы необходимо регулярно отслеживать ее работу, выявлять возможные проблемы и оптимизировать процессы для достижения максимальной эффективности.

MRP-система является мощным инструментом для оптимизации процессов планирования производства в сфере компьютерной техники. Она позволяет существенно улучшить управление материальными ресурсами, сократить издержки, повысить производственную эффективность и обеспечить своевременное выполнение заказов. В условиях высокой конкуренции и быстро меняющихся требований рынка внедрение MRP-системы становится необходимостью для производителей, стремящихся повысить свою конкурентоспособность и качество продукции.

\*\*\*

1. Калентьева, Ю. Н. Проблема реализации и внедрения MRP-систем на современном промышленном предприятии / Ю. Н. Калентьева. — Текст : непосредственный // Молодой ученый. — 2017. — № 17 (151). — С. 353-356. — URL: <https://moluch.ru/archive/151/42834/>.
2. Кизим А.А. ERP-системы в деятельности российских промышленных предприятий// Экономика устойчивого развития. 2013, №15.
3. Малахов М. Н. Эволюция подходов к производственному планированию в АСУ промышленных предприятий (от MRP к APS) / М.Н. Малахов //Известия Орловского государственного технического университета. Серия: Информационные системы и технологии. – 2007. – №. 4.
4. Цифровая экономика: 2020: краткий статистический сборник / Г. И. Абдрахманова, К. О. Вишневский, Л. М. Гохберг и др.; Нац. исслед. ун-т «Высшая школа экономики». – М.: НИУ ВШЭ, 2020. – 96 с.
5. Структура MRP системы. <http://www.kgau.ru/istiki/isu>

**Гочияева М.Д., Чмилев К.Е.**  
**Алгоритмы автоматизации юридического контроля при заказах печатной  
продукции: от контрактов до доставки**

*ФГБОУ ВО «Северо-Кавказская государственная академия»  
(Россия, Черкесск)*

*doi: 10.18411/trnio-12-2024-865*

**Аннотация**

В данной статье авторами рассматриваются алгоритмы для автоматизации юридического контроля в рамках жизненного цикла заказа, от заключения контракта до доставки продукции. Основное внимание уделено разработке алгоритмов для проверки условий договоров, контроля за соблюдением сроков, качества продукции и урегулирования юридических вопросов на всех этапах.

**Ключевые слова:** информационные системы, юридическое сопровождение, заказ печатной продукции, автоматизация, алгоритмы, контракт, контроль качества, логистика, жизненный цикл заказа.

**Abstract**

In this article, the authors consider algorithms for automating legal control within the life cycle of an order, from the conclusion of a contract to the delivery of products. The main attention is paid to the development of algorithms for verifying the terms of contracts, monitoring compliance with deadlines, product quality and resolving legal issues at all stages.

**Keywords:** information systems, legal support, printing order, automation, algorithms, contract, quality control, logistics, order lifecycle.

Типография занимается предпечатной и после печатной обработкой продукции, а также самой печатью тиража. Данные процессы сложны и имеют множество составляющих, что вызывает проблемы у управляющего персонала в связи с оформлением заказов и осуществлением контроля за выполнением заказов. Заказы печатной продукции, как правило, включают многочисленные этапы: от согласования условий контракта и контроля за выполнением условий до доставки и проверки качества продукции. Каждый из этих этапов сопровождается с юридическими рисками, такими как нарушение сроков, несоответствие продукции заявленным характеристикам, проблемы с оплатой и урегулирование споров.

Традиционное юридическое сопровождение процессов заказов печатной продукции зачастую требует значительных затрат времени и ресурсов, включая работу юристов, бухгалтеров и логистов. Введение автоматизированных систем, способных обрабатывать юридические данные в реальном времени, позволяет значительно снизить издержки, повысить точность выполнения условий контрактов и сократить время реагирования на возможные проблемы.

Цель данной статьи — предложить алгоритмы для автоматизации юридического контроля в рамках всего жизненного цикла заказа печатной продукции, включая этапы заключения контракта, контроля за качеством продукции и урегулирования юридических вопросов при доставке.

Жизненный цикл заказа печатной продукции можно представить как последовательность взаимосвязанных этапов, включающих (рисунок 1):

Заключение контракта. На этом этапе важно проверить соответствие условий контракта законодательным требованиям, а также подтвердить наличие всех необходимых юридических документов (договоры, лицензии, сертификаты).

Контроль качества и соблюдения сроков. Проблемы, связанные с несоответствием качества продукции или задержкой выполнения заказа, часто приводят к юридическим спорам. На этом этапе система должна контролировать выполнение обязательств поставщика.

Поставка и доставка. Завершающий этап, на котором важным является контроль за условиями доставки, проверка соответствия спецификаций и подтверждение получения продукции заказчиком.

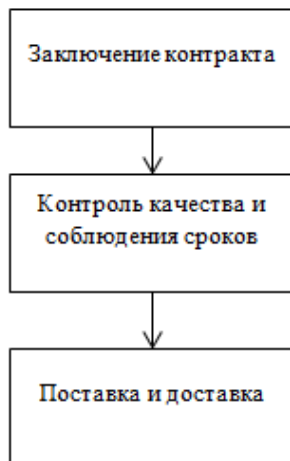


Рисунок 1. Алгоритм жизненного цикла заказа.

Для каждого из этих этапов можно выделить соответствующие алгоритмы контроля, которые должны быть интегрированы в единую информационную систему.

Одним из ключевых аспектов автоматизации юридического контроля является алгоритм проверки контракта. Он должен включать следующие элементы:

Автоматическая проверка реквизитов: алгоритм должен проверять на наличие обязательных реквизитов в контракте (наименование сторон, предмет договора, сроки, стоимость и т.д.).

Сравнение с законодательными требованиями: система должна быть оснащена базой актуальных законодательных актов, что позволит автоматически проверять, соответствуют ли условия контракта нормативам.

Верификация подписей и полномочий: система должна проверять подлинность подписей сторон и полномочия их представителей, используя цифровые подписи или другие методы идентификации.

Контроль качества и соблюдения сроков выполнения заказа важнейший элемент юридического контроля, поскольку часто нарушение этих условий ведет к юридическим последствиям. Алгоритм контроля может включать (рисунок 3):

Мониторинг выполнения этапов заказа: на основе данных о прогрессе выполнения заказа (производственные отчеты, графики, сроки поставки) система должна автоматически уведомлять о возможных отклонениях от графика.

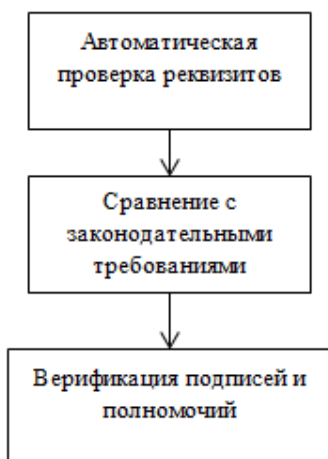


Рисунок 2. Алгоритм проверки контракта.

Контроль качества: использование стандартов и проверочных алгоритмов для сопоставления фактических характеристик продукции с указанными в контракте.

Уведомления и санкции: при нарушении сроков или качества система должна автоматически уведомлять ответственных лиц и предлагать соответствующие санкции или пути урегулирования конфликта.



Рисунок 3. Алгоритм контроля за качеством и сроками.

На этапе доставки заказчик может столкнуться с различными юридическими проблемами, такими как утрата или повреждение продукции, несоответствие количества или качества. Алгоритм урегулирования таких ситуаций может включать:

Формирование актов приема-передачи: автоматизация составления и подписания актов приема-передачи продукции.

Анализ претензий: система должна автоматически анализировать жалобы и претензии, формировать отчеты о возможных нарушениях условий контракта.

Решение спорных ситуаций: алгоритм может предлагать варианты разрешения спора (например, переговоры, возврат товара, компенсация и т.д.) на основе ранее установленных условий контракта.

Для проверки эффективности предложенных алгоритмов был проведен эксперимент по автоматизации юридического контроля в типографии, занимающейся производством печатной продукции на заказ. Исходно все процессы контроля выполнялись вручную, что требовало значительных трудозатрат и времени. После внедрения информационной системы, основанной на предложенных алгоритмах автоматическая проверка договоров с использованием законодательных баз и стандартных шаблонов снизила время на обработку каждого контракта на 30%. Интеграция системы контроля качества с производственными процессами позволила сократить количество возвратов продукции по причине несоответствия стандартам на 15%. Автоматизация сбора данных о претензиях и формирование актов ускорило процесс урегулирования споров в два раза, повысив удовлетворенность клиентов. Таким образом, внедрение информационных систем для автоматизации юридического контроля показало значительное улучшение как в точности выполнения условий контрактов, так и в сокращении времени на решение юридических вопросов.

Разработка и внедрение моделей и алгоритмов для автоматизации юридического контроля при заказах печатной продукции значительно повышают эффективность бизнес-процессов. Это позволяет не только снизить затраты на юридическое сопровождение, но и уменьшить юридические риски, улучшить контроль за выполнением условий контрактов, а также ускорить процессы урегулирования споров. Интеграция юридических, производственных и логистических процессов через информационные системы

представляет собой перспективное направление для повышения конкурентоспособности предприятий в области печатной продукции.

В будущем возможно развитие таких систем с использованием технологий искусственного интеллекта для более сложных задач, таких как анализ рисков, предсказание возникновения юридических проблем и автоматическое предложение оптимальных решений.

\*\*\*

1. Автоматизация типографий [Электронный ресурс] – Режим доступа: [http://print-expert.net/pubs/avtomatizaciya\\_tipografii.html](http://print-expert.net/pubs/avtomatizaciya_tipografii.html)
2. Верёвкин Д. М. Применение CASE-технологий для анализа бизнеспроцессов при проектировании информационных систем. Павлодар : Наука, техника и образование, 2022. 51 с.
3. Константинов П. К., Павлов И. С. Программы лояльности как неотъемлемая часть современного маркетинга. М.: Научно-образовательный журнал для студентов и преподавателей «StudNet», 2021. 25 с.
4. Мартыщенко Д. О. Современные технологии в маркетинге. РостовНа-Дону : Молодой исследователь Дона, 2020. 113 с.
5. Системы управления полиграфическим предприятием [Электронный ресурс] – Режим доступа: [http://www.nrap.ru/pub20\\_50\\_1\\_1390.html](http://www.nrap.ru/pub20_50_1_1390.html)

**Гочияева М.Д., Чотчаев Р.Б.**

**Чат-боты как инструмент для повышения вовлеченности школьников  
в учебный процесс**

*ФГБОУ ВО «Северо-Кавказская государственная академия»  
(Россия, Черкесск)*

*doi: 10.18411/trnio-12-2024-866*

**Аннотация**

В статье авторы рассматривают возможности использования чат-ботов для улучшения образовательной среды, повышения мотивации учеников, а также их роли в организации учебного процесса. Приводятся примеры успешного применения чат-ботов в школах, а также технологии, которые могут быть использованы для разработки подобных решений. Авторы приходят к выводу, что внедрение чат-ботов в образовательную практику имеет значительный потенциал для улучшения учебных результатов и взаимодействия всех участников образовательного процесса.

**Ключевые слова:** чат-боты, учебный процесс, геймификация, автоматизация учебного процесса, интерактивное обучение, мотивация.

**Abstract**

In the article, the authors consider the possibilities of using chatbots to improve the educational environment, increase the motivation of students, as well as their role in the organization of the educational process. Examples of successful use of chatbots in schools are given, as well as technologies that can be used to develop such solutions. The authors conclude that the introduction of chatbots into educational practice has significant potential to improve learning outcomes and the interaction of all participants in the educational process.

**Keywords:** chatbots, learning process, gamification, automation of the learning process, interactive learning, motivation.

Внедрение чат-ботов в образовательный процесс становится не только возможностью для улучшения качества образования, но и необходимостью в условиях цифровизации и стремительных изменений в сфере образования. Чат-боты могут стать, неотъемлемой частью современного учебного процесса, обеспечивая более гибкое, доступное и персонализированное обучение. В современном обществе наблюдается

увеличение образовательных платформ в интернете. Это создаёт потребность в удобных и многофункциональных информационных сервисах, одним из которых являются чат-боты. Многие российские ученые, такие как Зильберман Н.Н., Иванов А.Д., Кузнецов В.В., в своих трудах рассматривают различные аспекты использования и перспективность чат-ботов в области информационных технологий [5].

Чат-боты могут стать мощным инструментом для повышения вовлеченности школьников в учебный процесс. В условиях дистанционного обучения, а также на фоне увеличения нагрузки на педагогов, такие инструменты помогают снизить стресс, обеспечить доступность образовательных материалов и создать более персонализированную атмосферу для учеников. Более того, чат-боты могут служить связующим звеном между учащимися, учителями и родителями, обеспечивая эффективную и постоянную коммуникацию.

Чат-боты могут адаптировать контент под потребности каждого ученика. Это особенно важно, так как школьники могут иметь разные уровни подготовки. Бот может предлагать индивидуальные задания, повторение пройденного материала или дополнительные ресурсы для тех, кто сталкивается с трудностями. Благодаря такой индивидуализации, ученики получают именно ту информацию и помощь, которая им необходима, что способствует лучшему усвоению материала [3].

Например, если ученик задает вопрос по теме, которая была пройдена неделю назад, чат-бот может не только предоставить ответ, но и предложить дополнительные объяснения, ссылки на видеоуроки или интерактивные задания для закрепления знаний.

Введение элементов геймификации в учебный процесс помогает сделать его более увлекательным и мотивирующим. Чат-боты могут проводить викторины, конкурсы, ставить задачи с баллами и наградами. Это создает элемент соревнования, что стимулирует учащихся к более активному участию в учебе. Геймификация позволяет школьникам почувствовать процесс обучения как нечто более интересное и занимательное, что значительно повышает их вовлеченность.

Бот может предложить учащимся выполнить несколько задач на определенную тему, за каждую правильную задачу начислять баллы или виртуальные награды. В конце недели ученик может получить грамоту за успешное прохождение задания, что мотивирует продолжать обучение.

Чат-боты могут значительно снизить нагрузку на учителей и администраторов учебных заведений, автоматизируя рутинные процессы. Они могут напоминать учащимся о сроках сдачи домашних заданий, отслеживать успеваемость, а также сообщать о школьных мероприятиях и изменениях в расписании.

Например, чат-бот может отправлять ученикам автоматические уведомления о сроках сдачи контрольных работ, а также напоминать о родительских собраниях или школьных праздниках, что позволяет ученикам и родителям быть в курсе всех важных событий.

Использование чат-ботов в качестве инструмента интерактивного обучения дает возможность учащимся вовлечься в процесс на более глубоком уровне. Бот может задавать вопросы, поощрять рассуждения и помогать ученикам открывать новые подходы к решению задач. Это способствует развитию критического мышления и творческих способностей.

Вопросы с несколькими вариантами ответа, которые предоставляет чат-бот, могут стимулировать учащихся на обсуждение, что помогает развивать их логическое и аналитическое мышление. Например, бот может предложить ученику решить задачу по математике, а затем предложить несколько способов ее решения с комментариями и пояснениями.

Для создания чат-ботов используется множество инструментов и технологий. Для школьных нужд вполне подходят и простые решения, такие как создание чат-ботов на основе WhatsApp, Telegram или Viber, что позволяет интегрировать чат-ботов в повседневную практику общения школьников и учителей.

Приведем несколько примеров успешного применения чат-ботов в школах:

1. Чат-боты для вовлечения родителей в учебный процесс. В некоторых школах чат-боты используются для того, чтобы поддерживать связь с родителями. Например, они могут информировать о достижениях и успеваемости ребенка, отправлять уведомления о предстоящих мероприятиях или экзаменах. Такой подход помогает повысить уровень вовлеченности родителей в учебный процесс.
2. Чат-боты для поддержки учебного процесса. В школах, которые внедрили чат-ботов, они активно используются для решения вопросов, связанных с расписанием, домашними заданиями и текущими заданиями. Чат-боты позволяют учащимся получать ответы на часто задаваемые вопросы и находить полезную информацию без необходимости обращаться к учителям или администрации.
3. Чат-боты для обучения и тренировки навыков. В некоторых учебных заведениях чат-боты используются для тренировки навыков, таких как иностранные языки или математические вычисления. Такие боты предлагают ученикам разнообразные задания, адаптированные под их уровень, и дают обратную связь по выполнению.

Ниже приведен пример простого чат-бота на языке Python для образовательной среды, который может отвечать на вопросы школьников, напоминать о заданиях, а также предоставлять полезную информацию, такую как расписание, сроки сдачи домашних заданий и другие общие вопросы.

Пример фрагмента кода взаимодействия с ботом:

Hello! I am your SchoolBot. Ask me anything related to school.

Type 'exit' to stop.

You: When is the next exam?

SchoolBot: The next exam is scheduled for next Friday.

You: What is the homework for today?

SchoolBot: Today's homework is to read chapter 5 of the science book and solve problems 1-10.

You: How can I submit my homework?

SchoolBot: You can submit your homework online through the school portal.

You: exit

Goodbye! Have a great day!

В данном примере используется SQLite для хранения данных, что позволяет сохранить историю общения с ботом, в примере бот отвечает только на ограниченное количество заранее заданных вопросов. Бот можно дополнительно персонализировать, чтобы он задавал вопросы школьникам о предпочтениях в учебе, отслеживал их успехи и предлагал задания по конкретным темам.

Чат-бот может быть полезен для улучшения взаимодействия между учениками и образовательным процессом, автоматизируя ответы на часто задаваемые вопросы и помогая с организационными задачами.

Чат-боты это эффективный инструмент для повышения вовлеченности школьников в учебный процесс. Они позволяют создавать персонализированные и интерактивные образовательные среды, способствующие развитию самостоятельности, критического мышления и мотивации учащихся. Использование таких технологий помогает не только улучшить качество обучения, но и снизить нагрузку на педагогов, а также повысить уровень коммуникации между учащимися, учителями и родителями.

\*\*\*

1. Перспективы чат-ботов в 2020 году [Электронный ресурс]. Режим доступа: <https://yandex.m/turbo/hub.forklog.com/s/perspektivy-chat-botov-v-2020-godu/>.
2. Зильберман Н.Н. Технологии виртуальных собеседников и формы речевого взаимодействия [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/tehnologii-virtualnyh-sobesednikov-i-formy-rechevogo-vzaimodeystviya>.
3. Золотарюк А.В. Технологии создания и применения чат-ботов [Электронный ресурс]. Режим доступа: <https://cyberleninka.ru/article/n/tehnologii-sozdaniya-i-primeneniya-chat-botov/viewer>.
4. Чат и мессенджер-боты: тенденции в 2020 году [Электронный ресурс]. Режим доступа: <https://marketer.ua/chat-and-messenger-bots-trends-2020/>.
5. Чивилёв А.А. Межличностная коммуникация с виртуальными собеседниками в пространстве современной культуры [Электронный ресурс]. Режим доступа: <https://moluch.ru/conf/artcult/archive/195/10338/>.

**Григорьева Е.Г., Солонина А.П.**

**Технологии визуализации в архитектурном моделировании  
(на примере Царицынского парка г. Волгограда)**

*Волгоградский государственный университет  
(Россия, Волгоград)*

doi: 10.18411/trnio-12-2024-867

**Аннотация**

В данной работе рассматриваются возможности пакета трехмерного моделирования Blender при решении задачи трехмерной архитектурной визуализации. В работе приведены некоторые технологии моделирования отдельных объектов сцены, методы настройки освещения и материалов. В итоге продемонстрированы финальные виды трехмерной сцены на примере Царицынского парка города Волгограда.

**Ключевые слова:** Трехмерное моделирование, архитектурная реконструкция, Blender, технологии визуализации, рендеринг.

**Abstract**

This paper examines the capabilities of the Blender 3D modeling package when solving the problem of 3D architectural visualization. The paper presents some technologies for modeling individual scene objects, methods for setting up lighting and materials. As a result, the final views of the 3D scene are demonstrated using the example of Tsaritsyn Park in Volgograd.

**Keywords:** 3D modeling, architectural reconstruction, Blender, visualization technologies, rendering.

**Введение**

Данная статья посвящена применению современных технологий трехмерного моделирования для визуализации историко-культурных достопримечательностей. За основу проекта был взят комплекс Царицынского парка, располагающегося в городе Волгограде в пойме реки Царицы. Парк является популярной зоной прогулок среди жителей города Волгограда и его туристов. Уникальность и современность данного места делает его интересным выбором для создания 3D визуализации как интерактивной

туристической рекламы города. Подобная визуализация может быть использована для разработки виртуальных туров по городу, в том числе, в обучающих целях.

#### Анализ структуры парка

Парк содержит в себе обширную территорию, на которой находятся детские и спортивные площадки с множеством спортивных тренажеров, велосипедные асфальтированные дорожки вдоль всей длины парка, ряды двухместных качелей, выполненных в современном дизайне, большая площадь с озеленением и главное достояние парка – интерактивный музей «Россия – Моя история». Современные электронные карты позволяют получать локации объектов в нужном масштабе, поэтому одним из референсов парка послужил скриншот парка с «Яндекс Карт» (рис. 1).

Для моделирования детских и спортивных комплексов использовались их фото и проекты на сайтах официальных производителей (рис.2). Детский комплекс на территории парка выполнен в виде Кремлевской стены. Он включает в себя восемь башен, открытую спиральную горку, закрытую спираль-трубу, прямую трубчатую горку, два тоннеля, шесть мостов-переходов с витыми ограждениями, две металлические решетки для лазания. Кроме того, стандартная комплектация включает две сетки для лазания, два рукохода, четыре шведские стенки, два "пожарных" шеста, лиану с дисками, четыре стенки для скалолазания, спираль с шестом, изогнутый металлический лаз, платформу-вход на первом этаже, лестницу, а также декоративные элементы, повторяющие архитектуру Кремлевской стены.



Рисунок 1. Снимок «Яндекс-карты».



Рисунок 2. Фото детского комплекса.

Центральным объектом парка является интерактивный музей «Россия – моя история», который выполнен в современном минималистичном стиле, и гармонично вписывается в окружающий его ландшафт. Внешне музей повторяет простую прямоугольную форму с выдавленными силуэтами деревьев по периметру фасада, а также зеркальными и стеклянными элементами (рис. 3).



Рисунок 3. Фото музея "Россия - моя история" (вид сбоку и вид сверху).

На территории парка так же расположен ряд качелей в шахматном порядке с нестандартным дизайном (рис. 4). Они выполнены из деревянных балок с металлическим основанием.



Рисунок 4. Качели парка.



Рисунок 5. Фото фонаря.

По всей территории парка расположено множество мелких элементов архитектуры таких, как фонарные столбы (рис. 5), лавки, а так же имеется большая площадь с озеленением.

#### **Выбор программных средств для моделирования**

Для разработки трехмерной модели парка выбрана программа для 3D моделирования Blender. В сравнении с другими средами трехмерного моделирования, такими как Cinema 4D, 3ds Max, Autodesk Maya, Blender ряд преимуществ [1, 2]:

- Бесплатный и открытый исходный код;
- Удобный и гибкий интерфейс, который можно настроить по собственным предпочтениям;
- Blender имеет обширный выбор инструментов для разработки высоко детализированных текстурированных объектов, есть инструменты скульптинга, анимации.
- Постоянное расширение функционала с каждым выпуском обновления пакета;
- Программа доступна для часто используемых ОС таких, как Windows, Linux и macOS.

Методы, задействованные в реализации поставленной задачи

1. Импорт референсов. В Blender есть возможность добавить визуальный референс напрямую в проект. Это особенно удобно, если целью проекта является реализация точной копии предмета. [2]
2. Моделирование ландшафта. Благодаря инструментам скульптинга можно точно передать форму ландшафта и всего, что имеет нестандартную ассиметричную форму. [3]
3. Создание растений. В проекте были применены некоторые аддоны для оптимизации процесса и снижения загруженности проекта.
4. Настройка материалов. Программа позволяет гибко настраивать цветовую оболочку объектов и воспроизвести огромное множество материалов из реального мира. [3]
5. Освещение. Пакет Blender предлагает различные вариации освещения. В данном проекте используются точечные источники освещения Point Lamp для имитации света фонарей, глобальное солнечное освещение Sun с регулировкой угла падения теней, силы света и HDRI-карту. HDRI-карта – это изображение, с которого создается копия освещения. С ее помощью можно создать на сцене эффект освещения, который будет соответствовать тому пространству, которое показано на карте. [4,5]
6. Настройка рендера. Программа включает в себя встроенные движки для рендера: Cycles, Eevee. которые позволяют настроить резкость изображения, качество отображения теней, дымку от свечения световых объектов, имеют встроенные профили цвето-коррекции, [6]

Cycles является точным движком с поддержкой трассировки лучей, с помощью которого на выходе можно получить фотореалистичное изображение высокого качества. Он отлично подходит для обработки сцен с использованием тумана, дыма, различных отражений, поскольку Cycles обрабатывает подобные эффекты максимально приближенно к реальному виду. Однако, работа данного движка требует высокой производительности ПК для обработки изображения.

Движок Eevee является менее требовательной альтернативой Cycles, с его помощью тоже получают качественные изображения при высокой производительности. Его используют при создании видео и визуализации изображений с low-poly (низко полигональными) моделями. В отличие от Cycles, в движке не предусмотрена трассировка лучей и точная обработка отражений и света, тем не менее он также дает возможность добиться хорошего результата.

#### **Моделирование детского комплекса «Кремлевская стена»**

Для моделирования детской площадки использованы стандартные инструменты моделирования. Так, с помощью кривых были созданы все спиральные и изогнутые элементы. К примеру, для создания поручней в игровом комплексе была построена кривая по их форме и выдавлена (экструдирована) наружу методом Bevel > Round для придания округлой формы и объема (рис. 6, 7, 8). [1, 3]

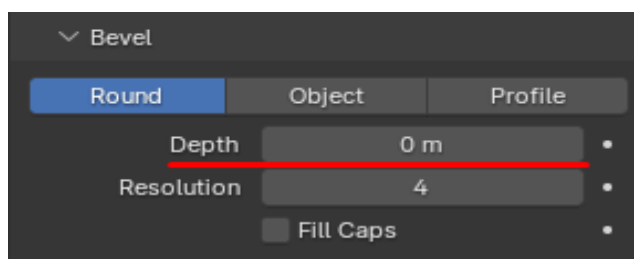


Рисунок 6. Параметры метода Bevel.

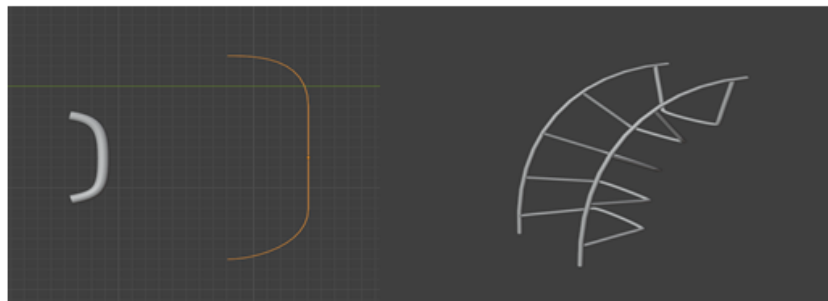


Рисунок 7. Модель ручки и рукоход.

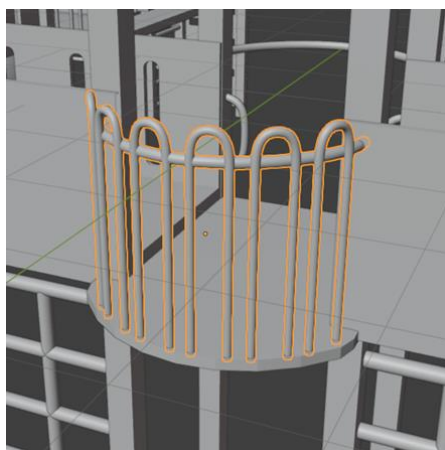


Рисунок 8. Винтовой спуск и ограждение.

Части, где необходимо множественное повторение элементов, были выполнены при помощи модификатора Array [1,2,3]. В нем присутствуют настройки размера массива для элемента, который нужно продублировать, и имеется возможность указать размер расстояния между каждой продублированной частью.

Некоторые элементы площадки имеют отверстия, и для их создания подходит модификатор Boolean. Данный модификатор дает возможность «вычесть» один объект из другого. Таким образом были сделаны ограждения безопасности в высоких частях комплекса (рис.9).

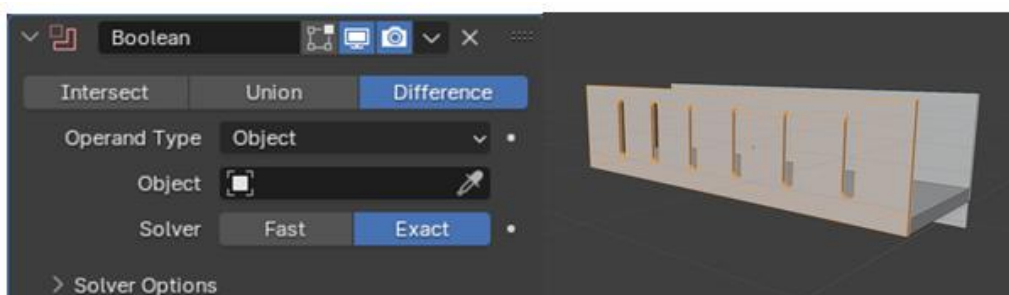


Рисунок 9. Модификатор Boolean и перегородка.

Для всех объектов были созданы материалы в разделе Shading, окрашенные в цвет соответственно референсам. Для смены цвета объекта редактировался параметр Base Color с использованием ноды (узла) Principled BSDF (рис. 10) [7].

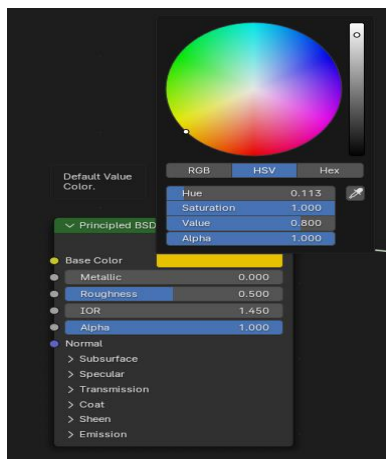


Рисунок 10. Настройка узла Principled BSDF.

Помимо цвета объекта данный узел позволяет настроить:

- поверхностное рассеивание, что влияет на пропускание материалом света сквозь себя и его последующее рассеивание внутри объекта;
- металлический блеск;
- интенсивность отражения света материалом;
- уровень шероховатости объекта;
- позволяет подключить карту нормалей;
- прозрачность.

#### Моделирование малых архитектурных элементов парка

В основе цепей, на которых подвешены планки для сидения, использовался тор, растянутый в направлении одной из осей. Достаточно продублировать фигуру несколько раз, повернуть элементы через один на 90 градусов, и композиция начинает выглядеть как цепь (рис. 11).



Рисунок 11. Моделирование цепей.

Так как качели состоят из множества повторяющихся в ряд балок, можно использовать модификатор Array. С его помощью был создан каркас качелей и объединен с цепями и сидениями (рис. 12).



Рисунок 12. Модель качелей и референс со скамейками.

Скамейки в парке располагаются по периметру лужаек и повторяют их форму (рис. 13). Для их моделирования использованы кривые и соответствующие модификаторы. Отдельно моделируются бруски основания (рис. 14).

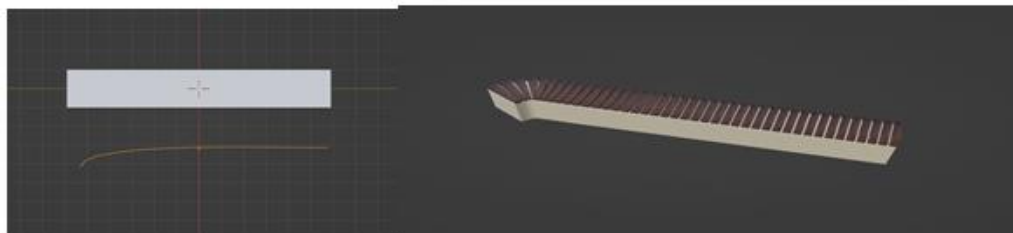


Рисунок 14. Моделирование скамеек.

В качестве базы для фонарей создается стандартный цилиндр и вытягивается по оси Z. Далее форму необходимо подразделить на сегменты и применить материалы для светящейся и металлической областей (рис. 15).

#### Моделирование музея «Россия – моя история»

Силуэт музея моделируется с помощью аффинных преобразований базового куба. Для реализации выдавленных на фасаде форм в виде деревьев используется модификатор Boolean. Для этого создается модель дерева, которая размещается на фасаде модели музея на необходимой глубине, а далее вычитается из него (рис. 16).

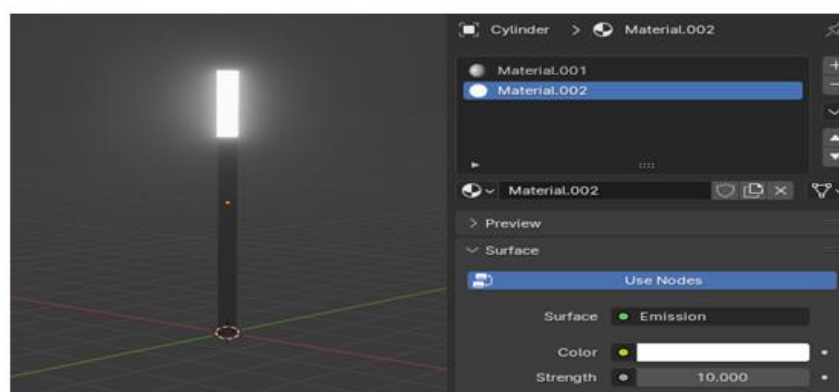


Рисунок 15. Разработка модели фонаря.



Рисунок 16. Оформление фасада здания музея.

#### Создание ландшафта и добавление объектов на общую сцену

Для того, чтобы в точности повторить очертания и расположение объектов в парке, была использована функция Images To Planes. Она позволяет добавить изображение в проект как плоскость с референсом, а затем воспроизвести форму газонов (рис. 17, 18).

После этого можно расставить остальные недостающие объекты, заготовленные ранее (рис. 19).

Для добавления растительности используется аддон «Botaniq» [8] (рис. 20, 21). Данная библиотека содержит встроенные модели различных видов растительности: деревьев и кустарников, разных типов газонов, которые менее нагружают проект, чем встроенный инструмент Particles.

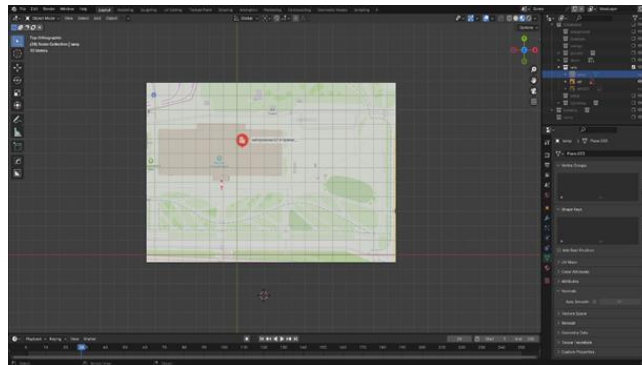


Рисунок 17. Добавление референса.

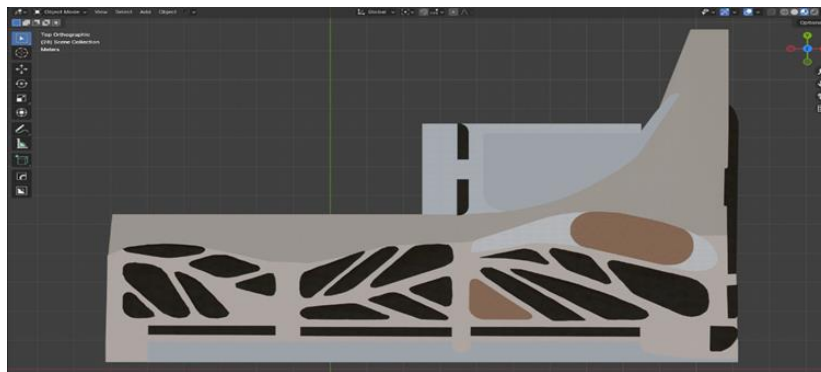


Рисунок 18. Моделирование газонов по референсам.

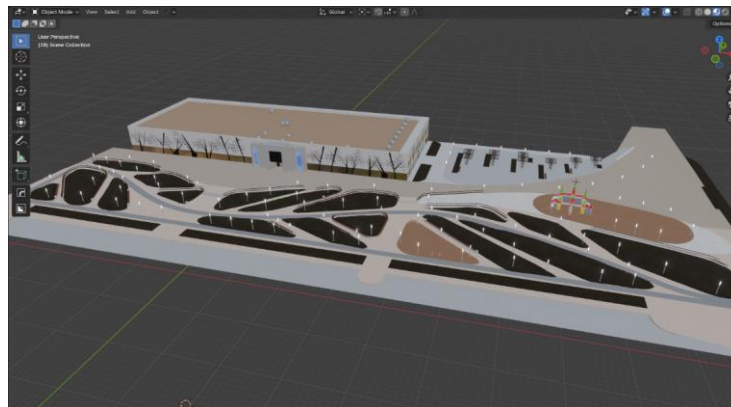


Рисунок 19. Расстановка малых архитектурных форм на сцене.

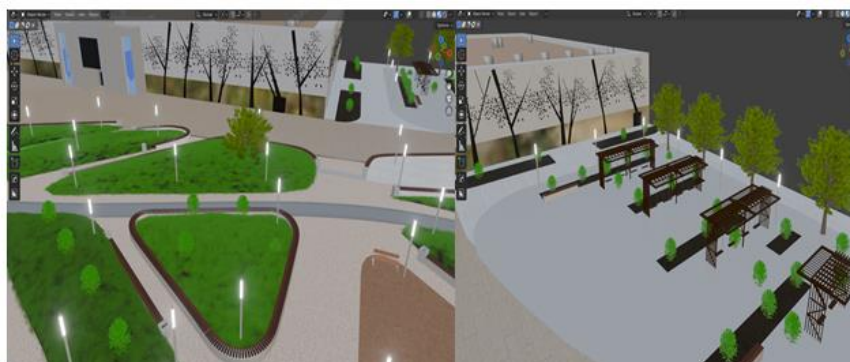


Рисунок 20. Применение аддона Botaniq: вид сбоку и вид сверху.

В настройке освещения помогли встроенные световые источники и HDRI-карта. Для фонарей использовался инструмент Point Lamp (рис. 21). Он излучает свет вокруг себя в некотором радиусе, который зависит от настроек мощности и рассеивания.

Глобальное освещение было задано инструментом Sun (рис. 22), смешанным с HDRI-картой (рис. 23), взятой из открытого источника.

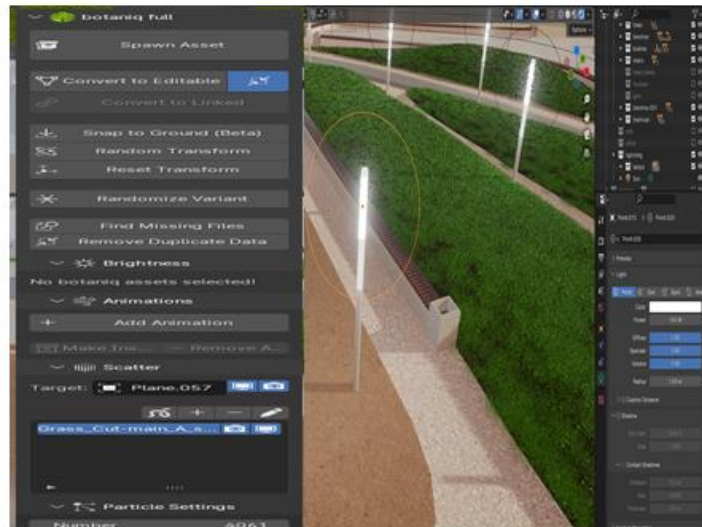


Рисунок 21. Параметры аддона Botaniq и освещение Point Lamp.



Рисунок 22. Настройка инструмента Sun.

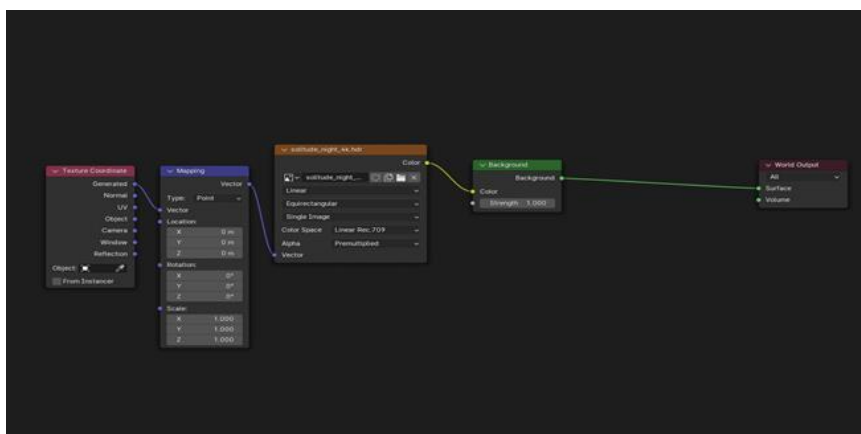


Рисунок 23. Настройка HDRI-карты.

### Визуализация итоговой сцены

Для рендеринга итоговой сцены (рис. 24, 25, 26) необходимо иметь выставленную в композицию камеру. Настройки композиции позволяют настроить четкость итогового изображения, параметры глубины освещения, рассеивания теней и цветовой коррекции.

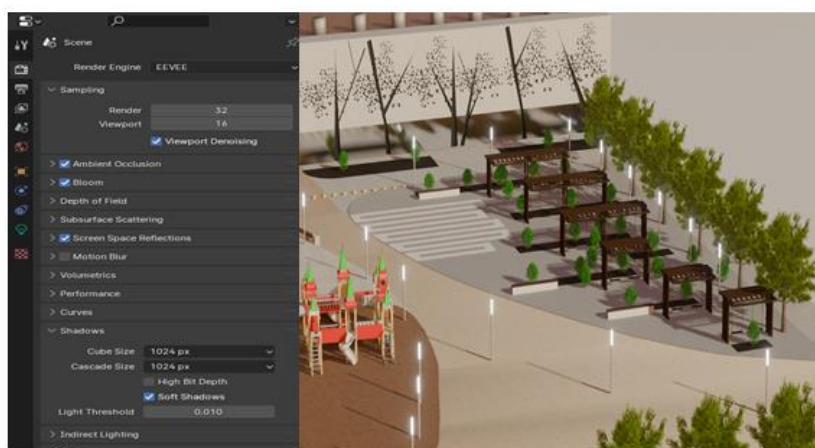


Рисунок 24. Настройка рендеринга сцены и финальный вид парка. Вид 1.



Рисунок 25. Финальный вид парка. Вид 2.

### Заключение

Результаты работы демонстрируют большие возможности пакета для трехмерного моделирования Blender для качественного моделирования реальных архитектурных объектов. Данный виртуальный архитектурный комплекс в дальнейшем может быть использован для образовательных и культурных целей.



Рисунок 26. Финальный вид парка. Вид 3.

Последующее развитие проекта может включать в себя моделирование новых объектов, перенос моделей в интерактивную среду [7] и улучшение качества смоделированных объектов.

\*\*\*

1. Блейн, Д.М. Полное руководство по графике в Blender: моделирование и анимация. — М.: Питер, 2020. — 672 с.
2. Валенца, Э. Blender. Практические рецепты. — СПб.: БХВ-Петербург, 2021. — 416 с.
3. Симондс, Б. Мастерство в Blender: Моделирование, скульптура, материалы и рендеринг. — М.: ДМК Пресс, 2020. — 416 с.
4. Валенца, Э. Blender Cycles. Материалы и текстуры. — М.: ДМК Пресс, 2019. — 352 с.
5. Кузнецов, А. Blender для архитекторов. — СПб.: Питер, 2019. — 288с.
6. Морозов, А. Blender для профессионалов. — СПб.: Питер, 2021. — 352 с.
7. Смирнов, П. Blender для Web-дизайнеров. — М.: ДМК Пресс, 2020. — 224с.
8. Документация по аддону Botaniq. – [Электронный ресурс]. URL: [https://docs. polygoniq. com/botaniq/6.4.3/](https://docs.polygoniq.com/botaniq/6.4.3/) (дата обращения 21.11.2024)

**Гурзов Н.А., Переверзев А.Л., Силантьев А.М.,  
Примаков Е.В., Солодовников А.П.**

**Разработка реконфигурируемого систолического массива для ускорения  
вычисления слоёв нейронной сети**

*Национальный исследовательский университет «МИЭТ»  
(Россия, Москва)*

*doi: 10.18411/trnio-12-2024-868*

#### **Аннотация**

Применение систолических массивов позволяет ускорить тензорные операции, исключив необходимость в перемещении промежуточных результатов через подсистему памяти. Открытые реализации решают узкий спектр задач и не имеют реконфигурации. Для решения указанной проблемы была разработана реконфигурируемая структура систолического массива, в основе которой лежат принципы: Output-stationary (OS) и Weight-stationary (WS). Результаты данной работы могут быть полезны для проектов, требующих гибкости в реализации тензорных вычислений.

**Ключевые слова:** систолический массив, output-stationary, weight-stationary, реконфигурируемая структура, квантование, нейронные сети.

#### **Abstract**

The use of systolic arrays allows you to speed up tensor operations, eliminating the need to move intermediate results through the memory subsystem. Open implementations solve a narrow range of tasks and do not have reconfiguration. To solve this problem, a reconfigurable systolic array structure was developed, which is based on the principles: Output-stationary (OS) and Weight-stationary (WS). The results of this work can be useful for projects that require flexibility in the implementation of tensor calculations.

**Keywords:** systolic array, output-stationary, weight-stationary, reconfigurable structure, quantization, neural networks.

#### **Введение**

Систолические массивы широко применяются при проектировании гетерогенных вычислительных систем и СФ-блоков, которые используются главным образом в областях, где преобладают итеративные процедуры (численная математика, комбинаторная оптимизация, обработка изображений, сигналов, речи, текста и т.д.).

Интеграция систолического массива на этапе обучения нейронных сетей и в процессе работы уже обученной нейросети на конечном устройстве (inference) позволяет существенно ускорить вычисления, при этом исключив необходимость в перемещении промежуточных результатов через подсистему памяти.

Существующие открытые аппаратные реализации и структуры базовых элементов систолических массивов позволяют решать только узкий спектр задач и не имеют возможностей реконфигурации, что не даёт обеспечить максимальную гибкости использования и не позволяет расширить спектр решаемых задач.

### Принципы построения систолических массивов

По способу организации межсоединений между Processing Elements (PE), систолические массивы могут быть устроены по-разному. Существуют одномерные (векторные) массивы, состоящие из одной строки или одного столбца PE. Также есть многомерные реализации, например двумерные (прямоугольные и шестиугольные) включающие несколько строк и столбцов PE, или пространственные-трёхмерные.

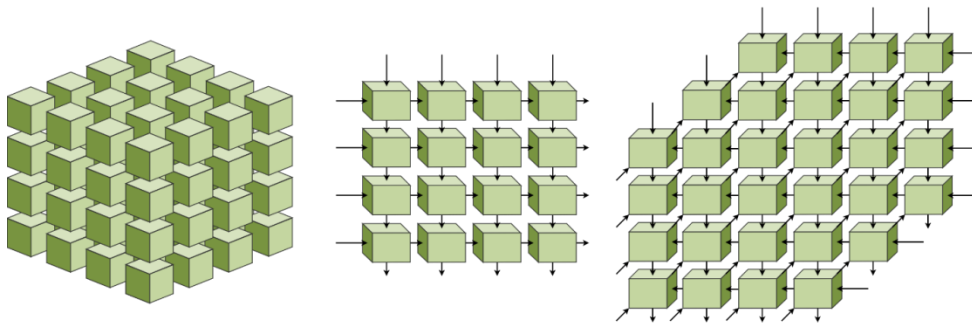


Рисунок 1. Способы построения систолических массивов.

По принципу управления систолические системы можно разделить на массивы с PE без управления, с локальным, глобальным и смешанным типом управления.

### Внутренняя организация вычислительных блоков

По способу внутренней организации и принципу работы ячейки систолического массива (PE) могут быть устроены следующим образом. Существует вариант со статичной суммой (Output-stationary), где на вход ячейки одновременно подаются данные и веса, а сумма их произведения с предыдущими результатами хранится в регистре внутри ячейки.

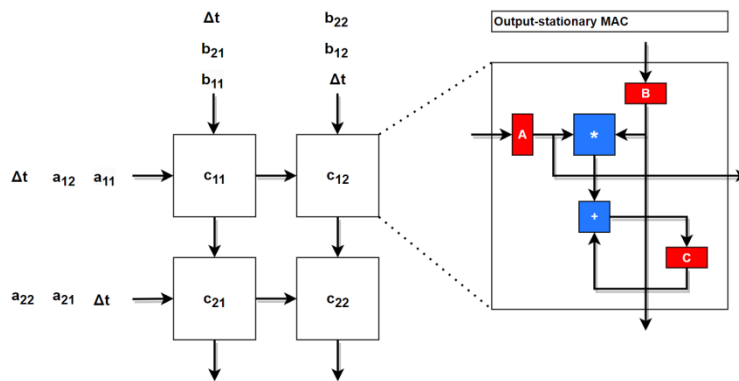


Рисунок 2. Схема алгоритма умножения с накоплением Output-stationary.

Также существует вариант со статичным весом (Weight-stationary), где на вход ячейки одновременно подаются данные и частичная сумма, вес хранится в самой ячейке.

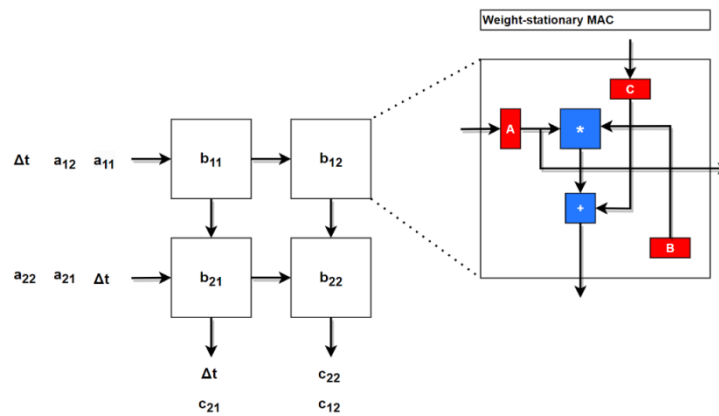


Рисунок 3. Схема алгоритма умножения с накоплением Weight-stationary.

### Квантование данных

С ростом популярности глубокого обучения возросла потребность в небольших решениях для нейронных сетей с низкой задержкой и высокой энергоэффективностью. Одним из наиболее эффективных способов сократить время вычислений и энергопотребление нейронных сетей является квантование. При квантовании нейронных сетей веса и тензоры активации хранятся с более низкой точностью, чем 16-ти или 32-х разрядная точность, на которой они обычно обучаются. При переходе от 32 к 8 битам нагрузка на память для хранения тензоров уменьшается в 4 раза, в то время как вычислительные затраты на умножение матриц уменьшаются в квадратичном выражении в 16 раз.

### Реализация Processing elements

Для решения проблем, связанных с гибкостью использования и уменьшением вычислительных затрат была разработана реконфигурируемая структура двумерного прямоугольного (COLxROW) систолического массива, в основе которой лежит два базовых принципа работы: Output-stationary (OS) и Weight-stationary (WS). Отличительной особенностью разработанного массива являются возможность переключения ширины потоков входных данных в Processing elements (PE). Систолический массив может использоваться в формате матрицы COLxROW с шириной входных данных 16-бит и весов 32-бит в каждом PE, что позволяет производить квантованные вычисления с высокой точностью. Для уменьшения энергопотребления данные передаются между PE вместе с valid, что позволяет исключить лишние переключения регистров. Реализованный систолический массив имеет аппаратную поддержку вычислений с квантованными данными.

### Объединение Processing elements в систолический массив

Модуль генерации систолического массива (draco\_syst\_matrix) принимает на вход 64-bit слова ( $a_i$  и  $m_i$ ), в которых содержатся либо  $8 \times 8$ -bit чисел, либо  $4 \times 16$ -bit числа (за размер отвечает  $size_i$ ). Сигналы валидности входных данных ( $valid_a_i$  и  $valid_m_i$ ), а также управляющие сигналы:

1.  $sync_i$  - сигнал синхронизации массива по столбцам;
2.  $state_i$  - управляет режимом работы массива (состояние КА);
3.  $enable_i$  - глобальное разрешение на запись в регистры.

На вход модуля поступают данные ( $a_i$ ,  $m_i$ ,  $m_o$ ,  $m_o$ ,  $m_o$ ,  $m_o$ ) и управляющие сигналы логики квантования.

Сначала данные поступают в модули ZP\_A\_i и ZP\_M\_i, где происходит вычитание из данных и весов zero-point. Далее данные поступают на входы массива через буферы, что позволяет обеспечить пространственно-временную синхронизацию данных в систолическом массиве.

В каждой ячейке происходит перемножение пришедших весов с данными, сложение результата умножения с суммой предыдущих вычислений и дальнейшая передача данных по матрице. Вычисления в каждом отдельном PE производятся только, если на его входы пришли валидные веса и данные. На рис.4 приведена структурная схема входов систолической матрицы. На схеме представлено подключение буферов A/M\_i\_BUF, модулей ZP\_A/M\_i, и Processing elements (PE).

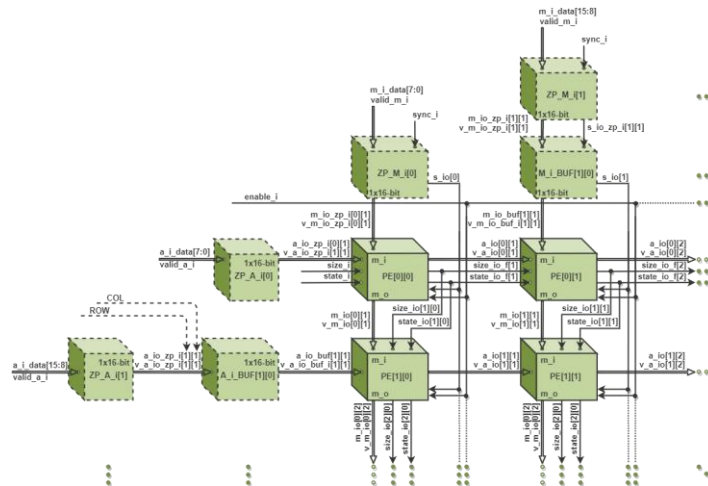


Рисунок 4. Схема входов систолического массива.

После прохождения матрицы PE данные поступают на вход модулей BIAS\_M\_o, где происходит их сложение с bias с целью коррекции смещения в распределении. Затем данные попадают в модули SCALE\_M\_o, где происходит их перемножение с масштабирующим коэффициентом scale, чтобы выровнять результат относительно фиксированной точки. Затем в модулях ZP\_M\_o происходит сложение результатов с zero-point, это позволяет гарантировать, что действительный ноль квантуется без ошибок. После чего данные приводятся к диапазону значений от -128 до +127 (int8). Далее данные поступают на выходы массива через буферы, что позволяет обеспечить пространственно-временную синхронизацию данных на выходе систолического массива. На рис.5 приведена структурная схема выходов систолической матрицы. На схеме представлено подключение Processing elements (PE), модулей BIAS\_M\_o, SCALE\_M\_o, ZP\_M\_o и буферов буферов BUF\_M\_o.

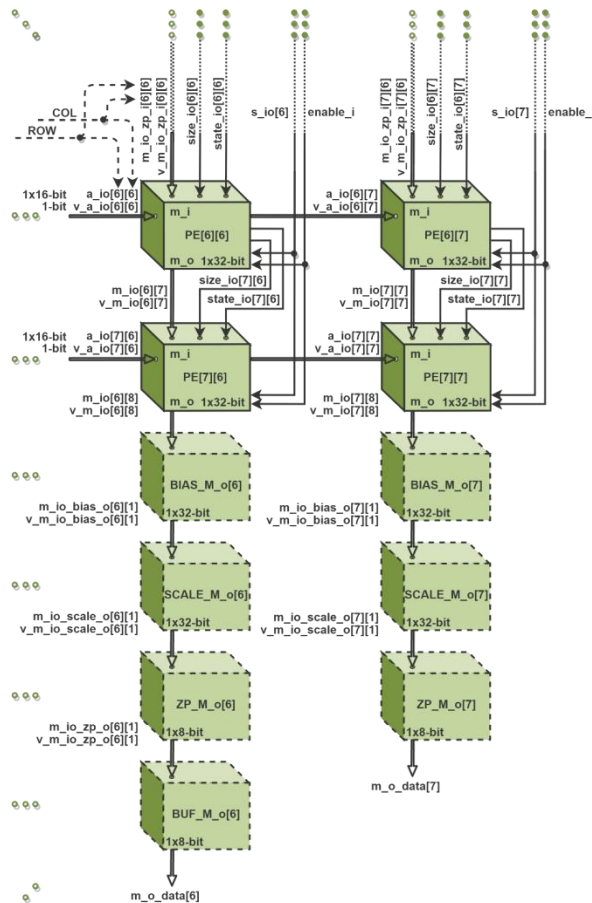


Рисунок 5. Схема выходов систолического массива.

### Производительность

В зависимости от режима работы систолического массива время, затраченное на выполнение умножения матриц, будет отличаться. В режиме со статичной суммой (Output-stationary) на ввод весов и данных в массив, вычисление и вывод результата через логику квантования уйдёт в общей сложности 39 тактов. В режиме со статичным весом (Weight-stationary) на загрузку весов, вычисление с одновременной загрузкой данных и вывод результата через логику квантования уйдёт в общей сложности 40 тактов. Отличие в один такт между режимами работы обусловлено необходимостью синхронизировать подачу весов в режиме weight-stationary, чтобы все веса при загрузке остановились в нужных PE. На рис.6 приведена продолжительность вычислений в двух режимах на разных этапах работы систолического массива, указанная в тактах.

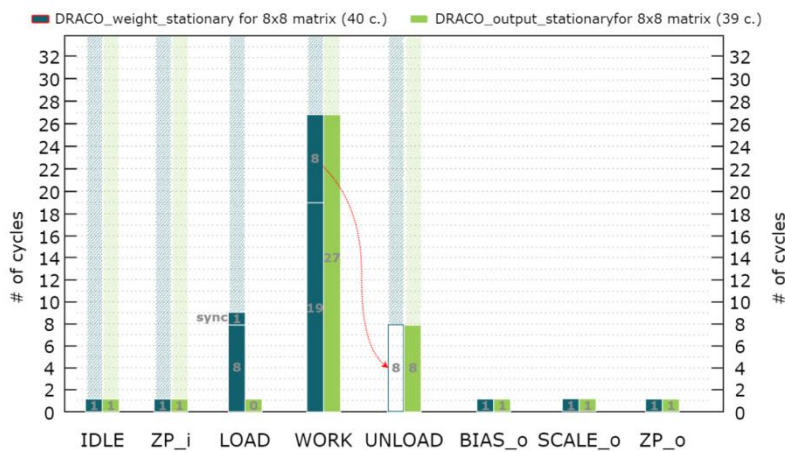


Рисунок 6. Продолжительность разных этапов вычислений в тактах.

### Результаты исследования

Проведенное сравнение с существующими аппаратными реализациями показало, что все функции вычислений в двух режимах были полностью реализованы. Вариант с использованием DSP-блоков продемонстрировал сопоставимую с конкурентами рабочую частоту при меньшей утилизации LUT. В то же время аппаратная версия, реализованная без применения DSP-блоков, уступает аналогам по рабочей частоте из-за отсутствия в текущей реализации специализированных сумматоров и умножителей с сокращенным критическим путем (Kogge-Stone, Wallace-tree и т.д.). Аппаратные затраты реализации без DSP-блоков оказались выше, чем у сравниваемых решений. Однако следует учитывать, что разрядность вычислений в представленном систолическом массиве выше разрядности аналогичных систем. Данное решение сохраняет преимущества масштабируемости и гибкой структуры. На рис.7 представлена диаграмма сравнения характеристик различных реализаций массивов.

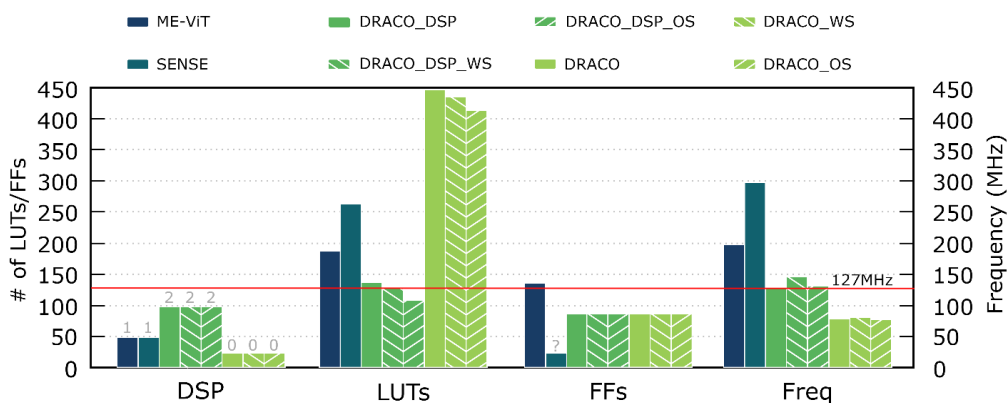


Рисунок 7. Сравнительные характеристики.

### Заключение

Конечным результатом работы является разработанный систолический массив. Благодаря объединению нескольких режимов работы реализованный массив обладает гибкой реконфигурируемой структурой и при этом не уступает существующим аналогам.

\*\*\*

1. Systolic Systems 12.1. Basic Concepts of Systolic Systems, URL: <http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/03-Algorithms%20of%20Informatics%201,%202,%203/Systolic30May.pdf> (дата обращения: 14.11.2024).
2. Understanding Matrix Multiplication on a Weight-Stationary Systolic Architecture, URL: <https://telesens.co/2018/07/30/systolic-architectures/> (дата обращения: 14.11.2024).
3. Sense: Model Hardware Co-design for Accelerating Sparse CNN on Systolic Array, Wenhao S., Deng L., Zhiwei Z., Wendi S., Yi K., Song C., URL: <https://arxiv.org/pdf/2202.00389> (дата обращения: 14.11.2024).
4. ME-ViT: Accelerator for Vision Transformers, Kyle M., Pengmiao Z., Viktor P. URL: <https://arxiv.org/pdf/2402.09709> (дата обращения: 14.11.2024).
5. Paper on Neural Network Quantization, Markus N., Marios F., Rana A. A., Yelysei B., Mart van B., Tijmen B. URL: <https://arxiv.org/pdf/2106.08295> (дата обращения: 14.11.2024).

### Данаев М.Х., Магамадов А-С.С., Шахбазова М.С. Перспективы внедрения нейросетей в образование

*Чеченский государственный университет имени А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-12-2024-869

#### Аннотация

В этой статье мы будем говорить о перспективе удачного внедрения нейросетей в образование. Есть ли вообще шанс на выживание у этой идеи и будет ли иметь смысл для предприятий и организаций реализовывать этот проект, и если да, то в каких именно проектах стоит использовать нейронные сети.

**Ключевые слова:** нейросети, образование, перспектива, искусственный интеллект, программа, инструмент.

#### Abstract

In this article, we will talk about the prospects for the successful implementation of neural networks in education. Is there a chance for this idea to survive at all and will it make sense for enterprises and organizations to implement this project, and if so, in which projects exactly should neural networks be used.

**Keywords:** neural networks, education, perspective, artificial intelligence, program, tool.

#### Введение

Хотя нейросетям в образовании в настоящий момент посвящено уже множество работ, эта тема не является полностью разработанной или решённой. Образование на основе нейросетевых технологий представляет собой многогранную систему, включающую целый ряд аппаратных и программных подсистем.

В этой статье мы не будем рассматривать технические аспекты и процессы разработки программных и алгоритмических приложений. Мы поговорим о возможности реализации и о том, какой характер для сферы образования будет нести тема использования нейросетей в образовании.

Есть ли смысл использования нейросетей в образовании?

Сегодня нейронные сети и искусственный интеллект все глубже входят в нашу жизнь. Компьютеры уже научились писать различные тексты, рисовать картины, обрабатывать данные и выдавать ответы на самые разнообразные задачи и, конечно же, решать задачи вместо школьников. Школьную программу искусственный интеллект знает

на отлично, дети по всему миру используют такие чат боты, как ChatGPT, который способен выдать ответ меньше чем за минуту. Школьники всегда списывали домашние задания, раньше у друзей или соседей по парте, сегодня же у искусственного интеллекта. Он способен решать задачи по математике с подробным пояснением решения или написать вполне логичное сочинение. Конечно, отличить результаты работ машин от работ реального ученика все еще можно, но это крайне сложно.

В каких проектах имеет смысл использовать нейросети

Нейросети могут быть полезны даже там, где казалось бы о вмешательстве машинных технологий не может идти и речи. Ученые использовали искусственный интеллект для подбора людей в команду. Это должны были быть не просто коллеги, а максимально эффективные коллективы, которые смогут создавать и реализовывать успешные стартапы.

Нейросети могут применяться и при организации времени. Машина может помочь составить ежедневное расписание, включающее в себя расписание учебных занятий, домашних заданий, кружков и других видов деятельности.

Для тех, кто учит языки или хочет развить свою дикцию нейросеть будет так же полезна. Ты можешь вести диалоги на различных языках и ИИ будет анализировать твою речь и указывать на ошибки.

Это были лишь несколько примеров успешной реализации нейросетей в образовании, не стоит полагать, что на этом польза нейросетей заканчивается. Вышеперечисленные примеры используются на сегодняшний день в различных школах и детских образовательных учреждениях.

Как искусственный интеллект может помочь в образовании

В то время как в России не запрещено использование нейронных сетей в учебных работах, в нескольких штатах США не разрешают использовать ChatGPT из-за опасений, что нейросети негативно повлияют на учащихся. Такое решение приняли и такие страны, как: Япония, Германия, Италия, Испания и некоторые другие.

Думаю, что все согласятся в том, что развитие нейросетей не остановить и вместо того, чтобы бороться с ними лучше как можно раньше интегрировать их в образования, ибо трансформация нынешней системы образования лишь вопрос времени.

Нейросети и искусственный интеллект можно применять для автоматизации рутинных процессов. Уже сегодня во многих школах страны используются виртуальные приложения, такие как дневники и журналы. Именно учителям приходится заполнять все это, учитель должен считать баллы, следить за успеваемостью всех учеников и проверять кучу типовых заданий. Внедрение ИИ в образование позволит передать эти задачи машине при этом освобождая время для учителей.

Также он отлично служит и для генерации “контента”. Это то, чем лично я и большинство студентов высших учебных заведений пользуются повседневно - самая привычная для нас область. Сети, которые генерируют тексты можно использовать в качестве вспомогательного инструмента при написании статей, докладов, курсовых работ и прочего. Существуют и нейросети помогающие при подготовке презентаций, при написании кода.

Проблемы и вызовы при использовании нейросетей в образовании

Преимущества нейросетей очевидно - они помогают учителям находить учебный материал, ученикам помогают гораздо легче и быстрее решать различного рода задачи, но не стоит забывать и о минусах и недостатках использования таких машин. Во-первых, преподаватель может использовать программу для оценивания знаний учащихся, но это может привести к необъективности и предвзятости.

Во-вторых, проблема контроля над информацией. Использование нейросетей может привести к утечке персональных данных учеников, если должным образом не следить и не защищать данные.

В-третьих, студенты могут потерять мотивацию и тягу к получению новых знаний, при частом использовании ИИ-машин. Если все время прибегать к помощи нейросетям, то существует риск, что без них учащиеся разучатся мыслить.

Не стоит забывать и цифровом неравенстве в разных слоях населения. Не у всех есть возможность получить доступ к цифровым устройствам, что приведет к неравенству в сфере образования.

#### Заключение

В заключении хочется отметить, что искусственный интеллект вскрыл проблемы современной системы образования. Бот может написать сочинение, решить задачу по математике, и ученик может прийти в школу с этими “решенными” домашними заданиями и учителя поставят им высокие баллы. Ученики вместо своих знаний переписывают формулировку машины, потому что школа не мотивирует их думать, основная мотивация – это стремление получить высокий балл, а школьники лишь ищут самый легкий путь.

Нейросети должны рассматриваться не как замена преподавателя, а как инструмент, обеспечивающий автоматизацию рутинных задач, в результате чего у преподавателя освобождается время для работы с учениками.

\*\*\*

1. Павлов Д.А. Искусственные нейросети в контексте науки и образования. Компьютерные инструменты в образовании. 2017; №6.
2. Васильев А.В., Попова Ю.Б. Применение искусственных нейронных сетей в образовательном процессе. 2018. Available at: <https://rep.bntu.by/bitstream/handle/>
3. Козлова О.А., Протасова А.А. Использование нейронных сетей в дистанционных образовательных технологиях для идентификации обучающихся // Открытое образование. 2021. Т. 25. №3. С. 26–35. DOI: 10.21686/1818-4243-2021-3-26-35.
4. Казаченок В.В. Применение нейронных сетей в обучении. Информатика и образование. 2020. №2(311). С. 41–47. DOI: 10.32517/0234-0453-2020-35-2-41-47.
5. Боровская Е. В., Давыдова Н. А. Основы искусственного интеллекта: учебное пособие. 4-е изд. М.: Лаборатория знаний, 2020.

**Данаев М.Х., Магамадов А-С.С., Шахбазова М.С.**

#### **Программирование с нуля: с чего начать и какой язык выбрать**

*Чеченский государственный университет имени А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-12-2024-870

#### **Аннотация**

Это статья - ваш путеводитель в мир программирования. Мы расскажем вам о первых шагах, которые вам предстоит сделать на вашем пути, о необходимых инструментах, а также поможем вам выбрать подходящий язык программирования для ваших целей. Вы узнаете о популярных и востребованных языках, таких как Python, C++ и Java, и поймете какой из этих языков подходит для новичка. Статья содержит практические советы и рекомендации, которые помогут вам заложить прочный фундамент для дальнейших успехов в мире кода.

**Ключевые слова:** Python, C++, JavaScript.

#### **Abstract**

This article is your guide to the world of programming. We will tell you about the first steps that you have to take on your way, about the necessary tools, and also help you choose the right programming language for your goals. You will learn about popular and in-demand languages such as Python, C++ and Java, and understand which of these languages is suitable for

a beginner. The article contains practical tips and recommendations that will help you lay a solid foundation for further success in the world of code.

**Keywords:** Python, C++, JavaScript.

Сегодня программирование уже не является узконаправленной специальностью, оно является высокооплачиваемой и востребованной областью. Это статья будет полезна для тех, кто всегда мечтал освоить этот навык, но не знал с чего стоит начать. В данной статье шаг за шагом будут разобраны все этапы вхождения в мир программирования, начиная с подготовки всего необходимого программного обеспечения и заканчивая выбором вашего первого языка. Вы освоите ключевые концепции кода, базовые принципы алгоритмизации и логики, а также разберетесь в чем же преимущества и недостатки того или иного языка программирования, таких как Python, C++ и Java. Вы будете готовы сделать первый и уверенный шаг в становлении программистом и освоению одной из самых перспективных профессий XXI века.

#### Операционная система

Для начала необходимо выбрать операционную систему. В первую очередь следует отметить, что инструменты, с которыми вы будете работать предназначены для вашего удобства и простоты выполнения различных задач. Если говорить о выборе операционной системы, то стоит рассмотреть их плюсы и минусы.

Первое место следует разделить между macOS и Ubuntu/Mint (и другие Linux-дистрибутивы). Именно для разработчика это семейство ОС является лучшим. Из плюсов нужно отметить «нативную» работу Docker и близость с production-окружением. Из минусов - не всегда оптимально используются ресурсы и гораздо меньше софта для разработчика.

Второе место конечно же Windows. Это базовая ОС для многих специалистов, также тут можно размеренно вести разработку. При наличии таких софтов как Docker или Vagrant многие проблемы легко обходимы, а также тут намного больше софта чем на Ubuntu или macOS. Основным минусом является отсутствие нормального терминала. Однако итоговое решение стоит предпринимать не из плюсов или минусов той или иной операционной системы, а из удобства и комфорта разработчика.

#### Программное обеспечение

Текстовые редакторы играют ключевую роль в процессе разработки. Хороший текстовый редактор может улучшить качество разработки, облегчить отладку, а также значительно повысить продуктивность. Все это в совокупности приведет к значительному улучшению качества кода. Текстовые редакторы обеспечивают подсветку синтаксиса, автодополнение, отладку и еще много других полезных функций, которые делают разработку приятной для самого программиста.

Рассмотрим популярные текстовые редакторы:

Visual Studio Code (VS Code) - это бесплатный редактор от компании Microsoft, который стал одним из самых популярных редакторов, которым пользуется огромное количество разработчиков по всему миру. Он поддерживает множество языков программирования и имеет богатую библиотеку всевозможных расширений. Однако минусом этого редактора является то, что он требователен к характеристикам вашего компьютера.

Sublime text - это редактор кода с поддержкой многих языков, например, Python, C++, Java, HTML и другие. Sublime text также превосходит другие редакторы своей высокой производительностью, однако, некоторые функции поддерживаются только при платной подписке.

Notepad++ редактор кода предназначенный для работы на ОС Windows. Одним из главных плюсов этого редактора является возможность настроить практически все функции под себя. Тут можно изменять, например, подсветку синтаксиса, причем для каждого языка есть возможность настроить отдельную подсветку, добавлять новые

языки, изменять цветовую схему и многое другое. Один существенный недостаток — неспособность обрабатывать тексты на нескольких языках одновременно. В случае многоязычного документа программа не сможет корректно его интерпретировать.

#### Логика и алгоритмы

Программа функционирует на основе алгоритмов — последовательностей инструкций, обеспечивающих решение конкретных задач. Эти инструкции определяют порядок действий, ведущих к желаемому результату.

Эффективная разработка программного обеспечения невозможна без логического проектирования. Логика — это основа программирования, определяющая создание рациональных алгоритмов. Она позволяет предвидеть последовательность шагов и потенциальные ошибки.

#### Переменные и типы данных

В программировании переменные представляют собой хранилища информации. Они могут содержать данные различных типов. Важно помнить, что значения в этих хранилищах изменяются по мере выполнения программы.

Переменные могут хранить числовые данные, текст, аудио- и визуальные материалы. Чтобы использовать переменную, ей необходимо присвоить значение. Оператор присваивания (=) связывает переменную с её значением. Этот процесс называется присваиванием.

#### Условные операторы

Условные операторы — ключевой элемент программирования. Они позволяют принимать решения в зависимости от условий, что делает код более гибким. Без них код был бы линейным и не адаптировался бы к нестандартным ситуациям. Фундаментальная условная конструкция — if-else. Она проверяет условие: если оно истинно, выполняется один блок кода, если ложно — другой. Это позволяет программе принимать решения и выполнять различные действия в зависимости от заданных условий.

#### Циклы

Цикл - это конструкция, которая заставляет какой-то блок кода выполняться несколько раз.

Основные типы циклов - это циклы for, while и do-while. Понимание этих циклов поможет вам упростить вашу программу при работе с условными конструкциями. Цикл for является самым простым и одним из самых распространенных и универсальных циклов. Он состоит из трех частей: инициализация, условие и итерация.

Цикл while проверяет условие перед каждой итерацией, если условие истинно выполняется тело цикла, если ложно, цикл завершается.

Цикл do-while выполняет тело цикла хотя бы один раз, после чего проверяет условие.

#### Функции

В программировании, функция - это отдельный блок программы, который выполняет одно конкретное действие. Функция решает проблему повторения одного и того же действия и помогает разбить сложную задачу на более простые подзадачи. Это упрощает понимание и написание кода для разработчика. Именно после понимания принципа работы функций для вас открывается дверь к эффективному программированию, после этого вы сможете начать создавать сложные программы с меньшими усилиями.

Функция - это не просто блок кода, а инструмент, который позволяет программистам избежать написание одного и того же кода и сделать программу более читаемой.

Чтобы использовать ранее определенную функцию, вам нужно в требуемом месте программы написать имя функции и передаваемые в нее параметры.

### Заключение

В заключении хочется сказать, что идеальной инструкции как стать программистом не существует. Главное - это практика, практика и еще раз практика! Не бойтесь ошибок. Чем больше вы будете пробовать, чем больше ошибаться и исправлять свои ошибки, тем быстрее вы будете учиться.

Программирование - это как изучение нового языка. Вначале будет сложно, но с практикой будет становиться все проще и проще.

\*\*\*

1. McConnell, S. "Code Complete by Steve McConnell", 2017.
2. Thomas, D. & Hunt, A. "The Pragmatic Programmer: Your Journey to Mystery", 20th Anniversary Edition (2nd Edition), 2020.
3. Spraul, V.A. "Think Like a Programmer: An Introduction to Creative Problem Solving", 2018.
4. Martin, R.C. "The Clean Coder: A Code of Conduct for Professional Programmers", 2019.
5. Knuth, D. "The Art of Computer Programming", 2020.

**Даниленко В.С.**

### Показатели качества в сети IoT и методы их обеспечения

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-871

### Аннотация

В статье анализируются ключевые показатели качества (KQI) в сетях IoT, включая задержку, пропускную способность, надежность, энергопотребление и безопасность. Рассмотрены методы их обеспечения, такие как edge computing, оптимизация протоколов и энергосберегающие технологии. Описаны подходы к оценке качества с использованием мониторинга, моделирования и аналитических платформ. Практические примеры демонстрируют эффективность решений в умных городах, промышленности и сельском хозяйстве.

**Ключевые слова:** интернет вещей, ключевые показатели качества, задержка, пропускная способность, надежность, энергопотребление.

### Abstract

The article analyzes key quality indicators (KQI) in IoT networks, including latency, bandwidth, reliability, power consumption and security. The methods of their provision, such as edge computing, protocol optimization and energy-saving technologies, are considered. Approaches to quality assessment using monitoring, modeling and analytical platforms are described. Practical examples demonstrate the effectiveness of solutions in smart cities, industry and agriculture.

**Keywords:** internet of Things, key quality indicators, latency, bandwidth, reliability, power consumption.

Интернет вещей (IoT) играет ключевую роль в цифровой трансформации, интегрируя устройства и системы для автоматизации и повышения эффективности в сферах умных городов, промышленности, транспорта и здравоохранения. Эти технологии создают новые возможности для сбора и анализа данных, формируя интеллектуальные экосистемы. Стабильная работа IoT-сетей требует высокого качества взаимодействия между устройствами, особенно в системах реального времени, где даже минимальные сбои могут привести к серьезным последствиям.

Эффективность и стабильность работы сетей IoT во многом определяются ключевыми показателями качества (Key Quality Indicators, KQI). Эти показатели служат

основой для оценки производительности и надежности системы в различных сценариях применения, от умных домов до промышленных комплексов. На рисунке 1 представлены основные ключевые показатели качества в сети IoT. Их значимость и подходы к их обеспечению рассматриваются далее.



Рисунок 1. Ключевые показатели качества (KQI) в сети IoT.

Задержка (Latency) — это время, которое требуется для передачи данных от источника к получателю. Этот показатель особенно важен для приложений реального времени, таких как управление транспортом или медицинский мониторинг. Минимизация задержки достигается за счет использования edge computing и легковесных протоколов, что позволяет обрабатывать данные быстрее и локально.

Пропускная способность (Throughput) отражает объем данных, которые могут быть переданы через сеть за единицу времени. Высокая пропускная способность необходима для приложений, генерирующих большие объемы информации, таких как видеонаблюдение. Оптимизация сети и компрессия данных помогают увеличить этот показатель даже при высокой плотности подключений.

Надежность (Reliability) характеризует способность сети передавать данные без потерь. Она критически важна для промышленных систем и умных городов. Надежность повышается за счет применения устойчивых протоколов связи, коррекции ошибок и дублирования маршрутов.

Энергопотребление (Energy Consumption) является ключевым показателем для устройств IoT, работающих от батарей. Продление срока службы достигается использованием узкополосных технологий связи, гибридных режимов работы и адаптивного управления мощностью передачи данных.

Безопасность (Security) обеспечивает защиту данных и предотвращение несанкционированного доступа. Методы шифрования, аутентификация устройств и блокчейн-технологии играют важную роль в создании безопасных IoT-сетей, особенно в критически важных приложениях.

После анализа ключевых показателей качества сетей IoT, таких как задержка, пропускная способность, надежность, энергопотребление и безопасность, логично перейти к обсуждению методов, которые позволяют поддерживать эти параметры на необходимом уровне. Каждый из этих показателей требует индивидуального подхода, но в то же время их оптимизация должна учитывать комплексные особенности сетей IoT: ограниченные ресурсы устройств, разнообразие протоколов и высокую плотность подключений. В таблице 1 рассмотрены конкретные методы и технологии, которые применяются для обеспечения высокого качества работы IoT-сетей.

Таблица 1

## Методы обеспечения ключевых показателей качества в сети IoT.

Ключевой показатель качества	Методы обеспечения
Задержка (Latency)	<i>Edge и fog computing: обработка данных ближе к источнику для сокращения времени передачи.</i>
	<i>Оптимизация протоколов: использование легковесных протоколов, таких как CoAP и MQTT.</i>
Пропускная способность (Throughput)	<i>Частотное разделение спектра: технологии OFDMA для параллельной передачи данных.</i>
	<i>Сжатие и фильтрация данных: передача только ключевых данных, например, событий с камер.</i>
	<i>Балансировка нагрузки: динамическое перераспределение ресурсов для предотвращения перегрузки.</i>
Надежность (Reliability)	<i>Коррекция ошибок: использование алгоритмов FEC и механизмов повторной передачи данных (ARQ).</i>
	<i>Устойчивые протоколы связи: применение Zigbee и LoRaWAN для работы в условиях высокой потери пакетов.</i>
Энергопотребление (Energy Consumption)	<i>Гибридные режимы работы: переход в спящий режим в периоды бездействия.</i>
	<i>Энергоэффективные протоколы: использование BLE и Zigbee для минимизации затрат энергии.</i>
	<i>Адаптивная мощность передачи: регулировка мощности сигнала в зависимости от расстояния до шлюза.</i>
Безопасность (Security)	<i>Шифрование данных: применение легковесных алгоритмов, таких как AES-128 или ECC.</i>
	<i>Аутентификация и авторизация: использование многофакторной аутентификации и цифровых сертификатов.</i>

Каждый из этих методов направлен на оптимизацию определенного показателя качества, но их эффективное внедрение требует учета взаимосвязей между показателями. Например, повышение пропускной способности может увеличить энергопотребление, а внедрение сложных механизмов безопасности может повысить задержку. Поэтому выбор методов должен быть основан на анализе требований конкретного приложения и возможностей инфраструктуры IoT. Такой комплексный подход позволяет создавать устойчивые, надежные и эффективные IoT-сети.

Оценка качества в IoT-сетях играет ключевую роль в поддержании их стабильной работы и выявлении узких мест. Эта задача требует системного подхода, включающего методы оценки, анализ отдельных ключевых показателей и использование специализированных инструментов. На рисунке 2 представлена структура основных элементов оценки качества.



Рисунок 2. Основные элементы оценки качества в IoT-сетях.

Методы оценки качества позволяют исследовать, как IoT-сети функционируют в реальных условиях. Мониторинг в реальном времени дает возможность собирать и анализировать данные о задержке, пропускной способности, надежности и других показателях. Например, IoT-аналитические платформы могут предоставлять информацию о джиттере и потерянных пакетах, что помогает оперативно реагировать на возникающие проблемы.

Для более глубокого анализа используются методы симуляции. Такие инструменты, как NS-3 или Cooja, позволяют моделировать различные сценарии работы сети, варьируя условия, например, плотность подключенных устройств или интенсивность трафика. Это особенно важно на этапе проектирования или тестирования новых решений.

Особое внимание уделяется методам оценки по отдельным показателям. Например, для измерения задержки используются ring-тесты, для пропускной способности — инструменты типа iperf, а надежность оценивается через анализ доли потерянных данных. Эти методы помогают точно измерить текущие параметры сети и сравнить их с целевыми значениями.

Инструменты и платформы для оценки качества играют важную роль в упрощении этого процесса. Например, системы мониторинга, такие как Zabbix или Nagios, предоставляют данные о состоянии сети в реальном времени, в то время как аналитические платформы, такие как AWS IoT Analytics, помогают визуализировать и интерпретировать эти данные для долгосрочного планирования.

Оценка качества в сети IoT — это не только процесс анализа текущих показателей, но и способ предотвращения возможных сбоев. Совмещение методов мониторинга, моделирования и анализа метрик обеспечивает надежность и производительность IoT-сетей в самых сложных условиях эксплуатации.

Применение методов обеспечения качества в IoT-сетях демонстрирует их эффективность в различных сценариях. Приведенные ниже примеры иллюстрируют, как эти подходы помогают решать практические задачи, улучшая производительность и надежность IoT-систем.

В проекте управления транспортом в крупном городе задержка данных от датчиков движения, ранее достигавшая 120 мс из-за передачи в облако, была снижена до 30 мс благодаря использованию edge computing. Локальные шлюзы обрабатывали данные в реальном времени, что позволило улучшить синхронизацию светофоров и сократить ошибки маршрутизации на 15%. Этот подход доказал свою эффективность в управлении высоконагруженными сетями.

На заводе, где IoT используется для мониторинга оборудования, перегрузка сети решалась с помощью фильтрации данных на шлюзах. Устройства передавали только аномалии, исключая избыточные данные. Это увеличило пропускную способность на 40% и снизило время простоя оборудования на 25%, обеспечив стабильность работы сети даже при максимальной загрузке.

В проектах мониторинга почвы и климата были внедрены узкополосные протоколы LoRaWAN и гибридные режимы работы датчиков. Устройства активировались только при изменении параметров, что сократило энергопотребление на 60% и увеличило срок службы батарей с 6 до 18 месяцев. Это позволило охватить большие территории с минимальными затратами на обслуживание.

Для защиты данных в системах умных домов использовались шифрование AES-128 и блокчейн-технологии. Эти меры исключили риск подмены устройств и данных, снижая количество кибератак на 80%. Система сохранила высокую производительность, при этом повысив доверие пользователей за счет усиленной защиты конфиденциальности.

Примеры показывают, что современные методы обеспечения качества позволяют IoT-сетям решать сложные задачи, соответствуя высоким требованиям к надежности, безопасности и эффективности. Комплексный подход обеспечивает их устойчивую работу в реальных условиях эксплуатации, от умных городов до сельского хозяйства.

\*\*\*

1. Парамонов А.И., Бушеленков С.Н. Анализ методов повышения эффективности сетей IoT // Информационные технологии и телекоммуникации. – 2022. – Т. 10, № 2. – С. 36–52.
2. Гавриленко А.И., Гавриленко М.А. Проблемы и перспективы реализации IoT в России // Вестник Санкт-Петербургского университета телекоммуникаций. – 2021. – № 4. – С. 45–58.
3. Королев А.В., Беляков П.Н. Подходы к оценке качества обслуживания в сетях IoT // Системы управления и информационные технологии. – 2020. – Т. 80, № 3. – С. 23–30.
4. Иванов В.А., Кузнецов Д.М. Моделирование и тестирование сетей IoT: проблемы и решения // Вопросы радиоэлектроники. – 2021. – № 6. – С. 19–28.

Денеко Д.А., Ротенштрайх Т.В.  
Создание гибкой архитектуры мобильного приложения на Flutter  
с использованием паттерна MVVM

Уральский государственный экономический университет  
(Россия, Екатеринбург)

doi: 10.18411/trnio-12-2024-872

#### Аннотация

Статья описывает использование паттерна MVVM для разработки мобильных приложений на Flutter. Раскрывается, как структурирование кода с помощью MVVM позволяет создавать гибкие и легко поддерживаемые проекты. Рассматриваются роли Model, ViewModel и View, а также приводится пример, демонстрирующий преимущества такой архитектуры для упрощения разработки и тестирования.

**Ключевые слова:** Flutter, MVVM, архитектура приложения, мобильная разработка, Model-View-ViewModel, разработка интерфейсов, структурирование кода.

#### Abstract

The article describes the use of the MVVM pattern for developing mobile applications on Flutter. It reveals how structuring code using MVVM allows you to create flexible and easily maintainable projects. The roles of Model, ViewModel and View are considered, and an example is given demonstrating the advantages of such an architecture for simplifying development and testing.

**Keywords:** Flutter, MVVM, application architecture, mobile development, Model-View-ViewModel, interface development, code structuring.

Архитектура приложения — это не просто организация кода, а ключ к созданию поддерживаемых, масштабируемых и легко изменяемых проектов. Особенно это важно в мобильной разработке, где приложение должно быстро реагировать на изменения и требования рынка. В самом начале может казаться, что для простого приложения, вроде калькулятора, хватит и одного файла. Возможно, на первых этапах это будет выглядеть проще, но со временем даже в таких проектах возникает необходимость в изменениях и дополнениях. Представьте, что вы решили обновить интерфейс калькулятора, добавив новые кнопки или изменив визуальные элементы. Если код не структурирован, то найти нужные части будет сложно, не говоря уже о высоком риске случайно повредить логику.

Для таких ситуаций используются архитектурные паттерны — это подходы к разделению приложения на отдельные слои с чёткой зоной ответственности у каждого. Один из наиболее популярных паттернов в мобильной разработке — Model-View-ViewModel (MVVM), который отлично подходит и для Flutter, благодаря наличию библиотек, поддерживающих управление состоянием (например, provider, riverpod, bloc). MVVM помогает отделить логику работы с данными (Model) от состояния и обработки событий (ViewModel) и отображения данных на экране (View), что значительно упрощает внесение изменений, тестирование и поддержку приложения.

В этом примере мы реализуем простое приложение-калькулятор на Flutter, где каждый компонент имеет своё чёткое назначение, а интерфейс и логика работают независимо друг от друга. Это не только делает проект структурированным, но и облегчает командную работу, поскольку несколько разработчиков могут одновременно работать над разными частями приложения.

#### Компоненты архитектуры MVVM

1. Model — отвечает за хранение данных и выполнение бизнес-логики, например, расчёт значений для калькулятора. Этот слой никак не связан с интерфейсом, что позволяет его легко изменять или тестировать.

2. **ViewModel** — посредник между **Model** и **View**, который управляет состоянием и обработкой пользовательского ввода. Именно **ViewModel** подписывает **View** на изменения, уведомляя её при каждом обновлении данных.
3. **View** — визуальное представление приложения, в котором отображается текущее состояние данных. **View** подписывается на **ViewModel** и обновляется, когда данные изменяются.

Рассмотрим, как реализовать каждый из этих компонентов на Flutter.

**CalculatorModel** — это основной компонент, который отвечает за выполнение вычислений. Он не взаимодействует напрямую с интерфейсом, что делает его независимым и лёгким для тестирования. В этом классе можно хранить данные, вводимые пользователем, и вычислять результаты, которые затем передаются в **ViewModel**,

```
class CalculatorModel {
  double _currentValue = 0;
  double _previousValue = 0;
  String _operation;
  double get currentValue => _currentValue;
  void inputDigit(double digit) {
    _currentValue = _currentValue * 10 + digit;
  }
  void setOperation(String operation) {
    _previousValue = _currentValue;
    _currentValue = 0;
    _operation = operation;
  }
  void calculate() {
    switch (_operation) {
      case '+':
        _currentValue = _previousValue + _currentValue;
        break;
      case '-':
        _currentValue = _previousValue - _currentValue;
        break;
      case '*':
        _currentValue = _previousValue * _currentValue;
        break;
      case '/':
        if (_currentValue != 0) {
          _currentValue = _previousValue / _currentValue;
        }
        break;
    }
  }
  void clear() {
    _currentValue = 0;
    _previousValue = 0;
    _operation = null;
  }
}
```

В этом классе **CalculatorModel** хранит данные для текущего и предыдущего значений, а также выполняет вычисления в методе `calculate()`, вызывая нужную операцию в зависимости от типа операции, которую выбрал пользователь.

**ViewModel:CalculatorViewModel** служит посредником между интерфейсом и моделью, управляя состоянием и передавая данные в интерфейс. В нем также обрабатываются пользовательские действия (например, нажатия на кнопки).

```
import 'package:flutter/material.dart';
class CalculatorViewModel extends ChangeNotifier {
  final CalculatorModel _calculatorModel = CalculatorModel();
  String get display => _calculatorModel.currentValue.toString();
}
```

```

void inputDigit(double digit) {
  _calculatorModel.inputDigit(digit);
  notifyListeners(); // Уведомляем View об изменении
}
void setOperation(String operation) {
  _calculatorModel.setOperation(operation);
  notifyListeners();
}
void calculate() {
  _calculatorModel.calculate();
  notifyListeners();
}
void clear() {
  _calculatorModel.clear();
  notifyListeners();
}
}

```

CalculatorViewModel управляет CalculatorModel и предоставляет интерфейс для взаимодействия с UI. Например, когда пользователь нажимает кнопку с цифрой, метод `inputDigit` обрабатывает это, обновляет состояние и уведомляет View, что нужно перерисовать интерфейс.

View: CalculatorScreen. В представлении CalculatorScreen используется `ChangeNotifierProvider` из пакета `provider`, который позволяет View подписаться на изменения ViewModel. Это основной экран приложения, на котором отображаются кнопки и текущий результат.

```

import 'package:flutter/material.dart';
import 'package:provider/provider.dart';
class CalculatorScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final calculatorViewModel =
    Provider.of<CalculatorViewModel>(context);

    return Scaffold(
      appBar: AppBar(title: Text('Calculator')),
      body: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        children: [
          // Экран для отображения текущего значения
          Expanded(
            child: Container(
              alignment: Alignment.centerRight,
              padding: EdgeInsets.all(24),
              child: Text(
                calculatorViewModel.display,
                style: TextStyle(fontSize: 48, fontWeight:
                FontWeight.bold),
              ),
            ),
          ),
          // Пример кнопок для калькулятора
          Row(
            mainAxisAlignment: MainAxisAlignment.spaceEvenly,
            children: [
              ElevatedButton(
                onPressed: () => calculatorViewModel.inputDigit(1),
                child: Text('1'),
              ),
              ElevatedButton(
                onPressed: () =>
calculatorViewModel.setOperation('+'),

```

```

        child: Text('+'),
      ),
      ElevatedButton(
        onPressed: calculatorViewModel.calculate,
        child: Text('='),
      ),
      ElevatedButton(
        onPressed: calculatorViewModel.clear,
        child: Text('C'),
      ),
    ],
  ),
),
);
}
}

```

В CalculatorScreen мы используем ChangeNotifierProvider для подписки на CalculatorViewModel, что позволяет автоматически обновлять экран при изменении состояния. Пользовательские кнопки вызывают соответствующие методы ViewModel, которые передают данные в Model и выполняют нужные операции.

Преимущество использования архитектурного паттерна MVVM в том, что каждый слой выполняет чётко определённую задачу. CalculatorModel управляет бизнес-логикой, CalculatorViewModel управляет состоянием и передаёт данные, а CalculatorScreen отвечает за отображение и взаимодействие с пользователем. Такая структура позволяет легко изменять отдельные компоненты без риска повредить остальной код, что особенно полезно для больших приложений с множеством функций и сложной логикой.

С этой архитектурой калькулятор можно легко модифицировать или расширить. Если потребуется изменить способ работы с данными, нужно просто внести изменения в CalculatorModel. Если нужно обновить интерфейс — можно работать только с CalculatorScreen. Разделение приложения на компоненты делает его более гибким и удобным для команды разработчиков, позволяя каждому работать над своей частью приложения, не затрагивая остальные.

\*\*\*

1. Приемы объектно-ориентированного проектирования. Паттерны проектирования. Учебное пособие / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – Москва : ДМК Пресс, 2007. – 368 с. – ISBN 5-93700-023-4. – EDN RAYGPH.
2. Indrawan, D. Analysis of the implementation of MVVM architecture pattern on performance of IOS mobile-based applications / D. Indrawan, D. S. Kusumo, Sh. Yu. Puspitasari // JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika). – 2023. – Vol. 8, No. 1. – P. 59-65. – DOI 10.29100/jipi.v8i1.3293. – EDN HUEDTO.
3. Хлопин, К. В. Современные frontend-приложения на основе MVVM-фреймворков / К. В. Хлопин, Д. А. Ломаш // Цифровые инфокоммуникационные технологии : сборник научных трудов, Ростов-на-Дону, 07 сентября 2021 года. – Ростов-на-Дону: Ростовский государственный университет путей сообщения, 2021. – С. 104-107. – EDN NUULEK.

**Дмитриев А.В., Алексеев Д.И.**

**Как блокчейн меняет финансовую индустрию**

*Уфимский университет науки и технологий  
(Россия, Уфа)*

doi: 10.18411/trnio-12-2024-873

#### **Аннотация**

Статья «Как блокчейн меняет финансовую индустрию» исследует влияние технологий блокчейн на традиционные финансовые операции и процессы. В последние годы блокчейн стал важным инструментом для обеспечения прозрачности, безопасности

и эффективности в финансовом секторе. Анализируются ключевые аспекты внедрения блокчейн-технологий, включая смарт-контракты, децентрализованные финансы (DeFi) и цифровые валюты, которые меняют роль финансовых посредников и транзакционных механизмов. Также рассматриваются вызовы и риски, связанные с использованием блокчейна, такие как юридическая регуляция, безопасность данных и вопросы конфиденциальности.

**Ключевые слова:** блокчейн, финансовая индустрия, смарт-контракты, децентрализованные финансы (DeFi), цифровые валюты, прозрачность, эффективность.

### Abstract

The article “How Blockchain is Changing the Financial Industry” explores the impact of blockchain technologies on traditional financial transactions and processes. In recent years, blockchain has become an important tool for transparency, security and efficiency in the financial sector. It analyzes key aspects of blockchain technology adoption, including smart contracts, decentralized finance (DeFi) and digital currencies, which are changing the role of financial intermediaries and transactional mechanisms. Challenges and risks associated with the use of blockchain such as legal regulation, data security and privacy issues are also discussed.

**Keywords:** blockchain, financial industry, smart contracts, decentralized finance (DeFi), digital currencies, transparency, efficiency.

В последние годы блокчейн-технологии приобрели огромную популярность, прочно войдя в обиход различных секторов экономики и бизнеса. Эта технология, изначально разработанная для поддержки виртуальной валюты Bitcoin, сегодня используется в самых различных областях – от управления цепочками поставок до медицины. Однако, пожалуй, наиболее значительное влияние блокчейн оказал на финансовую индустрию. Традиционные финансовые институты, включая банки, страховые компании и другие организации, находят перед собой как уникальные возможности, так и серьезные вызовы в свете внедрения блокчейн-технологий.

Актуальность изучения блокчейна в контексте финансовых услуг не подлежит сомнению. С каждым днем появляется все больше оснований утверждать, что данный технологический подход может радикально изменить классические финансовые модели, предложив более эффективные, безопасные и открытые альтернативы. Прозрачность транзакций, высокая степень защиты данных и возможность автоматизации процессов с помощью смарт-контрактов делают блокчейн привлекательным инструментом для оптимизации финансовых операций.

Целью данной статьи является глубокий анализ блокчейн-технологий и их влияния на финансовую индустрию. Мы стремимся осветить основные принципы работы блокчейна, его ключевые компоненты, а также различные типы блокчейнов. Помимо этого, мы рассмотрим, как применение этой технологии может изменить традиционные финансы, обеспечив новые возможности для бизнеса и потребителей.

Блокчейн — это распределенная база данных, в которой информация хранится в виде последовательности блоков, связанных друг с другом и защищенных криптографией. Каждый блок содержит набор транзакций, временную метку и ссылку на предыдущий блок, что делает невозможным его изменение без риска потери целостности всей цепочки. Это свойство делает блокчейн прозрачным и защищенным от манипуляций[1].

Ключевыми компонентами блокчейна являются узлы (ноды), которые представляют собой устройства, подключенные к сети и участвующие в обработке и хранилище данных. Также важны протоколы консенсуса, которые обеспечивают согласие участников сети по поводу того, какие транзакции были успешно добавлены в блокчейн. Наиболее распространенные протоколы включают Proof of Work (доказательство работы) и Proof of Stake (доказательство доли).

Принципы работы блокчейна основаны на децентрализации и прозрачности. Вместо того чтобы хранить данные в одном центральном месте, информация распределена среди всех участников сети. Каждая транзакция проверяется узлами сети, что минимизирует риск фальсификаций и мошенничества.

Структура данных в блокчейне организована так, что каждый новый блок содержит хэш предыдущего, что создает цепочку, защищенную от изменения. Если кто-то попытается изменить информацию в одном из блоков, хэш этого блока изменится, и цепочка станет недействительной. Это делает блокчейн высокозащищенной системой для ведения записей [2].

Существует несколько типов блокчейнов, каждый из которых служит своим целям и задачам.

Публичные блокчейны – это открытые сети, в которых любой может участвовать. Примеры включают Bitcoin и Ethereum. Они обеспечивают полный уровень децентрализации и прозрачности, так как все транзакции доступны для проверки.

Частные блокчейны – это закрытые сети, доступные только авторизованным пользователям. Они чаще используются организациями для внутренних процессов и могут предложить более высокую скорость обработки транзакций, чем публичные сети.

Консорциумные блокчейны – это гибридные решения, где несколько организаций совместно управляют блокчейном и могут устанавливать свои правила. Они идеально подходят для тех случаев, когда требуется определенный уровень децентрализации, но при этом необходимо контролировать доступ к информации.

Одной из самых значительных областей, где блокчейн демонстрирует свое влияние, являются платежные системы и переводы. Традиционные банковские операции часто связаны с длительными процессами отправки и получения средств, особенно когда дело касается международных переводов. Блокчейн позволяет производить эти транзакции практически мгновенно, так как информация об операциях записывается и подтверждается децентрализованными узлами сети. Это сокращает время обработки и позволяет избежать задержек, связанных с банковскими святками или праздничными днями.

Блокчейн устраняет необходимость в посредниках, таких как банки и платежные системы. Это особенно полезно в случае международных переводов, которые часто требуют взаимодействия между различными финансовыми учреждениями, что ведет к задержкам и сложностям с конверсией валют. В результате, пользователи могут отправлять и получать средства напрямую, сокращая количество этапов в процессе и тем самым упрощая его.

Снижение комиссии за транзакции — еще один аспект, по которому блокчейн достигает значительных результатов. Традиционные системы часто взимают высокие комиссии за каждую транзакцию, включая скрытые издержки на конвертацию и межбанковские сборы. В отличие от этого, блокчейн использует модели микроплатежей, что позволяет значительно сократить затраты на обработку транзакций. Это особенно важно для пользователей, которые переводят небольшие суммы, так как высокие комиссии могут значительно снизить выгоду от транзакций [3].

Блокчейн также приводит к революционным изменениям в области учета и хранения данных. Все данные о транзакциях записываются в защищенную цепочку блоков, которая обеспечивает неизменность и прозрачность информации.

Доступность и безопасность информации являются ключевыми преимуществами использования блокчейна в финансовых операциях. Данные хранятся на сети распределенных узлов, что делает их менее уязвимыми для кибератак. Кроме того, все участники сети имеют доступ к общей информации, что позволяет минимизировать возможность мошенничества и повысить доверие в транзакциях.

Современные финансовые учреждения сталкиваются с вызовами, связанными с отчетностью и аудитом данных. Блокчейн предлагает инструменты для автоматизации

этих процессов, что позволяет значительно упростить и ускорить аудиторские процедуры. Прозрачность и возможность отслеживания каждой транзакции делают процесс отчета более надежным и менее подверженным рискам [4].

Не менее важным аспектом влияния блокчейна на финансовую индустрию являются кредитование и займы. В то время как традиционная банковская система требует от заемщиков предоставления множества документов и достаточного кредитного рейтинга, блокчейн вводит новые подходы.

Децентрализованные кредитные платформы стали активно развиваться благодаря технологии блокчейн. Эти платформы позволяют пользователям давать и получать займы без участия традиционных банков. Используя смарт-контракты, платформа автоматически подписывает и исполняет условия соглашений, минимизируя риски для обеих сторон. Это привлекает большое количество пользователей, которые могут быть исключены из традиционной системы кредитования.

В результате развития децентрализованных решений роль традиционных банков меняется. Если ранее они выступали посредниками в финансовых операциях, то с внедрением блокчейна эти функции перераспределяются. Блокчейн предоставляет возможность участникам рынка взаимодействовать напрямую, что угрожает существующим бизнес-моделям банков.

С учетом всего вышесказанного, будущее блокчейна в финансовой индустрии представляется многообещающим. Прогнозы указывают на то, что с каждым годом будет возрастать число организаций, использующих эту технологию для оптимизации своих процессов.

Эксперты прогнозируют дальнейшее расширение сфер применения блокчейна, включая не только финансовые услуги, но и другие отрасли, такие как здравоохранение и логистика. Модель «умных контрактов», которые автоматически исполняются при выполнении определенных условий, получит широкую популярность, что облегчит взаимодействие между участниками различных сделок.

Инновации, которые принесет блокчейн, позволят создавать новые решения для работы с данными и финансами. Такие технологии, как тенизация активов, могут изменить способ, которым мы владеем и передаем ценности. Это откроет новые возможности для инвестирования, а также создания новых рынков.

На глобальном уровне блокчейн способен трансформировать такие аспекты, как финансовая инклюзия, снижая барьеры для доступа к финансовым услугам для людей в развивающихся странах. Локальные экономики, в свою очередь, могут выигрывать от большей прозрачности и эффективности в распределении ресурсов [5].

Таким образом, блокчейн-технология имеет значительное влияние на традиционные финансовые операции, преобразовывая платежные системы, процессы учета, кредитование и условия работы банков. Важно отметить, что, несмотря на все преимущества, внедрение блокчейна также сопряжено с вызовами, включая необходимость нормативного регулирования и обеспечения безопасности.

\*\*\*

1. Роббек А. Е. BITCOIN как явление в мировой экономике. Научная статья. Журнал «Вестник СФВУ». 2020. С. 5.
2. Консалтинговая компания «Deloitte». Международное исследование, посвященное внедрению блокчейна, за 2022 год. Электронный ресурс: <https://www2.deloitte.com/ru/ru/pages/energy-and-resources/articles/gx-innovation-blockchain-survey.html> (дата обращения: 27.11.2024).
3. Пехтерева Е. А. Инновации в Финансовой сфере и практика их применения: технология блокчейн и крипто-валюта в России. Научная статья. 2018. С. 19.
4. Ковальчук. А. В., Сайбель Н. Ю. Блокчейн-технологии в финансовом секторе экономики: преимущества и проблемы использования. Научная статья. Журнал «Концепт». 2018. С. 6.
5. Цветкова Л. А. Перспективы развития технологии блокчейн в России: конкурентные преимущества и барьеры. Журнал «Экономика науки». 2017. С. 275-296

**Игнатова Я.А.**

**Применение машинного обучения для обнаружения и предотвращения угроз в приложениях**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-874

**Аннотация**

Статья посвящена применению машинного обучения для обнаружения и предотвращения угроз в приложениях. Рассмотрены ограничения традиционных методов безопасности и преимущества ML, включая анализ аномалий, классификацию угроз и прогнозирование атак. Приведены примеры использования контролируемого, неконтролируемого и глубокого обучения в системах кибербезопасности.

**Ключевые слова:** машинное обучение, кибербезопасность, обнаружение угроз, контролируемое обучение, глубокое обучение, системы предотвращения вторжений (IDS/IPS).

**Abstract**

The article is devoted to the application of machine learning to detect and prevent threats in applications. The limitations of traditional security methods and the advantages of ML, including anomaly analysis, threat classification and attack prediction, are considered. Examples of the use of supervised, unsupervised and deep learning in cybersecurity systems are given.

**Keywords:** machine learning, cybersecurity, threat detection, supervised learning, deep learning, intrusion prevention systems (IDS/IPS).

Обеспечение безопасности приложений стало ключевой задачей в условиях роста числа кибератак. Традиционные методы защиты, основанные на сигнатурах и правилах, не всегда эффективны против современных угроз, таких как zero-day атаки или сложные целевые атаки (APT). В этой связи машинное обучение становится перспективным инструментом для автоматизации процессов обнаружения угроз и повышения их точности.

Традиционные методы обнаружения угроз, такие как сигнатурный анализ и эвристический подход, на протяжении многих лет являлись основой систем безопасности. Однако в современных условиях они сталкиваются с рядом фундаментальных ограничений, которые снижают их эффективность в борьбе с новыми и сложными атаками.

Сигнатурные методы основываются на сравнении входящих данных с известными шаблонами угроз. Этот подход обеспечивает высокую точность при работе с уже известными угрозами, но его слабое место заключается в невозможности обнаружения ранее неизвестных атак, включая так называемые "атаки нулевого дня". Более того, поддержание актуальности сигнатурной базы требует постоянного обновления, что создает значительные временные задержки в реакции на новые угрозы. Современные атаки часто характеризуются высокой степенью уникальности, что делает традиционные базы данных сигнатур недостаточно эффективными.

Эвристические методы, использующие наборы правил и шаблонов для выявления потенциальных угроз, предоставляют более широкие возможности, однако их применение связано с высокой частотой ложных срабатываний. Например, в случае сложных систем, таких как корпоративные сети, где характер поведения пользователей и приложений может быть разнообразным, эвристика часто воспринимает легитимные действия как потенциальную угрозу. Это, в свою очередь, увеличивает нагрузку на системы реагирования и персонал, ответственный за анализ инцидентов.

Еще одной проблемой является ограниченность традиционных методов в анализе больших объемов данных, генерируемых современными приложениями и сетями. Рост числа устройств и усложнение инфраструктуры приводят к экспоненциальному увеличению объема логов и событий безопасности. Традиционные подходы не обладают достаточной вычислительной мощностью и аналитическими возможностями для обработки таких данных в реальном времени. Это создает временные задержки и снижает оперативность в реагировании на угрозы.

Проблемы, присущие традиционным методам обнаружения угроз, обусловили необходимость разработки более адаптивных и интеллектуальных решений. Одним из таких подходов является использование машинного обучения (ML), которое предоставляет возможность автоматической обработки больших объемов данных и выявления сложных закономерностей. Машинное обучение эффективно справляется с задачами анализа угроз благодаря своей способности обучаться на основе данных и адаптироваться к новым сценариям атак.

Машинное обучение основывается на создании моделей, которые изучают зависимости в данных и используют их для принятия решений. В контексте кибербезопасности ML может применяться для обнаружения аномалий, классификации угроз, прогнозирования событий, что отображено на рисунке 1.

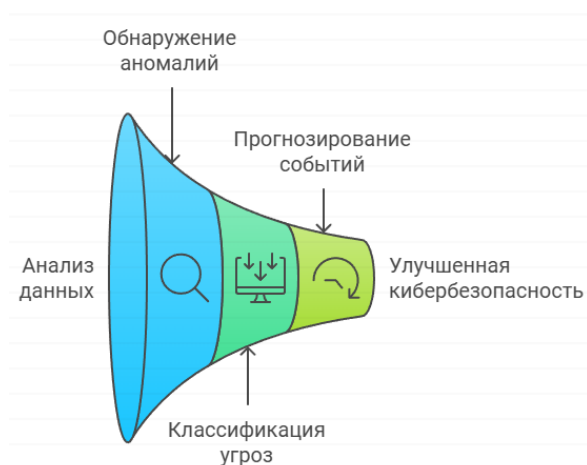


Рисунок 1. Машинное обучение в кибербезопасности.

Для реализации этих задач применяются как контролируемые, так и неконтролируемые методы обучения.

Контролируемое обучение использует размеченные данные для создания моделей, способных различать нормальные и подозрительные действия. Например, алгоритм случайного леса (Random Forest) может обучаться на исторических данных о сетевом трафике, где каждая запись помечена как "безопасная" или "вредоносная". После обучения модель способна классифицировать новые запросы, определяя их статус на основе изученных закономерностей.

Примером применения контролируемого обучения является фильтрация фишинговых писем. Используя алгоритмы, такие как логистическая регрессия, модель анализирует заголовки, содержимое и метаданные писем, чтобы определить вероятность того, что письмо является фишинговым. Этот подход уже используется в системах защиты электронной почты, таких как Google Workspace.

Неконтролируемое обучение применяется в ситуациях, когда данные не имеют заранее определенной разметки. Основная задача таких алгоритмов — выявление аномалий или кластеризация данных. Например, алгоритм K-means может быть использован для сегментации сетевого трафика, выделяя кластеры, которые соответствуют нормальному поведению, и идентифицируя те, что отклоняются от нормы.

В реальных системах неконтролируемое обучение часто используется для анализа поведения пользователей (User Behavior Analytics, UBA). Например, если пользователь

внезапно начинает скачивать большие объемы данных в нехарактерное для него время, модель, обученная на его нормальном поведении, сможет зафиксировать эту аномалию и выдать предупреждение.

Глубокое обучение (Deep Learning), как более сложная форма машинного обучения, активно применяется для анализа сложных и многомерных данных. Например, рекуррентные нейронные сети (RNN) используются для анализа последовательностей событий, таких как системные вызовы или действия пользователей. Это позволяет эффективно обнаруживать продвинутое атаки, такие как АРТ (Advanced Persistent Threats).

Другим примером является использование сверточных нейронных сетей (CNN) для анализа содержимого пакетов трафика, что позволяет обнаруживать атаки на уровне приложений, такие как внедрение вредоносного кода.

Основное преимущество машинного обучения заключается в его способности выявлять новые угрозы, которые не могут быть обнаружены традиционными методами. Алгоритмы ML анализируют данные в реальном времени, минимизируя задержки в реакции на инциденты. Кроме того, их способность адаптироваться к изменениям делает их особенно эффективными в условиях постоянно меняющегося ландшафта угроз.

Например, при использовании ML в системах IDS/IPS (Intrusion Detection/Prevention Systems), модель может не только обнаружить подозрительный трафик, но и автоматически обновить свои параметры для учета новой информации о поведении атакующего.

Таким образом, машинное обучение представляет собой мощный инструмент для обнаружения и предотвращения угроз, особенно в условиях, когда традиционные методы становятся недостаточными. Его способность выявлять аномалии, классифицировать угрозы и адаптироваться к новым сценариям делает ML ключевым компонентом современных систем кибербезопасности. Однако для его эффективного использования требуется не только качественная техническая реализация, но и учет возможных ограничений, чтобы минимизировать риски и максимизировать преимущества.

\*\*\*

1. Красов А. В., Сахаров Д. В., Тасюк А. А. Проектирование системы обнаружения вторжений для информационной сети с использованием больших данных // Научные исследования в космических исследованиях Земли. – 2020. – Т. 12. – № 1. – С. 70-76.
2. Леснова Е. М., Пестов И. Е. Разработка метода обнаружения и коррекции ошибок для распределенной информационной сети на основе больших данных // Региональная информатика и информационная безопасность. – 2018. – С. 236-240.
3. Калинин М. О., Штеренберг С. И. Анализ информационной безопасности предприятия на основе мониторинга информационных ресурсов с использованием машинного обучения // Интеллектуальные технологии на транспорте. – 2018. – № 3 (15). – С. 47-54.
4. Лаврова Д. С. и др. Предупреждение Dos-атак путем прогнозирования значений корреляционных параметров сетевого трафика // Проблемы информационной безопасности. Компьютерные системы. – 2018. – № 3. – С. 70-77.

**Игнатова Я.А.**

**Сетевые технологии связи**

*Санкт-Петербургский государственный университет телекоммуникаций*

*им. проф. М.А. Бонч-Бруевича*

*(Россия, Санкт-Петербург)*

*doi: 10.18411/trnio-12-2024-875*

#### **Аннотация**

Статья посвящена обзору современных сетевых технологий беспроводной связи, включая ZigBee, Bluetooth и Wi-Fi. Рассматриваются их особенности, такие как дальность действия, скорость передачи данных, энергопотребление и безопасность. Особое внимание уделено применению технологий в системах IoT и умных домах. Проведено

сравнение технологий для различных сценариев использования, что позволяет определить оптимальные решения в зависимости от задач и условий эксплуатации.

**Ключевые слова:** сетевые технологии, ZigBee, беспроводная связь, Bluetooth, Wi-Fi, IoT (Интернет вещей).

### Abstract

The article is devoted to an overview of modern wireless network technologies, including ZigBee, Bluetooth and Wi-Fi. Their features such as range, data transfer rate, power consumption and security are considered. Special attention is paid to the application of technologies in IoT systems and smart homes. A comparison of technologies for different use cases has been carried out, which makes it possible to determine the optimal solutions depending on the tasks and operating conditions.

**Keywords:** network technologies, ZigBee, wireless communication, Bluetooth, Wi-Fi, IoT (Internet of Things).

Технологии беспроводной связи можно разделить на два типа: с лицензией и без лицензии, а также по диапазону действия и скорости передачи данных. На рисунке ниже показаны различные технологии связи, классифицированные по дальности передачи и скорости данных, включая BLE, Zigbee, Z-Wave для небольших помещений и технологии LPWAN, такие как LoRa и Sigfox, для связи на больших расстояниях. Технологии беспроводной связи передают данные через модулированные электромагнитные волны, которые затем демодулируются на приемной стороне. Однако беспроводные сигналы подвержены помехам и затуханию, что ограничивает их эффективность на больших площадях зданий.

На рисунке 1 представлено сравнение технологий по дальности и скорости передачи данных, что помогает выбрать подходящее решение для различных сценариев.

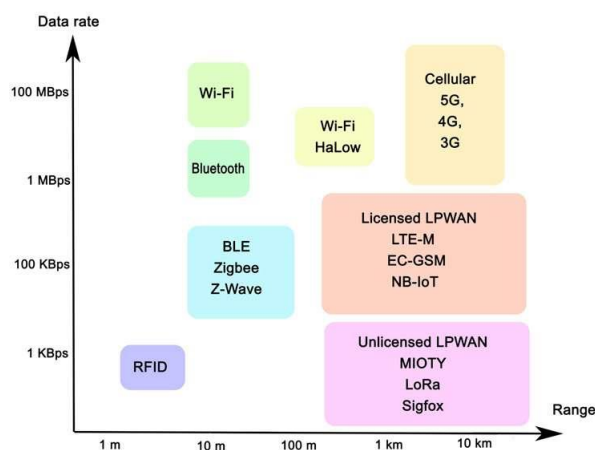


Рисунок 1. Сравнение технологий связи по дальности и скорости передачи данных.

Zigbee — это спецификация, основанная на IEEE 802.15.4, разработанная для низкоэнергетичных сетей с малой пропускной способностью и радиусом действия до 100 метров. Работающий на частоте 2,4 ГГц, Zigbee обеспечивает скорость передачи данных до 250 кбит/с и требует минимального энергопотребления, что делает его подходящим для применения в устройствах с ограниченными ресурсами, таких как микроконтроллеры и датчики.

Первоначальная спецификация Zigbee была утверждена как стандарт IEEE в 2003 году, а уже в 2006 году появились первые коммерчески доступные OEM-модули. Этот стандарт, ориентированный на телеметрию и беспроводное считывание данных с

приборов учета, создан для связи между различными типами датчиков и устройств мониторинга. Zigbee использует топологию самонастраивающейся и самовосстанавливающейся сетки, что упрощает его установку и обслуживание, а также позволяет легко масштабировать сеть до тысяч узлов.

Как протокол пакетной передачи данных, Zigbee устойчив к ошибкам и защищен от несанкционированного доступа, что делает его надежным выбором для гаджетов и сенсоров, работающих в сложных условиях. Благодаря открытости стандарта, на рынке доступно множество устройств, поддерживающих Zigbee, что способствует его популярности и широкому применению.

ZigBee находит основное применение в устройствах с низким уровнем энергопотребления, работающих на батареях, таких как переключатели для управления бытовой техникой, датчики движения, температуры и влажности. Благодаря поддержке низкой скорости передачи данных, ZigBee отлично подходит для приложений, где требуется периодическая передача небольших объемов информации.

Эта технология широко используется в умных домах, позволяя различным устройствам взаимодействовать друг с другом через единую сеть. На рисунке 2 показано, как ZigBee объединяет различные элементы умного дома, включая освещение, термостаты, бытовую технику и другие устройства, создавая единую сеть, которую можно легко контролировать и настраивать.

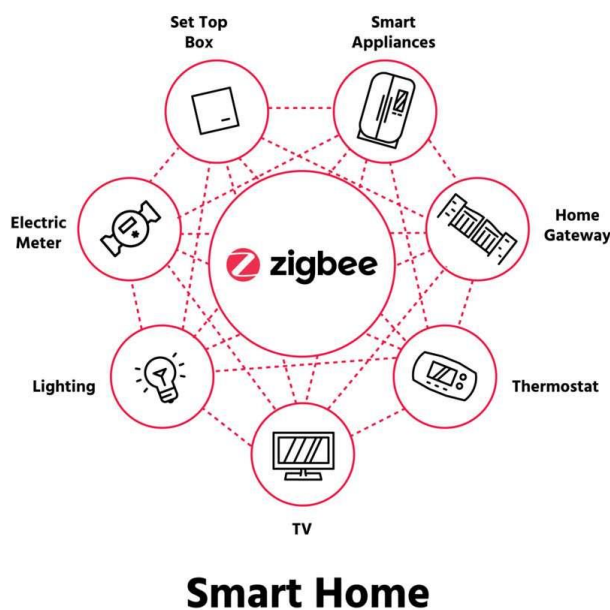


Рисунок 2. Применение ZigBee в умном доме.

Bluetooth — это технология беспроводной связи малого радиуса действия, предназначенная для обмена данными между стационарными и мобильными устройствами на коротких расстояниях, что позволяет строить персональные сети. Работая на частоте 2,4 ГГц, Bluetooth поддерживает дальность передачи от 50 до 150 метров. Современный стандарт Bluetooth 5 предлагает четыре различных скорости передачи данных, от 125 кбит/с до 2 Мбит/с, что позволяет выбирать оптимальную скорость и радиус действия для разных приложений.

Bluetooth включает три класса мощности, определяющих радиус действия: 100 метров для первого класса, 10 метров для второго и 1 метр для третьего. Наиболее популярным является второй класс, так как он позволяет подключать устройства, находящиеся в разных комнатах или даже на разных этажах.

Передача данных в Bluetooth происходит через пакеты по одному из 79 каналов (в Bluetooth 1.0) или 40 каналов (в Bluetooth 4.0 и выше) с пропускной способностью до 2 МГц, что обеспечивает более высокую скорость передачи и улучшенную защиту данных в последних версиях. Важно, что новые стандарты Bluetooth сохраняют обратную

совместимость, обеспечивая гибкость подключения между устройствами разных поколений.

Bluetooth активно используется в умных бытовых приборах, таких как шлюзы для умного дома, смартфоны, фитнес-браслеты, а также для передачи аудио и управления умными кондиционерами. Благодаря этой технологии можно подключать к сети множество электронных устройств — от телефонов и ноутбуков до гарнитур и принтеров, что делает Bluetooth универсальным инструментом для беспроводной связи в рамках умного дома и персональных сетей.

Wi-Fi — это семейство беспроводных протоколов на основе стандартов IEEE 802.11, используемое для локальных сетей и доступа в Интернет, позволяя устройствам в непосредственной близости обмениваться данными по радиоканалу. Wi-Fi работает на определенных частотах, обычно 2,4 ГГц и 5 ГГц, что позволяет распределить нагрузку между устройствами и снизить вероятность прерываний. Последняя версия протокола, Wi-Fi 6, обеспечивает максимальную пропускную способность до 9,6 Гбит/с по нескольким каналам.

Радиус действия стандартного соединения Wi-Fi может достигать 100 метров, однако на практике он часто ограничен 10-35 метрами, в зависимости от мощности антенны и условий окружающей среды. При приближении к источнику сигнала скорость передачи данных увеличивается, а при удалении — снижается. Диапазон и скорость Wi-Fi также зависят от того, находится ли сеть в помещении или на открытом пространстве.

Wi-Fi востребован для передачи больших объемов данных, что делает его идеальным для устройств умного дома, таких как видеокамеры и дверные звонки, которым требуется надежное подключение с высокой пропускной способностью. Для создания Wi-Fi сети необходимо устройство, способное передавать беспроводной сигнал, например маршрутизатор, подключенный к интернету. В пределах радиуса действия Wi-Fi соединяет устройства с Интернетом и позволяет устройствам, таким как компьютеры и смартфоны, обмениваться соединением, расширяя возможности беспроводной сети в доме или офисе.

Для выбора подходящей технологии связи в системах IoT важно учитывать параметры, такие как диапазон действия, частота, сложность установки и энергопотребление. В таблице 1 ниже представлено сравнение трех популярных технологий беспроводной связи: ZigBee, Bluetooth и Wi-Fi.

Таблица 1

Сравнение Bluetooth, ZigBee и Wi-Fi.

<i>Параметр</i>	<i>ZigBee</i>	<i>Bluetooth</i>	<i>Wi-Fi</i>
<i>Стандарт</i>	802.15.4	802.15.1	802.11ah
<i>Диапазон</i>	10–100 метров	10–100 метров	50–100 метров
<i>Рабочая частота</i>	868 МГц (Европа), 900–928 МГц (Северная Америка), 2.4 ГГц (по всему миру)	2.4 ГГц	2.4 и 5 ГГц
<i>Сетевая топология</i>	Одноранговая, звездообразная или сетевая	Ad-hoc, очень маленькие сети	Необходим специальный узел, который будет централизованно управлять подключениями устройств.
<i>Сложность</i>	Низкий	Высокий	Высокий
<i>Потребляемая мощность</i>	Очень низкий	Средний	Высокий
<i>Безопасность</i>	128-битное AES плюс безопасность на уровне приложений	64- и 128-битное шифрование	WPA2-PSK (AES)
<i>Типичные варианты использования</i>	Промышленный контроль, интеллектуальные сенсорные сети, автоматизация зданий, домашний контроль и автоматизация	Беспроводная связь между устройствами BLE	Подключение к локальной сети и доступ в Интернет

Таблица показывает, что каждая из технологий имеет свои сильные стороны и подходит для определенных сценариев. ZigBee оптимален для низкоэнергетичных приложений с большим количеством устройств, таких как системы автоматизации и мониторинга. Bluetooth хорошо подходит для обмена данными на коротких расстояниях между персональными устройствами, тогда как Wi-Fi обеспечивает высокую скорость и подходит для приложений с требованием к широкополосному доступу в Интернет. Выбор технологии зависит от конкретных потребностей системы, таких как радиус действия, энергопотребление и требования к безопасности.

\*\*\*

1. Леснова Е. М., Пестов И. Е. Разработка метода обнаружения и коррекции ошибок для распределенной информационной сети на основе больших данных // Региональная информатика и информационная безопасность. – 2018. – С. 236-240.
2. Пестов И. Е. Методика разработки управляющего воздействия на инстансы облачной инфраструктуры // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. – 2020. – № 4. – С. 72-76.
3. Горбань С. А., Красов А. В., Цветков А. Ю. Оценка эффективности механизмов контроля правами доступа в ОС Linux // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). – 2023. – С. 345-348.
4. Волкогонов В. Н. и др. Применение физически неклонированных функций для выполнения аутентификации в среде интернета вещей // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2021. – С. 409-414.

**Игнатова Я.А.**

### **Тестирование на проникновение в системах IoT**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-876

#### **Аннотация**

В статье анализируется тестирование на проникновение в системах Интернета вещей (IoT), включая основные особенности IoT-устройств, уязвимости сетевых протоколов и облачных сервисов, а также аппаратные риски. Приведен пример пентеста системы управления освещением, демонстрирующий ключевые подходы и методы, включая анализ прошивки, тестирование API и сетевых протоколов. Рассматриваются практические выводы и рекомендации по устранению выявленных проблем.

**Ключевые слова:** тестирование на проникновение, IoT, безопасность сетевых протоколов, уязвимости облачных сервисов, аппаратная безопасность, пентест IoT.

#### **Abstract**

The article analyzes penetration testing in Internet of Things (IoT) systems, including the main features of IoT devices, vulnerabilities of network protocols and cloud services, as well as hardware risks. An example of a lighting control system pentest is provided, demonstrating key approaches and methods, including firmware analysis, API testing and network protocols. Practical conclusions and recommendations on how to eliminate the identified problems are considered.

**Keywords:** penetration testing, IoT, network protocol security, cloud service vulnerabilities, hardware security, IoT pentest.

В эпоху стремительного развития Интернета вещей (IoT) устройства становятся неотъемлемой частью повседневной жизни, обеспечивая удобство и автоматизацию множества процессов. Однако широкое распространение IoT сопровождается ростом киберугроз, направленных на эксплуатацию уязвимостей этих устройств. Тестирование на

проникновение (пентест) в системах IoT приобретает особую значимость, позволяя выявлять и устранять потенциальные слабые места до того, как ими воспользуются злоумышленники.

Тестирование на проникновение в системах IoT обладает рядом уникальных особенностей, связанных с архитектурой и функциональными характеристиками таких устройств. В отличие от традиционных IT-систем, устройства IoT представляют собой сочетание аппаратных и программных компонентов, что усложняет анализ их безопасности. Ключевые особенности, которые необходимо учитывать при пентесте IoT, включают:

**Гетерогенность IoT-устройств.** Системы IoT часто включают устройства от разных производителей с уникальными аппаратными и программными архитектурами. Это приводит к необходимости адаптации подходов тестирования для работы с широким спектром платформ и протоколов, таких как MQTT, CoAP, ZigBee и Bluetooth. Например, уязвимости, характерные для одного протокола, могут отсутствовать в другом, что требует глубокого понимания особенностей каждого из них.

**Ограниченные вычислительные ресурсы.** Многие IoT-устройства обладают ограниченными вычислительными мощностями и энергопотреблением. Эти ограничения могут приводить к компромиссам в реализации безопасности, например, упрощению алгоритмов шифрования или отсутствию механизмов аутентификации. Задача пентестера — проверить, насколько такие компромиссы влияют на общую защищенность системы.

**Физический доступ к устройствам.** В отличие от традиционных систем, IoT-устройства могут быть расположены в легко доступных местах, что увеличивает риск физической компрометации. Пентест должен включать анализ уязвимостей, связанных с возможностью физического вмешательства, таких как извлечение данных из флеш-памяти или обход аппаратных механизмов защиты.

**Взаимодействие с облачными сервисами.** Многие IoT-устройства взаимодействуют с облачными платформами для передачи данных и управления. Это открывает новые векторы атак, такие как компрометация API, слабая аутентификация или незащищенная передача данных. Тестирование должно учитывать эти аспекты и включать проверку безопасности всех компонентов экосистемы IoT.

Тестирование IoT-устройств требует комплексного подхода, охватывающего как программные, так и аппаратные компоненты системы. На рисунке 1 представлена структура ключевых направлений пентестинга IoT, включающая анализ прошивки, проверку сетевых протоколов, тестирование облачных сервисов и API, аппаратный анализ.

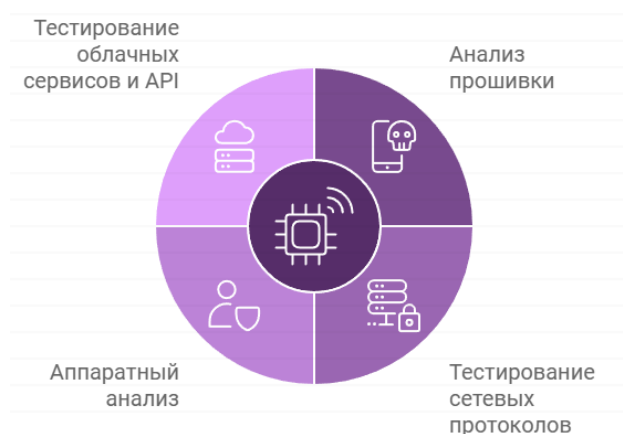


Рисунок 1. Комплексное тестирование на проникновение IoT.

Анализ прошивки устройства позволяет выявить встроенные уязвимости, такие как использование слабых паролей, наличие отладочных функций или включение

уязвимого кода. Этот метод начинается с извлечения прошивки, её декомпиляции и изучения её содержимого на наличие проблем. Инструменты, такие как Binwalk и Firmware Mod Kit, являются стандартами в этой области, предоставляя возможность автоматизированного анализа.

Сетевые протоколы, такие как MQTT, CoAP и ZigBee, являются критическими компонентами IoT-устройств. Тестирование сетевых протоколов включает анализ их защищенности от атак, таких как Man-in-the-Middle, перехват данных и атаки на аутентификацию. Использование инструментов, таких как Wireshark и Burp Suite, позволяет идентифицировать слабости в реализации протоколов и защитных механизмов.

Многие IoT-устройства интегрированы с облачными платформами, через которые они передают данные и управляются пользователями. Тестирование облачных сервисов включает проверку безопасности API, аутентификации и передачи данных. Инструменты, такие как Postman и OWASP ZAP, позволяют выявлять уязвимости в реализации API и защищенности облачных компонентов.

Аппаратная безопасность IoT-устройств играет ключевую роль, так как физический доступ к устройству часто возможен для злоумышленника. Аппаратный анализ включает тестирование устойчивости к физическим атакам, извлечение данных из памяти и проверку устойчивости к обходу систем защиты. JTAG и OpenOCD являются популярными инструментами для выполнения подобных задач.

В качестве примера тестирования на проникновение рассмотрим исследование безопасности интеллектуальной системы управления освещением, использующей протокол ZigBee для связи между устройствами и облачной платформой для удаленного управления. Целью тестирования было выявление уязвимостей в передаче данных, архитектуре сети и защите устройства от атак.

#### **Анализ сетевой безопасности**

Для проверки сетевой защищенности было развернуто окружение, включающее шлюз ZigBee, интеллектуальные лампы и облачный сервер управления. На начальном этапе была проведена пассивная фиксация трафика с использованием SDR (Software Defined Radio). Анализ перехваченных пакетов выявил, что сеть использует статический сетевой ключ для шифрования, который передается в открытом виде во время процедуры сопряжения устройств.

Для проверки возможности атаки был проведен повторный ввод устройства в сеть с фиксацией процесса сопряжения. Получив ключ шифрования, злоумышленник может расшифровать передаваемые данные, включая команды управления, а также отправлять собственные команды на устройство.

#### **Уязвимости облачной платформы**

Дополнительно была проведена проверка взаимодействия между мобильным приложением и облачным сервером. Для этого использовался инструмент Burp Suite, позволяющий перехватывать и анализировать HTTP-запросы. Анализ показал, что система аутентификации API была реализована без проверки привязки токена к устройству. Это позволило отправить поддельный запрос с корректным токеном аутентификации, имитируя действия пользователя, и получить контроль над всеми устройствами в сети.

#### **Физическая компрометация устройства**

На заключительном этапе тестирования был исследован шлюз ZigBee. Через интерфейс UART были извлечены данные флеш-памяти, включающие список подключенных устройств и их идентификаторы. Кроме того, доступ к внутреннему файловому хранилищу позволил получить неизменный заводской ключ ZigBee, который мог быть использован для компрометации других сетей, использующих устройства данного производителя.

### Практические выводы

Результаты тестирования показали, что уязвимости на всех уровнях системы, включая использование статических ключей шифрования, недостатки аутентификации API и отсутствие аппаратной защиты устройства, делают систему подверженной как локальным, так и удаленным атакам. В условиях реального сценария злоумышленник может:

- Перехватывать и расшифровывать команды управления устройствами.
- Получить контроль над всеми подключенными устройствами через API.
- Компрометировать сеть, используя извлеченные ключи.

Для устранения выявленных проблем было предложено:

1. Внедрить динамическую генерацию ключей шифрования и безопасный процесс сопряжения устройств.
2. Реализовать привязку токенов аутентификации к устройствам на уровне API.
3. Обеспечить аппаратную защиту шлюза от несанкционированного доступа, например, через внедрение обнаружения вскрытия корпуса.

Этот пример демонстрирует важность тестирования на проникновение как инструмента для комплексного анализа безопасности IoT-устройств, способного выявить уязвимости как на уровне сетевой архитектуры, так и в программной и аппаратной компонентах системы.

\*\*\*

1. Красов А. В., Сахаров Д. В., Тасюк А. А. Проектирование системы обнаружения вторжений для информационной сети с использованием больших данных // Научные исследования в космических исследованиях Земли. – 2020. – Т. 12. – № 1. – С. 70–76.
2. Волкогонов В. Н. и др. Применение физически неклонированных функций для выполнения аутентификации в среде интернета вещей // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2021. – С. 409–414.
3. Лаврова Д. С. и др. Предупреждение Dos-атак путем прогнозирования значений корреляционных параметров сетевого трафика // Проблемы информационной безопасности. Компьютерные системы. – 2018. – № 3. – С. 70–77.
4. Пестов И. Е. Методика разработки управляющего воздействия на инстансы облачной инфраструктуры // Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. – 2020. – № 4. – С. 72–76.

**Каленикин В.Н., Панова М.О.**

### **Автоматизированная информационная система составления и оптимизации расписания высшего учебного заведения: актуальность внедрения, ключевые параметры**

*Уральский государственный экономический университет  
(Россия, Екатеринбург)*

doi: 10.18411/trnio-12-2024-877

#### **Аннотация**

В данной статье рассматривается вопрос актуальности создания и внедрения автоматизированной системы расписания высшего учебного заведения, а также ключевые параметры, на основании которых проектируется создание автоматизированной информационной системы.

**Ключевые слова:** автоматизированная информационная система, составление расписания, автоматизация расписания, модули, методы создания автоматизированной информационной системы.

#### **Abstract**

This article addresses the relevance of developing and implementing an automated scheduling system for higher education institutions, as well as the key parameters on which the design of this automated information system is based.

**Keywords:** automated information system, schedule creation, schedule automation, modules, methods for developing an automated information system.

Современный мир характеризуется стремительным внедрением различных средств вычислительной техники, которые помогают человечеству оптимизировать и упростить рабочие процессы. Глобальное использование информационных технологий не обошло стороной и образовательный процесс, так как ускоренное развитие технических средств меняет приоритеты современной системы образования.

Качественная организация учебного процесса является ключевой задачей каждого учебного заведения и заключается в эффективном и правильном создании расписания учебных занятий. Ручное создание расписания является достаточно долгим, сложным и трудоемким процессом, в ходе которого существует вероятность большего допущения ошибок в рамках распределения аудиторного фонда и профессорско-преподавательского состава. Именно поэтому в современных реалиях актуально использовать специально созданные автоматизированные информационные системы, которые оптимизируют и упростят процесс составления расписания.

Интерес к автоматизации составления и оптимизации расписания высших учебных заведений проявляли такие исследователи, как Н.Н. Клеванский, Р.В. Конвей, Э.Г. Коффман, В.Л. Максвелл, М.Г. Маслова, и др.

Существует несколько методов создания автоматизированной информационной системы составления расписания.

1. ГИС-методология анализирует и создает автоматизированное расписание на основе введенных данных.
2. Использование генетического алгоритма позволяет сформировать большое количество вариантов расписания под определённые требования. В итоге варианты комбинируются, меняются и создается новый и самый оптимальный вариант расписания.
3. Имитационное моделирование. Процесс создания начинается с пустого неоформленного расписания, где все занятия не учтены. Работа строится таким образом, что программа переходит от одного расписания к другому, пока не сформируется эффективный и «идеальный» вариант.
4. Последовательный метод предполагает последовательное заполнение шаблона, пока все окна и часы не будут заполнены.

Каждый из вышеперечисленных методов имеет свои положительные стороны, выражающиеся в простых настройках и наличии поискового отдела.

Преимущество использования автоматизированной информационной системы составления расписания занятий облегчает ряд таких действий как:

- проверка выполнения предъявленных требований;
- поиск ошибок в составленном расписании;
- возможность корректировки расписания, внесения изменений;
- устранение варианта бумажного заполнения расписания.

Все эти преимущества, безусловно, экономят личное время и сохраняют физическую трудоспособность людей.

Существует немалое количество автоматизированных информационных систем по составлению и оптимизации расписания. Например, «БИТ.ВУЗ.Расписание», «Галактики Расписание учебных занятий», «1С: Автоматизированное составление расписания для университета», и «АВТОРасписания». Однако каждая из этих систем имеет определенный спектр недостатков и не в полной мере удовлетворяет требованиям организации, в частности специалиста, который планирует и создает расписание.

Некоторые организации используют программные продукты Microsoft Office, такие, как Microsoft Excel, Microsoft Access, которая позволяет создавать базы данных для хранения информации.

Для качественной организации учебного процесса важно создать такую систему, которая будет отвечать и основным, и специфическим требованиям, что обеспечит максимально эффективную и бесперебойную работу в организации учебной деятельности.

Ключевыми параметрами в АИС является:

- сформированная база данных с информацией о преподавателях и учебных кабинетах;
- программная составляющая и интерфейс, в которые внедрены функции загрузки информации из баз данных, редактирования, выгрузки, а также проверки на выявление ошибок и их устранения.

Как правило, базы данных хранят основную информацию по определенным модулям. В АИС составления и оптимизации расписания высшего учебного заведения можно создать базу данных, которые включает в себя следующую информацию:

- модуль «Преподавательский состав», где хранится вся информация о профессорско-преподавательском составе;
- модуль «Учебные аудитории», где отражены данные и номера корпусов и аудиториях. Также можно внедрить функцию, которая проанализировав состав группы будет подбирать аудиторию в соответствии с количеством человек, иными словами «функция вместимости аудитории»;
- модуль «Учебные дисциплины» представляет собой информацию об учебных планах тех или иных факультетов и групп;
- модуль «Факультет/группа» хранит в себе данные о факультетах, их наполняемости и составе групп;
- модуль «Пары», где будет отражено уже готовое расписание со всеми необходимыми данными: состав преподавателей, номер аудитории, название группы/факультета, преподаваемый предмет, время проведения с указанием дня недели, дня и времени.

Одной из главных функций, а также преимуществом в данной автоматизированной системе будет создание вариантов доступа, которые будут соответствовать ролям пользователей и их запросам. Данная функция подразумевает пять вариантов доступа:

1. Доступ для заведующего кафедрой/декана: возможность создания предложений по расписанию, их корректировка.
2. Доступ для студентов: ограничивается просмотром существующего расписания и его выгрузки.
3. Доступ для преподавательского состава: возможность просмотра составленного расписания, формирование запроса на его изменение в случае необходимости.
4. Доступ для сотрудника учебно-методического отдела: возможность просмотра и редактирования расписания, разделение и структурирование учебных потоков.
5. Доступ для специалистов технической поддержки: возможность диагностировать и корректировать возникающие технические проблемы.

Эффективное функционирование АИС предполагает создание понятного пользовательского интерфейса:

- наличие меню, которое открывает доступ к вышеперечисленным модулям в базе данных;
- наличие кнопки выгрузки и сохранения составленного расписания в формате Excel;
- наличие кнопки ручного редактирования расписания;
- наличие кнопки проверки расписания на ошибки.

Добавление функции генетического алгоритма делает процесс расписания еще проще, быстрее и эффективнее. Данный алгоритм будет комбинировать большое количество расписаний и в завершение выдаст самый оптимальный вариант, который будет соответствовать всем требованиям.

Таким образом, создание данной автоматизированной системы позволит снизить объем работы диспетчера расписания и допущение ошибок при формировании расписания для учебных групп.

\*\*\*

1. Постольник В.С, Цебрено К.Н. Концепция автоматизированной системы составления расписания образовательной организации. Академия маркетинга и социально-информационных технологий – ИМСИТ.2024. №5–4 (92). С.38.
2. Гольченко Ю.В. К вопросу об использовании компьютерных автоматизированных систем для составления учебных занятий в вузе. Московский гуманитарно-экономический институт Волгоградский филиал. 2015. №3(6). С.67-70.
3. Ильин М.А. Разработка информационной системы по составлению расписания проведения занятий в ВУЗе с использованием ГИС-методологии URL: <https://nauchkor.ru/pubs/razrabotka-informatsionnoysistemy-po-sostavleniyu-raspisaniya-provedeniya-uebnyh-zanyatyi-v-vuze-sispolzovaniem-gis-metodologii-5ef4b657cd3d3e000106d259>
4. Кузнецов А.А. Разработка автоматизированной системы составления расписания для педагогического Вуза // Современные научные исследования и инновации. 2016. № 9.

**Калмыков С.Е., Вострова Е.А.**

### **Оптимизация мониторинга подключений и ресурсов в PostgreSQL**

*Санкт-Петербургский государственный университет телекоммуникаций им. проф.*

*М.А. Бонч-Бруевича*

*(Россия, Санкт-Петербург)*

*doi: 10.18411/trnio-12-2024-878*

#### **Аннотация**

В статье рассматривается оптимизация мониторинга подключений и ресурсов в PostgreSQL, что является важным аспектом для обеспечения высокой производительности и стабильности баз данных. Основное внимание уделяется разработке метода отслеживания количества одновременных запросов и потребляемых ими ресурсов. Статья будет полезна администраторам баз данных, разработчикам и всем, кто интересуется оптимизацией работы PostgreSQL.

**Ключевые слова:** мониторинг, подключения, запросы, ресурсы, PostgreSQL, база данных.

#### **Abstract**

The article discusses the optimization of connection and resource monitoring in PostgreSQL, which is an important aspect to ensure high database performance and stability. The main focus is on developing a method for tracking the number of simultaneous requests and the resources they consume. This article will be useful for database administrators, developers, and anyone interested in optimizing PostgreSQL.

**Keywords:** monitoring, connections, queries, resources, PostgreSQL, database.

В современном мире, где данные играют ключевую роль в функционировании организаций, эффективное управление базами данных становится критически важным. А для достижения максимальной производительности и стабильности требуется тщательный мониторинг подключений и ресурсов. Одним из ключевых аспектов мониторинга является отслеживание количества одновременных запросов и потребляемых ими ресурсов. Это позволяет администраторам баз данных и

разработчикам своевременно выявлять и устранять узкие места, оптимизировать производительность и предотвращать возможные сбои, а также выявить кибератаку.

Сегодня важно быстро определять производительность СУБД, анализировать проблемы с дисковым пространством, контролировать количество подключений. Поэтому современные GUI-клиенты предлагают собственные панели мониторинга для СУБД. Наиболее развитые системы мониторинга существуют в графических приложениях pgAdmin и DBeaver.

В DBeaver по умолчанию есть готовые шаблоны графиков для мониторинга: активные и неактивные сеансы, количество транзакция в секунду, количество блокировок при операциях ввода/вывода в секунду. Однако есть возможность по готовому шаблону настроить новые графики.

В pgAdmin также используются шаблоны, но их больше, чем в DBeaver. Появились шаблоны, отображающие количество вставленных/обновленных/удаленных записей и количество полученных/возвращённых записей. Также в pgAdmin есть вкладка со статистикой, которая позволяет выполнить более глубокий анализ.

Чтобы лучше понять принцип работы таких приложений, нами была поставлена цель создать свою версию такого средства мониторинга. Для визуализации нагрузки на ресурсы, создаваемой подключениями к базе данных, нами был создан скрипт на языке программирования Python, который способен имитировать необходимое количество подключений и выводить графики зависимости нагрузки на процессор, память и сеть от количества одновременных запросов.

Опишем код, начиная с используемых библиотек, их можно увидеть на рисунке 1.

```
import psutil
import datetime
import time
import threading
import tkinter as tk
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from matplotlib.figure import Figure
import psycpg2
```

Рисунок 1. Используемые библиотеки.

Библиотека psutil позволяет получать информацию о процессах и системе, datetime и time позволяют работать со временем, threading используется для работы с потоками. Использование потоков позволяет выполнять сбор данных и обновление графиков параллельно с интерфейсом пользователя, обеспечивая отзывчивость приложения. Tkinter – библиотека для создания графических интерфейсов, matplotlib.backends.backend\_tkagg позволяет интегрировать графики, а matplotlib.figure – создавать их, psycpg2 – библиотека для работы с PostgreSQL.

Затем создаем глобальные списки для хранения данных и функции для записи данных о производительности и для создания подключений, данная часть кода показана на рисунке 2. Функция collect\_data() запускает бесконечный цикл, в котором каждые 1 секунду собирает данные о загрузке ЦП, использовании памяти и сетевой активности. Эти данные затем добавляются в соответствующие списки (cpu\_data, memory\_data, network\_data). Функция create\_manual\_connections (num\_connections) устанавливает num\_connections подключений к базе данных PostgreSQL. При этом, если подключение невозможно, ошибка будет выведена в консоль. Время и количество подключений добавляются в список connection\_data.

```

# Глобальные списки для хранения данных
cpu_data = []
memory_data = []
network_data = []
connection_data = []

# Функция для записи данных о производительности
def collect_data():
    while True:
        # Получение текущей производительности
        cpu_usage = psutil.cpu_percent(interval=1)
        memory_info = psutil.virtual_memory()
        memory_usage = memory_info.percent

        # Получение текущей статистики сети
        network_info = psutil.net_io_counters()
        network_sent = network_info.bytes_sent
        network_recv = network_info.bytes_recv

        # Получение текущего времени
        current_time = datetime.datetime.now()

        # Сохранение данных
        cpu_data.append((current_time, cpu_usage))
        memory_data.append((current_time, memory_usage))
        network_data.append((current_time, network_sent + network_recv)) # Общее использование сети

        # Задержка перед следующим измерением
        time.sleep(0.5)

# Функция для создания заданного количества подключений
def create_manual_connections(num_connections):
    conn_string = "dbname='firma' user='bd' password='15426378' host='localhost' port='5431'"
    for _ in range(num_connections):
        try:
            conn = psycopg2.connect(conn_string)
        except Exception as e:
            print(f"Error connecting to the database: {e}")
        finally:
            if 'conn' in locals():
                conn.close()

# Добавляем запись о количестве подключений
connection_data.append((datetime.datetime.now(), num_connections)) # Добавляем количество подключений
update_connection_graph() # Обновляем график после добавления подключений

```

Рисунок 2. Первая часть кода.

Также будут необходимы функции для обновления графиков, часть кода с их описанием изображена на рисунке 3. Функция `update_graphs()` обновляет графики для всех собранных данных каждые 0.5 секунды, вызывая функции `update_cpu_graph()`, `update_memory_graph()`, `update_network_graph()` и `update_connection_graph()`. Каждая из функций обновления графиков очищает текущий график и строит новый на основе собранной информации. Здесь используются библиотеки `matplotlib` для построения графиков.

```

# Функция для обновления графиков
def update_graphs():
    while True:
        # Обновление графиков
        update_cpu_graph()
        update_memory_graph()
        update_network_graph()
        update_connection_graph()
        time.sleep(0.5)

# Функция для обновления графика CPU
def update_cpu_graph():
    if cpu_data:
        times, usages = zip(*cpu_data)
        ax1.clear()
        ax1.plot(times, usages, label='CPU (%)', color='red')
        ax1.set_title('CPU')
        ax1.set_xlabel('Time')
        ax1.set_ylabel('Usage (%)')
        ax1.legend()
        canvas1.draw()

# Функция для обновления графика Меморы
def update_memory_graph():
    if memory_data:
        times, usages = zip(*memory_data)
        ax2.clear()
        ax2.plot(times, usages, label='Memory (%)', color='green')
        ax2.set_title('Memory')
        ax2.set_xlabel('Time')
        ax2.set_ylabel('Usage (%)')
        ax2.legend()
        canvas2.draw()

# Функция для обновления графика Network
def update_network_graph():
    if network_data:
        times, usages = zip(*network_data)
        ax3.clear()
        ax3.plot(times, usages, label='Network (bytes)', color='blue')
        ax3.set_title('Network')
        ax3.set_xlabel('Time')
        ax3.set_ylabel('Usage (bytes)')
        ax3.legend()
        canvas3.draw()

```

Рисунок 3. Вторая часть кода.

Создание графических интерфейсов для мониторинга и для ввода количества подключения показано на рисунке 4 и на рисунке 5. Функция `create_gui()` создает основное окно приложения с графиками для мониторинга ресурсов и подключений. Функция `create_manual_connection_gui()` создает отдельное окно для ввода количества

подключений к базе данных. При нажатии кнопки запускается поток, который создает указанное количество подключений. В главной части кода (в блоке `if __name__ == "__main__":`) запускаются потоки для сбора данных, обновления графиков и создания GUI. Все потоки работают в фоновом режиме (демон).

```
# Функция для обновления графика Connections
def update_connection_graph():
    if connection_data:
        times, counts = zip(*connection_data)
        ax4.clear()
        ax4.plot(times, counts, label='Database Connections', color='purple')
        ax4.set_title('Database Connections')
        ax4.set_xlabel('Time')
        ax4.set_ylabel('Connections')
        ax4.legend()
        canvas4.draw()

# Создание графического интерфейса для мониторинга
def create_gui():
    global ax1, ax2, ax3, ax4, canvas1, canvas2, canvas3, canvas4

    root = tk.Tk()
    root.title("Monitoring")

    # Создание рамок для графиков
    left_frame = tk.Frame(root)
    left_frame.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

    right_frame = tk.Frame(root)
    right_frame.pack(side=tk.RIGHT, fill=tk.BOTH, expand=True)

    # Графики
    fig1 = Figure(figsize=(6, 4), dpi=100)
    ax1 = fig1.add_subplot(111)

    fig2 = Figure(figsize=(6, 4), dpi=100)
    ax2 = fig2.add_subplot(111)

    fig3 = Figure(figsize=(6, 4), dpi=100)
    ax3 = fig3.add_subplot(111)

    fig4 = Figure(figsize=(6, 4), dpi=100)
    ax4 = fig4.add_subplot(111)

    canvas1 = FigureCanvasTkAgg(fig1, master=left_frame)
    canvas1.get_tk_widget().pack(fill=tk.BOTH, expand=True)

    canvas2 = FigureCanvasTkAgg(fig2, master=left_frame)
    canvas2.get_tk_widget().pack(fill=tk.BOTH, expand=True)

    canvas3 = FigureCanvasTkAgg(fig3, master=right_frame)
    canvas3.get_tk_widget().pack(fill=tk.BOTH, expand=True)

    canvas4 = FigureCanvasTkAgg(fig4, master=right_frame)
    canvas4.get_tk_widget().pack(fill=tk.BOTH, expand=True)

    root.mainloop()
```

Рисунок 4. Третья часть кода.

```
# Создание графического интерфейса для ввода количества подключений
def create_manual_connection_gui():
    def on_button_click():
        try:
            num_connections = int(entry.get()) # Получаем количество подключений из текстового поля
            if num_connections > 0:
                threading.Thread(target=create_manual_connections, args=(num_connections,)).start()
            else:
                print("Please enter a positive integer.")
        except ValueError:
            print("Invalid input. Please enter a valid integer.")

    manual_root = tk.Tk()
    manual_root.title("Manual Connections")

    label = tk.Label(manual_root, text="Number of manual connections:")
    label.pack(pady=10)

    entry = tk.Entry(manual_root) # Текстовое поле для ввода количества подключений
    entry.pack(pady=10)

    button = tk.Button(manual_root, text="Create Connections", command=on_button_click)
    button.pack(pady=20)

    manual_root.mainloop()

# Запуск функций сбора данных и обновления графиков
if __name__ == "__main__":
    threading.Thread(target=collect_data, daemon=True).start()
    threading.Thread(target=update_graphs, daemon=True).start()
    threading.Thread(target=create_manual_connection_gui, daemon=True).start() # Запускаем GUI для ручных подключений
    create_gui() # Запускаем основной GUI для мониторинга
```

Рисунок 5. Четвертая часть кода.

В результате выполнения скрипта будут открыты два графических окна, которые предназначены для анализа нагрузки. В первом окне потребуется указать количество подключений к базе данных, это окно изображено на рисунке 6. А во втором окне будут выводиться четыре графика – нагрузка на процессор, нагрузка на память и нагрузка на сеть, а также график, показывающий сколько было подключений и в какое время, это окно можно увидеть на рисунке 7.

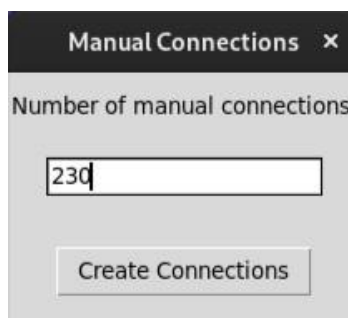


Рисунок 6. Первое окно.

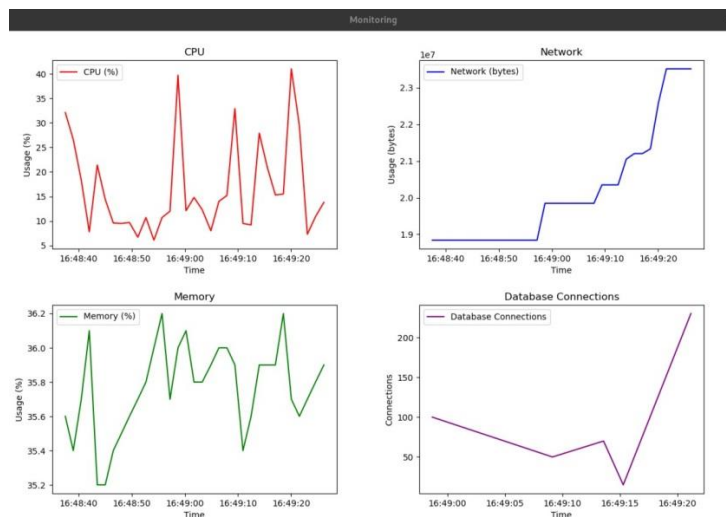


Рисунок 7. Второе окно.

В данной статье была рассмотрена оптимизация мониторинга подключений и ресурсов в PostgreSQL. Мы разработали средство, позволяющее отслеживать количество одновременных запросов к базе данных и потребляемых ими ресурсов, таких как нагрузка на CPU, память и сеть. Для этого был создан скрипт на Python, который имитирует требуемое количество одновременных запросов и выводит графики зависимости нагрузки на ресурсы от количества запросов.

\*\*\*

1. Документация по PostgreSQL 17.2 [Электронный ресурс]. – режим доступа URL: <https://www.postgresql.org/docs/>.
2. Psycorp2 [Электронный ресурс]. – режим доступа URL: <https://psycorp.org/>.
3. Matplotlib [Электронный ресурс]. – режим доступа URL: <https://matplotlib.org/>.
4. Tkinter [Электронный ресурс]. – режим доступа URL: <https://docs.python.org/3/library/tkinter.html>.
5. pgAdmin [Электронный ресурс]. – режим доступа URL: <https://www.pgadmin.org/>.
6. DBeaver [Электронный ресурс]. – режим доступа URL: <https://dbeaver.io/>.

**Лыткин К.А.**

**Разработка решения для автоматизированного парсинга ошибок  
в банковских системах**

*ФГБОУ ВО «КГЭУ»  
(Россия, Казань)*

doi: 10.18411/trnio-12-2024-879

#### **Аннотация**

Статья посвящена разработке автоматизированного решения для парсинга ошибок в логах банковских систем. Рассмотрены принципы работы с лог-файлами, включая идентификацию ошибок, фиксацию временных меток и уведомление о критических

событиях. Приведён пример Python-скрипта, который можно адаптировать для различных форматов логов и расширить для интеграции с системами оповещения. Описаны ключевые преимущества таких решений, включая повышение эффективности мониторинга, оперативное реагирование на сбои и возможность кастомизации.

**Ключевые слова:** парсинг логов, анализ ошибок, автоматизация мониторинга, Python-скрипты, банковские системы, уведомления о сбоях, регулярные выражения, обработка лог-файлов, безопасность данных, лог-анализ.

### Abstract

The article is devoted to the development of an automated solution for parsing errors in the logs of banking systems. The principles of working with log files are considered, including error identification, timestamp fixation and notification of critical events. An example of a Python script is provided that can be adapted for various log formats and extended for integration with notification systems. The key advantages of such solutions are described, including improved monitoring efficiency, prompt response to failures and the possibility of customization.

**Keywords:** log parsing, error analysis, monitoring automation, Python scripts, banking systems, crash notifications, regular expressions, log file processing, data security, log analysis.

Современные банковские системы ежедневно обрабатывают миллионы транзакций. Логично, что в таком сложном и высоконагруженном контексте сбои неизбежны. Вовремя выявлять ошибки и минимизировать их последствия — одна из ключевых задач для IT-департаментов банков. Но как справиться с этим эффективно, если объем данных огромен, а требования к скорости реакции высоки? Ответ очевиден: автоматизация.

Почему парсинг логов так важен?

Лог-файлы — это не просто строки текста. Они рассказывают историю работы систем, указывают на ошибки, предупреждают о возможных проблемах. Однако вручную анализировать такие данные — дело бесполезное. Огромный объем информации, сложность структурирования, необходимость быстрых реакций делают старые методы устаревшими. Автоматизированный парсинг ошибок — это способ не просто ускорить процесс, но и исключить человеческий фактор, повысив точность анализа.

Линукс как платформа для автоматизации.

Линукс давно стал стандартом для разработки автоматизированных решений в сфере IT. Его преимущества очевидны: открытый код, гибкость, огромный выбор инструментов. Для парсинга логов отлично подходят такие утилиты, как `grep`, `awk`, `sed`, а также более современные решения на основе Python.

Но, прежде чем углубляться в технические детали, представьте себе ситуацию: банк фиксирует проблему в системе онлайн-платежей. Задержки растут, клиенты недовольны. Здесь на сцену выходит автоматизация: специальный скрипт моментально находит ключевые ошибки, идентифицирует источник проблемы и уведомляет ответственных специалистов.

Архитектура решения.

Автоматизация анализа логов может быть реализована с использованием скриптов на Linux. Вот общая схема процесса:

1. Сбор данных. Логи разных систем (например, приложения интернет-банкинга, процессинга карт, серверов базы данных) агрегируются в центральное хранилище.
2. Фильтрация и разбор. С помощью скриптов фильтруются строки, содержащие ошибки, например, по ключевым словам: `error`, `exception`, `timeout`.
3. Структуризация информации. Данные преобразуются в удобный для анализа формат, например JSON.

4. Уведомление о проблемах. Результаты передаются в системы оповещения (Telegram, email, SMS).

Ниже предоставлен пример скрипта на языке python (рисунок 1). Этот скрипт не только находит ошибки, но и фиксирует их время. Его можно адаптировать для разных логов и расширить для отправки уведомлений.

```
import re

# Путь к лог-файлу
log_file = '/var/log/banking_system.log'

# Ключевые слова для поиска
keywords = ['error', 'exception', 'timeout']

# Открываем лог и ищем ошибки
with open(log_file, 'r') as file:
    for line in file:
        if any(keyword in line.lower() for keyword in keywords):
            match = re.search(r'\[(\d+-\d+-\d+ \d+:\d+)\]', line) # Извлекаем временную метку
            timestamp = match.group(1) if match else "Unknown"
            print(f"{timestamp}: {line.strip()}")
```

Рисунок 1. Пример скрипта для парсинга ошибок.

Преимущества данного решения:

- Универсальность: Подходит для различных систем, будь то банковские или IT-платформы.
- Эффективность: Позволяет быстро находить и классифицировать ошибки без ручного анализа.
- Автоматизация: Включение механизма уведомлений значительно упрощает мониторинг.

Альтернативы.

Существуют готовые инструменты, такие как Logstash (часть Elastic Stack), которые позволяют обрабатывать и визуализировать логи на более высоком уровне. Однако у них есть минусы: сложность настройки, зависимость от сторонних компонентов и высокая стоимость внедрения в больших масштабах. Самописные решения, напротив, дешевле и гибче, но требуют технической компетенции.

Вопросы безопасности.

Автоматизация не должна идти вразрез с принципами безопасности. Все операции с логами должны выполняться с учетом защиты конфиденциальных данных. Например, в логах могут содержаться персональные данные клиентов, что обязывает использовать методы маскировки или шифрования.

Что же касается реальных примеров, то большинство компаний пишут свои скрипты для работы с логами. Например, некоторые из них:

1. Анализ и сбор логов с использованием Graylog. В крупных компаниях, таких как ITSoft, активно применяются самописные решения на основе Graylog и Elasticsearch для анализа логов. Такие системы собирают данные с серверов, фильтруют их и отправляют алерты администратору в случае сбоев. Это помогает оперативно реагировать на критические события и предотвращать простои систем.
2. Мониторинг системных параметров и автоматизация уведомлений. Bash-скрипты используются для мониторинга загрузки процессора, памяти и диска, а также отправки уведомлений при превышении порогов. Такие скрипты полезны для серверных администраторов, чтобы избежать перегрузки системы и реагировать на возможные сбои до их возникновения.
3. Работа с API для автоматизированного сбора данных. Скрипты интегрирующиеся с API (например, для получения данных о состоянии системы или внешних курсов валют), используются для автоматического

уведомления и формирования отчетов. Они позволяют администраторам сосредоточиться на стратегических задачах, минимизируя ручной труд.

4. Скрипты для финансовых организаций. В банках и финансовых компаниях, таких как «Сбер», пишут собственные утилиты для обработки журналов транзакций. Это позволяет обнаруживать подозрительную активность в реальном времени, выявляя потенциальные угрозы безопасности и оптимизируя работу аналитиков.

В заключении важно ответить, что автоматизированный парсинг ошибок в банковских системах — не просто тренд, а насущная необходимость. Благодаря доступным инструментам и платформам, таким как Linux, это решение может быть внедрено быстро и эффективно. В результате же мы получаем снижение времени простоя, повышение удовлетворенности клиентов и, конечно, конкурентное преимущество на рынке.

\*\*\*

1. Григорьев, Н. А. Нейросетевая система парсинга логов / Н. А. Григорьев // Мавлютовские чтения : Материалы XVI Всероссийской молодежной научной конференции. В 6-ти томах, Уфа, 25–27 октября 2022 года. Том 5. – Уфа: Уфимский государственный авиационный технический университет, 2022. – С. 533-540.
2. Шелухин, О. И. Обнаружение аномальных состояний компьютерных систем средствами интеллектуального анализа данных системных журналов / О. И. Шелухин, В. С. Рябинин, М. А. Фармаковский // Вопросы кибербезопасности. – 2018. – № 2(26). – С. 33-43.
3. Шелухин, О. И. Мониторинг аномальных состояний компьютерных систем средствами интеллектуального анализа данных системных журналов / О. И. Шелухин, Д. В. Костин // Нейрокомпьютеры: разработка, применение. – 2020. – Т. 22, № 2. – С. 53-65.
4. Свидетельство о государственной регистрации программы для ЭВМ № 2018661676 Российская Федерация. Распределенная система сбора, обработки и анализа событий информационной безопасности сетевой инфраструктуры предприятия : № 2018619151 : заявл. 28.08.2018 : опубл. 12.09.2018 / Е. А. Басыня ; заявитель Общество с ограниченной ответственностью "Научно-исследовательский институт информационно-коммуникационных технологий".

**Лыткин К.А., Ильина А.А.**

**Raspberry Pi и система умного дома: настройка и мониторинг через Telegram бота**

*ФГБОУ ВО «КГЭУ»*

*(Россия, Казань)*

*doi: 10.18411/trnio-12-2024-880*

#### **Аннотация**

Статья посвящена разработке интеграции системы умного дома на базе одноплатного компьютера Raspberry Pi с Telegram ботом. Рассматриваются принципы настройки и управления устройствами умного дома, а также периферийным оборудованием. Подробно описаны этапы разработки интеграции с использованием языка Python и скриптов Bash, что позволяет обеспечить удобный мониторинг и управление через Telegram.

**Ключевые слова:** умный дом, интеграция, Raspberry Pi, Telegram, бот, Python, Bash, разработка, мониторинг, управление, VPN.

#### **Abstract**

The article is devoted to the development of integration of a smart home system based on a Raspberry Pi single-board computer with a Telegram bot. The principles of setting up and managing smart home devices, as well as peripheral equipment, are considered. The stages of integration development using Python and Bash scripts are described in detail, which allows for convenient monitoring and management via Telegram.

**Keywords:** smart Home, integration, Raspberry Pi, Telegram, Bot, Python, Bash, development, monitoring, management, VPN.

Современные системы умного дома часто ассоциируются с высокой стоимостью и сложностью настройки. Основная доля затрат приходится на интеграцию оборудования и разработку программного обеспечения, специфичного для конкретных устройств. Однако требования к вычислительным ресурсам таких систем сравнительно невысоки, что позволяет использовать доступные и компактные устройства, такие как Raspberry Pi.

Цель данной статьи — разработать систему умного дома, которая объединяет управление устройствами через Telegram-бота. Это решение демонстрирует, как минимизировать затраты и упростить процесс настройки за счет доступных технологий.

Актуальность работы обусловлена необходимостью автоматизации управления устройствами в повседневной жизни, что способствует повышению удобства и эффективности взаимодействия человека с окружающей средой.

Для начала важно определить, что потребуется для реализации системы. В качестве вычислительного устройства будет использоваться Raspberry Pi 4B. Технические характеристики этой модели, как и младших версий, таких как Raspberry Pi 2 или Raspberry Pi Zero, подходят для решения подобных задач. В данном проекте будет использована версия с 4 ГБ оперативной памяти и накопителем на 128 ГБ формата NVMe, что обеспечит высокую производительность. Дополнительным преимуществом Raspberry Pi является наличие GPIO-пинов, которые позволяют подключать умные устройства напрямую.

Пользователь будет управлять системой через Telegram-бот, отправляя команды или нажимая кнопки. Для разработки бота выбрана популярная библиотека Python Telegram Bot. Связь между оборудованием и Telegram-ботом осуществляется через GPIO-пины, предоставляемые Raspberry Pi, что обеспечивает прямое взаимодействие с подключёнными устройствами.

Процесс интеграции бота с GPIO-пинами Raspberry Pi достаточно прост и не требует сложных настроек. Для реализации нам потребуются такие библиотеки, как RPi.GPIO, которая отвечает за управление пинами, и Python Telegram Bot, предназначенная для обработки пользовательских команд, передаваемых через интерфейс мессенджера. В коде бота активно используются callback-кнопки, которые связываются с действиями, такими как включение или выключение света, активация других устройств, а также выполнение более сложных команд. Когда пользователь нажимает кнопку в интерфейсе Telegram, боту передаётся специальный сигнал, запускающий соответствующую функцию. Эта функция может, например, управлять GPIO-пинами для активации осветительных приборов или отправлять команды через инфракрасный порт для управления бытовой техникой, такой как кондиционер или телевизор (рис. 1). Такой подход обеспечивает гибкость и удобство управления различными устройствами с помощью Telegram.

```

GPIO.py > ...
1 import RPi.GPIO as GPIO
2 from telegram import Update, InlineKeyboardButton, InlineKeyboardMarkup
3 from telegram.ext import Updater, CommandHandler, CallbackQueryHandler
4 GPIO.setmode(GPIO.BCM)
5 GPIO.setup(17, GPIO.OUT)
6 def toggle_light(update: Update, context):
7     keyboard = [
8         [InlineKeyboardButton("Включить свет", callback_data='on'),
9          InlineKeyboardButton("Выключить свет", callback_data='off')]
10    ]
11    reply_markup = InlineKeyboardMarkup(keyboard)
12    update.message.reply_text("Выберите действие:", reply_markup=reply_markup)
13 def button(update: Update, context):
14     query = update.callback_query
15     query.answer()
16     if query.data == 'on':
17         GPIO.output(17, GPIO.HIGH) # Включение света
18         query.edit_message_text(text="Свет включен")
19     elif query.data == 'off':
20         GPIO.output(17, GPIO.LOW) # Выключение света
21         query.edit_message_text(text="Свет выключен")
22 updater = Updater("YOUR_API_KEY", use_context=True)
23 dp = updater.dispatcher
24 dp.add_handler(CommandHandler('start', toggle_light))
25 dp.add_handler(CallbackQueryHandler(button))
26 updater.start_polling()
27 updater.idle()

```

Рисунок 1. Пример кода по управлению GPIO-пинами через Telegram бота.

После того как основной функционал управления устройствами через Telegram-бота настроен, следующим шагом является внедрение автоматизации процессов. Это

позволит пользователю не только вручную управлять освещением, кондиционером и другими устройствами, но и настроить их работу по расписанию или в зависимости от событий. Для этого можно использовать Telegram-бота в сочетании с планировщиком задач, например, cron, чтобы создавать гибкие сценарии автоматизации (рис. 2). Таким образом можно создать свои пользовательские сценарии через бота.

```
def handle_time_input(update: Update, context):
    time_str = update.message.text
    try:
        time_obj = time.strptime(time_str, "%H:%M")
        scheduled_time = time.strptime("%H:%M", time_obj)

        # Запланировать задачу через cron
        command = f"{scheduled_time} * * * * /usr/bin/python3 /home/pi/{context.user_data['command']}.py"
        os.system(f"crontab -l; echo '{command}' | crontab -")

        update.message.reply_text(f"Задача на {context.user_data['action']} света запланирована на {scheduled_time}")
    except ValueError:
        update.message.reply_text("Неверный формат времени. Пожалуйста, введите время в формате HH:MM.")
```

Рисунок 2. Пример функции с использованием cron (планировщик задач).

После настройки системы важно тщательно протестировать функциональность как аппаратных, так и программных компонентов. Отладка же может включать в себя просмотр логов, проверку системных настроек и устранение неполадок в связи между Telegram-ботом и Raspberry Pi.

Не стоит забывать о важности удаленного доступа к Raspberry Pi для корректировки настроек, мониторинга работы устройства и анализа нагрузки на систему. Одним из наиболее простых и удобных решений является настройка VPN-сети, которая позволяет устройствам внутри сети взаимодействовать между собой. В данном случае мы выбрали ZeroTier, создав в личном кабинете отдельную виртуальную сеть. Это обеспечивает доступ к устройству из любой точки мира при наличии подключения к интернету.

Преимущества у данного решения значительны:

- Экономия бюджета: экономичное решение, которое использует недорогое оборудование, такое как Raspberry Pi, и обеспечивает простое в использовании управление с помощью Telegram.
- Легкость в настройке: написание кода не займет много времени, так как все используемые функции уже заложены в необходимыми библиотеками.
- Высокая адаптация к различным устройствам и сценариям: можно подключить практически любое устройство, прописав соответствующие функции

Из минусов можно выделить, что рост устройств способствует большему отклику и потере производительности, так как будут затрачены большие ресурсы. Например, обработка видео или фото добавит куда большую задержку, чем включение или выключение света.

В заключении, важно отметить, что, объединив мощь Raspberry Pi и простоту использования Telegram, мы продемонстрировали, как управлять такими устройствами, как освещение, камеры и системы климат-контроля. Кроме того, мы показали, как можно реализовать автоматизацию с помощью планирования задач, предлагая пользователям возможность управлять своими устройствами как вручную, так и автоматически. Хотя система обладает многими преимуществами, включая низкую стоимость, высокую настраиваемость и простоту использования, она также имеет определенные ограничения, связанные с аппаратными возможностями Raspberry Pi и зависимостью от стабильного интернет-соединения.

\*\*\*

1. Берчук, Д. Ю. Построение системы "Умный Дом" с использованием платформы "Arduino" и компьютера "Raspberry Pi" / Д. Ю. Берчук, Д. В. Журавлев // Молодежь и современные информационные технологии : Сборник трудов XI Международной научно-практической конференции студентов, аспирантов и молодых ученых, Томск, 13–16 ноября 2013 года. – Томск: Национальный исследовательский Томский политехнический университет, 2013. – С. 198-199.
2. Пархоменко, А. А. Реализация системы "умный дом" на основе Raspberry Pi / А. А. Пархоменко // Новые информационные технологии в исследовании сложных структур : материалы Десятой

- российской конференции с международным участием, пос. Катунь, 09–11 июня 2014 года / МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ, Томский государственный университет, Горно-Алтайский государственный университет, Институт оптики атмосферы им. В.Е. Зуева СО РАН. – пос. Катунь: Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования Национальный исследовательский Томский государственный университет, 2014. – С. 30-31.
3. Пасюков, А. А. Теория разработки системы "Умный дом" на базе платформы Arduino и микрокомпьютера Raspberry Pi / А. А. Пасюков, А. С. Якимов // Информационные технологии и прикладная математика : Всероссийский межвузовский сборник аспирантских и студенческих научных работ, Арзамас, 25 ноября 2016 года. Том Выпуск 7. – Арзамас: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, Арзамасский филиал, 2017. – С. 153-157.
  4. Разработка системы умного дома на базе raspberry pi и arduino / Б. Омуралиева, Н. Батырканов, К. Абдылдаев, Б. Б. Кошоева // Известия Кыргызского государственного технического университета им. И. Раззакова. – 2015. – № 2(35). – С. 35-40.

**Магамадов А-С.С., Данаев М.Х., Шахбазова М.С.**

### **Java vs Python: какой язык выбрать начинающему программисту**

*Чеченский государственный университет имени А.А. Кадырова  
(Россия, Грозный)*

doi: 10.18411/trnio-12-2024-881

#### **Аннотация**

В этой статье анализируется выбор между двумя языками программирования. Обсуждаются преимущества и недостатки языков, их области применения в разных сферах, а также востребованность на рынке труда. Материал должен помочь новичкам в программировании сделать осознанный выбор в пользу какого-либо из двух обсуждаемых языков в зависимости их интересов и целей в данной сфере.

**Ключевые слова:** Python, JavaScript, JS, Java, программа, программирование.

#### **Abstract**

This article analyzes the choice between two programming languages. The advantages and disadvantages of languages, their fields of application in various fields, as well as their relevance in the labor market are discussed. The material should help beginners in programming to make an informed choice in favor of one of the two languages discussed, depending on their interests and goals in this area.

**Keywords:** Python, JavaScript, JS, Java, program, programming.

Человечество постоянно совершенствует технологии, вместе с которыми совершенствуются инструменты для их использования. В эру цифровых технологий программирование является неотъемлемой частью различных сфер общества. Некоторые люди иногда не понимают, что основой, благодаря которой работают разные устройства, является именно программирование. Языки программирования являются той самой основой и инструментами, или же, точнее, фундаментом многих технологий современного мира.

С постоянным развитием науки в этой сфере вакансий на работу в ней становится лишь больше, тем самым еще больше расширяя ее важность и влияние в мире. Далее будет обсуждено, какой из двух самых распространенных языков следует выбрать для начала, чтобы ворваться в этот круговорот событий и стать разработчиком программ.

JavaScript для начинающих:

JavaScript – язык программирования, который разработчики используют для создания веб-страниц. Функции данного языка улучшают удобство взаимодействия пользователя с веб-сайтом.

JavaScript – это один из тех языков, который можно освоить без опыта работы в программировании. На нем можно создавать не только веб-страницы, но и серверные приложения, 3D-игры и даже программировать робо-технику.

Основа любого языка программирования – это типы данных, переменные, операторы, циклы и функции. Знания в этом направлении будут достаточны, чтобы научиться самостоятельно создавать программы на JavaScript.

Однако же, одновременно с JavaScript, или даже раньше, нужно будет освоить HTML и CSS. Это нужно, чтобы знать, как делать каскадные таблицы стилей и разметки, с помощью которых и создаются веб-страницы.

Плюсы и минусы JavaScript:

Одним из главных преимуществ JavaScript – это «незаменимость» в области веб-разработки. JS является основным инструментом для клиент-серверных моделей и веб-программирования.

Также плюсом JS является простота (в отличие от многих языков) и рациональность применения. Решить какую-либо задачу на JavaScript удастся за несколько минут, а сам язык обладает понятным и легким в написании синтаксисом. То есть, синтаксис языка не требует углубленного изучения.

Еще одним плюсом языка является его скорость и производительность – JS может сделать итоговую программу более быстрой и удобной. Также преимущество JS – наличие собственной мощной инфраструктуры.

О кроссплатформенности языка также стоит упомянуть: благодаря ей, язык может адаптировать один и тот же код, как для персональных компьютеров, так и для мобильных устройств, что очень полезно при написании сайтов.

Далее, к минусам: во-первых, это, конечно же, отсутствие возможности чтения или загрузки документов. Во-вторых, язык не используется для сетевого ПО полноценно – минус в отсутствии удаленного доступа.

К еще одному минусу JavaScript можно отнести его «вольную трактовку типов». Говоря другими словами: нестыковки в JS-коде будут полностью игнорироваться, что может привести к некоторым проблемам в разработке.

Однако, несмотря на ограничения и некоторые минусы, JS постоянно развивается и является одним из самых популярных языков программирования, тем самым являясь востребованным до сих пор. Множество библиотек и инструментов, которые делают язык универсальным, определенно требуют к себе внимания при выборе языка программирования для дальнейшего обучения.

Python для начинающих:

Python – это высокоуровневый язык программирования с богатой историей. Несмотря на свой возраст до сих пор пользуется популярностью и постоянно совершенствуется. Имеет огромный арсенал библиотек, тем самым делая его разносторонним и востребованным в разных областях разработок.

Python широко используется в разработке программного обеспечения, интернет-приложениях, а также в машинном обучении. Разработчики используют Python из-за его простоты кода, как в написании, так и в понимании, а также из-за того, что он работает на разных платформах.

К основным характеристикам этого языка относят объектно ориентированность. В его основе лежат понятие объекта и класса.

Плюсы и минусы Python:

Естественно, как у любого языка, у Python есть свои плюсы и минусы. К основным его преимуществам относят его интерпретируемость. Как уже было сказано выше, код на этом языке можно запустить на любой платформе, если на ней установлен интерпретатор, который выполняет последовательное исполнение оператора. Плюсом можно назвать и его динамическую типизацию, благодаря которой переменная, которой присвоит значение, сразу же связывается с определенным типом. Конечно, это как плюс, так и минус – ибо из-за данной функции возможны ошибки.

Преимущества Python перед другими языками – это его простота в освоении: язык простой, логичный и с понятным синтаксисом, из-за чего является излюбленным языком программирования у новичков. Отсюда вытекает его следующий плюс: благодаря простоте, создание кода на Python осуществляется очень быстро.

Благодаря своей универсальности, язык используется в различных задачах: от тестов до научных работ. Из-за, собственно, возраста и достаточно долгой стоянки на рынке труда из-за отсутствия конкурентов получил огромное количество инструментов. Библиотеки данного языка написаны на мощных и быстрых языках, из-за чего инструменты Python мощнее, а отладка кода быстрее. Конечно, несмотря на все это, язык до сих пор считается медленным. Из-за этого недостатка на Python нельзя написать программу с высокой производительностью.

К еще одному недостатку относится невозможность перенести программу на языке Python на другую систему из-за его зависимости от библиотек. Ну и, собственно, язык не подходит для разработки программ, которые тратят большое количество памяти.

Какой из двух языков выбрать для начала:

Стоит для начала сказать, что выбор языка зависит исключительно от настроек и интересов новичка. Однако, если давать короткую характеристику обоим языкам, то можно сказать следующее: если JS – это сложность и строгость, то Python – это простота и читаемость. Несмотря на легкость JS, Python обходит его по простоте изучения в несколько раз. JavaScript имеет концепцию объектно-ориентированного программирования и имеет синтаксис более строгий, чем у Python.

Естественно, как было сказано выше о JS, учить его может быть чуть сложнее для начинающих программистов в отличие от Python, однако, его строгость поможет понять основы программирования лучше, чем само изучение языка Python. Да – сложнее, да – чуть долго, но фундамент знаний будет крепче.

Еще в обсуждении хотелось бы отметить следующую поговорку о Java:

«Все, что нужно написать на JavaScript, в конечном итоге будет написано на JavaScript»

Данной поговоркой, которая часто гуляет среди JS-разработчиков, можно представить, что из себя представляет этот язык программирования, а именно, его можно назвать разносторонним, что сделает программиста, который знает Java, востребованным везде.

В противовес этой поговорке, есть маленькое объяснение своего языка от Python-разработчиков:

Python: `print('hello world')`

JavaScript: `console.log('hello world')`

Тут можно заметить, почему Python проще, чем JS. Глубокое изучение языков даст лучше понять, как Python обходит по простоте JavaScript. Среди Python-разработчиков есть даже такая шутка, что для программирования на Python достаточно основ английского языка.

### Заключение

Выбор между Java и Python для начинающего программиста – это важный шаг, который зависит от целей в обучении. Python предлагает простой и очень легкий вход в сферу программирования, что делает его лучшим для тех, кто хочет начать создавать свои проекты и изучить основы. С другой стороны, Java представляет более структурированный подход, который заложит крепкий фундамент разработчика, являясь идеальным выбором для работы в области корпоративной разработки или же мобильных приложений.

\*\*\*

1. Крокфорд, Д. «JavaScript. Сильные стороны», 2013.
2. Флэнаган, Д. «JavaScript. Подробное руководство, 6-е издание»,
3. Симпсон, К. «Вы пока не знаете {JS}. Познакомьтесь, JavaScript», 2022.
4. Любанович, Б. «Простой Python. Современный стиль программирования», 2021.
5. Лутц, М. «Изучаем Python. 4-е издание», 2011.
6. Гэддиес, Т. «Начинаем программировать на Python. 5-е издание», 2022.

**Мизаев М.М.<sup>1</sup>, Шахбиев Д.О-Х.<sup>1</sup>, Муцурова З.М.<sup>2</sup>**  
**Особенности языка C++, применение и преимущества**

<sup>1</sup>ФГБОУ ВО «Чеченский государственный университет им А.А. Кадырова»

<sup>2</sup>ФГБОУ ВО «Чеченский государственный педагогический университет»  
(Россия, Грозный)

doi: 10.18411/trnio-12-2024-882

### Аннотация

Данная статья посвящена ключевым аспектам языка C++, таким как управление памятью, оптимизацию производительности, объектно-ориентированное программирование. Подчеркивается практическая значимость C++ для разработки сложных приложений.

**Ключевые слова:** программирование, программа, компьютер, технологии, разработка.

### Abstract

This article is devoted to key aspects of the C++ language, such as memory management, performance optimization, and object-oriented programming. The practical importance of C++ for the development of complex applications is emphasized.

**Keywords:** programming, software, computer, technology, development.

C++ — один из самых влиятельных языков программирования в истории. Он появился ещё в 1980-х, как "расширение" языка C, добавившего поддержку объектов и более удобных инструментов для разработки. Но C++ не остался просто дополнением. Он стал самостоятельным языком, который используется во множестве областей: от создания игр и операционных систем до машинного обучения и разработки ПО для авиакосмической отрасли.

Почему он так популярен? Потому что C++ даёт программисту полный контроль над ресурсами компьютера, при этом оставаясь достаточно высокоуровневым, чтобы создавать сложные программы с меньшим количеством кода. Он позволяет писать код, который одновременно быстрый, гибкий и масштабируемый.

Эта статья поможет вам сделать первые шаги в изучении C++. Мы разберём, как он работает, изучим основные конструкции языка и научимся писать простые программы. Всё будет объяснено на примерах, чтобы каждый смог понять и повторить. Независимо от вашего опыта, C++ станет отличной основой для развития навыков программирования.

Пример 1: Первая программа

Вот с чего начинают почти все:

```
#include <iostream>
int main() {
    std::cout << "Привет, мир!" << std::endl;
    return 0;
}
```

Объяснение:

- `#include <iostream>` — подключаем библиотеку, которая позволяет выводить текст на экран.
- `int main()` — это главный вход в программу. Именно отсюда она начинает выполняться.
- `std::cout` — команда, которая выводит текст. В данном случае мы печатаем "Привет, мир!".
- `std::endl` — добавляет перенос строки.

Когда запускаешь этот код, в консоли появится текст. Всё просто!

Пример 2: Работа с переменными

Переменные — это ячейки, в которых хранятся данные. Например, возраст человека, его имя или рост. Вот пример:

```
#include <iostream>
#include <string>
int main() {
    int age = 25;           // Целое число
    double height = 1.8;   // Число с запятой
    std::string name = "Аня"; // Текст
    std::cout << "Имя: " << name << ", возраст: " << age << std::endl;
    std::cout << "Рост: " << height << " м" << std::endl;
    return 0;
}
```

Объяснение:

- `int`, `double`, `std::string` — это типы данных. Они определяют, что именно мы храним.
- `<<` позволяет соединить текст и данные для вывода.
- В итоге программа напечатает:
- Имя: Аня, возраст: 25
- Рост: 1.8 м

Пример 3: Условия

Теперь добавим немного логики. Например, проверим, положительное число ввёл пользователь или нет:

```
#include <iostream>
int main() {
    int number;
    std::cout << "Введите число: ";
    std::cin >> number; // Считываем данные с клавиатуры
    if (number > 0) {
        std::cout << "Это положительное число." << std::endl;
    } else if (number < 0) {
        std::cout << "Это отрицательное число." << std::endl;
    } else {
        std::cout << "Это ноль." << std::endl;
    }
    return 0;
}
```

Объяснение:

- `std::cin` — это ввод данных от пользователя.
- `if` проверяет условие: больше нуля или нет.
- `else if` добавляет ещё одну проверку.
- `else` выполняется, если все предыдущие условия не подошли.

Попробуй запустить код и поэкспериментировать с разными числами.

Пример 4: Циклы

Циклы нужны, чтобы повторять действия. Вот пример, где мы печатаем числа от 1 до 5:

```
#include <iostream>

int main() {
    for (int i = 1; i <= 5; ++i) {
        std::cout << "Число: " << i << std::endl;
    }
    return 0;
}
```

Объяснение:

- `for` — это цикл.
- `int i = 1` — начинаем с числа 1.
- `i <= 5` — продолжаем, пока `i` меньше или равно 5.
- `++i` — увеличиваем `i` на единицу на каждой итерации.

Результат в консоли:

Число: 1

Число: 2

Число: 3

Число: 4

Число: 5

Пример 5: Классы

В C++ можно создавать свои структуры данных — классы. Это удобно, если нужно работать с чем-то более сложным, чем переменные. Например, с машинами:

```
#include <iostream>
#include <string>
class Car {
public:
    std::string brand;
    int year;
    void displayInfo() {
        std::cout << "Марка: " << brand << ", год выпуска: " << year << std::endl;
    }
};
int main() {
    Car car1;
    car1.brand = "Toyota";
    car1.year = 2020;
    car1.displayInfo();
    return 0;
}
```

Объяснение:

- `class Car` — это шаблон. Он описывает, какие свойства и действия есть у машин.

- car1 — это объект, экземпляр класса Car. У него есть свойства brand и year, которые мы задаём.
- displayInfo() — это метод класса. Он выводит данные об объекте.

#### Работа с памятью в C++

Одна из самых мощных и уникальных возможностей C++ — это управление памятью. В отличие от многих современных языков, где сборщик мусора делает всю работу за программиста, в C++ вы сами решаете, как выделять и освобождать память. Это даёт невероятный контроль над производительностью, но требует внимательности.

#### Динамическое выделение памяти

Когда вы не знаете заранее, сколько данных нужно программе, можно использовать динамическую память. Для этого в C++ есть операторы new и delete. Вот пример:

```
#include <iostream>

int main() {
    int* ptr = new int; // Выделяем память для одного целого числа
    *ptr = 42;         // Присваиваем значение

    std::cout << "Число: " << *ptr << std::endl;

    delete ptr;      // Освобождаем память
    return 0;
}
```

#### Объяснение:

- new int выделяет место в памяти для хранения одного числа и возвращает указатель на него.
- \*ptr = 42 записывает значение в выделенную память.
- delete ptr освобождает память, чтобы избежать утечек.

#### Массивы в динамической памяти

Если вам нужен массив переменной длины, используйте динамическое выделение для массивов:

```
#include <iostream>

int main() {
    int size;
    std::cout << "Введите размер массива: ";
    std::cin >> size;
    int* arr = new int[size]; // Создаём массив
    for (int i = 0; i < size; ++i) {
        arr[i] = i * 2; // Заполняем массив числами
    }
    std::cout << "Элементы массива: ";
    for (int i = 0; i < size; ++i) {
        std::cout << arr[i] << " ";
    }
    std::cout << std::endl;
    delete[] arr; // Освобождаем память
    return 0;
}
```

Здесь важно помнить о delete[], который освобождает память, выделенную для массива.

### Умные указатели

Писать `delete` вручную неудобно и чревато ошибками. В современном C++ используются умные указатели из библиотеки `<memory>`, которые автоматически освобождают память, когда она больше не нужна. Например:

```
#include <iostream>
#include <memory> // Для умных указателей

int main() {
    std::unique_ptr<int> ptr = std::make_unique<int>(42); // Умный указатель
    std::cout << "Число: " << *ptr << std::endl;
    // Умный указатель автоматически освободит память
    return 0;
}
```

С умными указателями вы можете забыть о ручном управлении памятью и сосредоточиться на логике программы.

### Ошибки при работе с памятью

В C++ важно избегать утечек памяти и использования неинициализированных указателей. Вот несколько советов:

1. Всегда освобождайте память, выделенную с помощью `new`.
2. Не используйте указатели после `delete`.
3. Пользуйтесь инструментами для поиска утечек памяти, например `Valgrind`.

Работа с памятью в C++ может быть сложной, но она даёт невероятную мощь и гибкость. Разобравшись с этим, можно писать высокопроизводительный код для любых задач.

Изучение C++ может показаться сложным, но именно этот язык даст вам глубокое понимание программирования. Его ключевые преимущества — это скорость и контроль. Благодаря C++ вы можете управлять памятью, писать высокопроизводительные программы и разрабатывать как низкоуровневое, так и высокоуровневое ПО. На C++ создают множество приложений: от игр с потрясающей графикой до сложных систем для финансовых институтов и научных исследований. Даже современные языки, такие как Python или JavaScript, за кулисами часто опираются на библиотеки, написанные на C++. Были рассмотрены базовые концепции: работу с переменными, условия, циклы и классы. Это лишь верхушка айсберга. Дальше вы узнаете о более сложных темах: указатели, шаблоны, многопоточность. Чем больше вы будете практиковаться, тем легче станет писать эффективный и красивый код. C++ это инструмент, который пригодится в любой области разработки. Освоив его, вы получите универсальный навык, который будет ценен в течение всей вашей карьеры.

C++ может показаться сложным на первый взгляд, но он открывает двери в мир программирования на высоком уровне. На нём пишут игры, программы для космоса, банковские системы. Он быстрый, универсальный и даёт полный контроль над процессами.

\*\*\*

1. Karin Kelley. What Is Artificial Intelligence: Types, History and Future. 2022. [Электронный ресурс]. URL: <https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/what-is-artificial-intelligence>
2. Rebecca Reynoso. A Complete History of Artificial Intelligence. 2021. [Электронный ресурс]. URL: <https://www.g2.com/articles/history-of-artificial-intelligence>
3. What is artificial intelligence and how is it used? 2020. [Электронный ресурс]. URL: <https://www.europarl.europa.eu/news/en/headlines/society/20200827STO85804/what-is-artificial-intelligence-and-how-is-it-used>

**Напалков С.В.**

**Центры обработки данных в мире. Глобальное распределение и тренды развития**

*ООО «Уральский центр систем безопасности»*

*(Россия, Екатеринбург)*

*doi: 10.18411/trnio-12-2024-883*

**Аннотация**

В статье анализируется текущее состояние и глобальное распределение крупных центров обработки данных (ЦОД) с учетом ведущих стран, таких как США, которые обладают значительным числом и мощностью ЦОД благодаря развитой ИТ-инфраструктуре. Также рассматриваются тенденции роста в других странах, включая Россию, которая стремится развивать собственную инфраструктуру ЦОД для повышения цифрового суверенитета. Обсуждаются ключевые факторы, влияющие на рост числа и мощности ЦОД, такие как увеличение объема данных, развитие искусственного интеллекта и сетевых технологий. Прогнозируется дальнейший рост числа и мощности ЦОД в странах, активно инвестирующих в развитие своей цифровой инфраструктуры.

**Ключевые слова:** центры обработки данных, ИТ-инфраструктура, глобальная цифровизация, распределение ЦОД, гипермасштабные центры обработки данных, мега ЦОД, ЦОД в России.

**Abstract**

The article analyzes the current state and global distribution of large data centers (DCs), taking into account leading countries like the United States, which holds a significant number and capacity of DCs due to its developed IT infrastructure. Growth trends are also examined in other countries, including Russia, which seeks to enhance its own DC infrastructure for digital sovereignty. Key factors influencing the increase in the number and capacity of DCs are discussed, such as data volume growth, the development of artificial intelligence, and network technologies. Further growth in the number and capacity of DCs is projected in countries actively investing in their digital infrastructure.

**Keywords:** data centers, IT infrastructure, global digitalization, data center distribution, hyperscale data centers, mega data centers, data centers in Russia.

Глобальный рынок центров обработки данных демонстрирует стремительный рост, обусловленный развитием цифровой экономики, ростом объемов данных и увеличением спроса на облачные технологии. В 2024 году в мире насчитывается около 8 тысяч крупных центров обработки данных (ЦОД), преимущественно размещенных в странах с высоким уровнем технологического развития, таких как США, Германия и Великобритания. Лидером остаются США, на долю которых приходится большая часть объектов ЦОД в мире.

Но тем не менее среди крупных стран игроков есть и Россия. В последние годы Россия активно развивает свою инфраструктуру центров обработки данных (ЦОД), стремясь повысить технологическую независимость и создать условия для локального хранения и обработки данных. В 2024 году в России насчитывается порядка 110 крупных ЦОД, что составляет около 1.15% от общего числа мировых объектов. Хотя этот показатель относительно невелик, Россия продолжает наращивать мощности, особенно в контексте усиления ограничений на импорт зарубежного оборудования и программного обеспечения.

География распределения ЦОД отражает различия в уровне развития технологий и потребности в ИТ-инфраструктуре. Основное количество крупных ЦОД сосредоточено в странах, где высока потребность в хранении и обработке данных. Данные Statista и Data Centre Magazine за 2024 год показывают следующее распределение, которое представлено в таблицах ниже.

Таблица 1

Количество крупных ЦОД и серверных помещений по странам мира.

Страна	Кол-во крупных ЦОД <sup>1</sup>	Доля от мирового числа крупных ЦОДов	Доля крупных ЦОД от общего числа ЦОД страны	Оценочное кол-во всех ЦОД и серверных помещений <sup>1</sup>	Доля от мирового числа всех ЦОД и серверных помещений <sup>3</sup>
США	5375	56,25%	0,11%	5000000 – 6000000	49,90%
Германия	522	5,46%	0,10%	500000 – 600000	0,15%
Великобритания	517	5,41%	0,13%	400000 – 500000	0,15%
Китай	448	4,69%	0,04%	1000000 – 1200000	0,20%
Канада	335	3,51%	0,11%	300000 – 400000	0,20%
Франция	314	3,29%	0,16%	200000 – 300000	0,20%
Австралия	306	3,20%	0,20%	150000 – 250000	0,30%
Япония	222	2,32%	0,11%	200000 – 300000	0,30%
Нидерланды	180	1,88%	0,09%	150000 – 200000	0,40%
Южная Корея	155	1,62%	0,10%	150000 – 200000	0,50%
Индия	130	1,36%	0,02%	500000 – 700000	0,70%
Бразилия	120	1,26%	0,05%	200000 – 300000	0,80%
<b>Россия</b>	<b>110</b>	<b>1,15%</b>	<b>0,04%</b>	<b>300000 – 500000</b>	<b>0,80%</b>
Испания	100	1,05%	0,07%	150000 – 250000	1,00%
Италия	95	0,99%	0,08%	150000 – 250000	1,00%
ЮАР	80	0,84%	0,05%	100000 – 150000	1,50%
Мексика	75	0,78%	0,07%	100000 – 150000	1,50%
ОАЭ	65	0,68%	0,06%	80000 – 120000	2,99%
Турция	60	0,63%	0,05%	80000 – 120000	2,00%
Швеция	55	0,58%	0,07%	70000 – 100000	4,99%
Швейцария	50	0,52%	0,04%	50000 – 80000	1,50%
Сингапур	45	0,47%	0,06%	40000 – 60000	1,50%
Финляндия	40	0,42%	0,05%	30000 – 50000	2,00%
Польша	35	0,37%	0,07%	30000 – 50000	1,50%
Австрия	30	0,31%	0,04%	20000 – 30000	2,00%
Аргентина	28	0,29%	0,06%	20000 – 30000	2,99%
Норвегия	25	0,26%	0,05%	20000 – 30000	9,98%
Малайзия	20	0,21%	0,05%	15000 – 25000	3,99%
Колумбия	18	0,19%	0,05%	15000 – 25000	4,99%
<b>Итого</b>	<b>9555</b>	<b>100%</b>		<b>10 020 000 - 12 970 000</b>	<b>100%</b>

Примечание: Данные приведены в приблизительных значениях. Критерии отнесения объекту крупный ЦОД приведены в тексте ниже статьи. Данные приведены по минимальным значениям.

Данные основаны на текущих оценках и могут незначительно отличаться в зависимости от методологии подсчета объектов в разных странах.

Тем не менее эти данные демонстрируют доминирующее положение США, которые обладают значительно большим числом ЦОД, что связано с высокой концентрацией технологических компаний и гипермасштабных облачных провайдеров, таких как Amazon, Google и Microsoft.

По географии распределения крупные ЦОД показаны на рисунке ниже.



Рисунок 1. География распределения крупных ЦОД по части стран.

### Характеристики крупных ЦОД

Крупные ЦОД выделяются своими размерами, мощностью и требованиями к инфраструктуре. Характерные особенности крупных ЦОД приведены ниже.

Для классификации таких ЦОД обычно используются следующие ориентировочные показатели:

1. Количество стоек:
  - Крупный ЦОД, как правило, включает от 500 стоек и более. В гипермасштабных ЦОД это число может достигать до нескольких тысяч стоек.
2. Площадь:
  - Площадь крупных ЦОД составляет от 5000 квадратных метров и может достигать 100 000 квадратных метров в гипермасштабных объектах.
2. Мощность:
  - Энергопотребление крупных ЦОД начинается от 10 мегаватт (МВт) и может превышать 100 МВт в крупнейших объектах, обслуживающих облачные и крупные коммерческие сервисы.

Эти значения варьируются в зависимости от классификации и стандартов (например, Uptime Institute, ANSI/BICSI).

Большинство крупных ЦОДов оснащены многоуровневыми системами защиты и резервирования для обеспечения непрерывности работы, что особенно важно для гипермасштабных центров, обслуживающих крупные компании и облачные сервисы.

Совокупная мощность крупных ЦОД оценивается в 18 228 МВт, где основная доля приходится на США, Китай и Великобританию.

Таблица 2

Суммарная мощность крупных ЦОД и серверных помещений по странам мира.

Страна / Регион	Мощность крупных ЦОД, МВт	Доля от общей мощности
США	6994	38,37%
Китай	2524	13,85%
Великобритания	1053	5,78%
Сингапур	876	4,81%
Япония	865	4,75%
Германия	864	4,74%
Австралия	667	3,66%
Нидерланды	531	2,91%
<b>Россия</b>	<b>460</b>	<b>2,52%</b>
Гонконг	417	2,29%
Франция	391	2,14%
Южная Корея	345	1,89%

Страна / Регион	Мощность крупных ЦОД, МВт	Доля от общей мощности
Индия	310	1,70%
Канада	267	1,46%
Бразилия	200	1,10%
Испания	180	0,99%
Италия	165	0,91%
ЮАР	150	0,82%
Швеция	140	0,77%
Швейцария	130	0,71%
Мексика	125	0,69%
Турция	110	0,60%
Финляндия	95	0,52%
ОАЭ	90	0,49%
Польша	85	0,47%
Норвегия	70	0,38%
Малайзия	65	0,36%
Колумбия	60	0,33%
<b>Итого</b>	<b>18 228,63</b>	<b>100%</b>

Дополнительно стоит отметить, что доли по мощности основаны на оценке средней мощности и могут меняться в зависимости от обновлений данных по инфраструктуре.

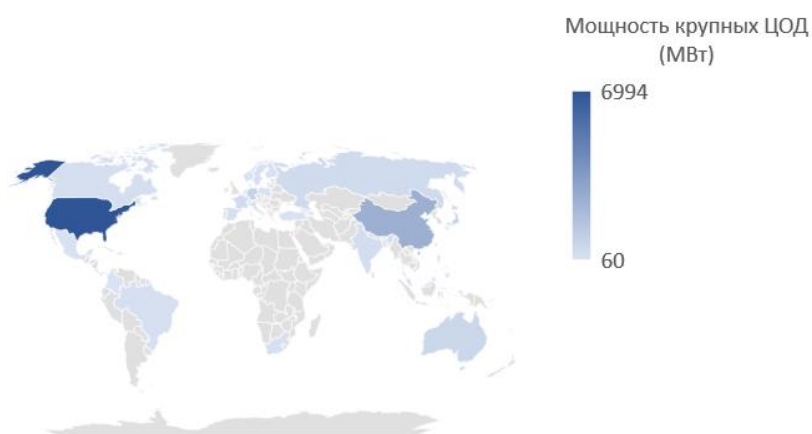


Рисунок 2. География распределения крупных ЦОД по части стран.

Кроме крупных центров, существует значительное число небольших серверных помещений и малые ЦОД, используемых малым и средним бизнесом для локального хранения данных. Крупные, малые ЦОД и серверные помещения можно отнести к ИТ-объектам (объекты информационных технологий). Подсчитать точное количество таких объектов сложно, но по оценкам, в мире около 213 миллионов МСП (малые и средние предприятия), и если 5% из них имеют свои серверные, это составит примерно 10–13 миллионов объектов.

Как правило в категорию МСП входят компании с небольшим числом работников и относительно ограниченным объемом доходов.

Критерии отнесения к категории МСП могут варьироваться в зависимости от страны. Например, в России к малым предприятиям относят компании с числом сотрудников до 100 человек и доходом до 800 млн рублей в год, а к средним — с числом сотрудников до 250 человек и доходом до 2 млрд рублей в год. В других странах могут использоваться разные числовые и финансовые пороги для определения категории МСП.

Собственно, исходя из этих различных критерий точно подсчитать количество малых ЦОД и серверных весьма сложно.

Таблица 3

Типы ИТ-объектов, количество и их типовая роль.

Тип ИТ-объекта	Оценочное количество в мире, шт	Типовая роль ИТ-объекта
Крупные ЦОД	0.8 млн.	Обеспечение глобальной инфраструктуры, поддержка облачных провайдеров.
Малые ЦОД, серверные помещения	10–13 млн.	Локальное хранение и обработка данных, поддержка малого и среднего бизнеса.

**Основные тенденции, влияющие на рост числа крупных ЦОД в мире, включают:**

1. Рост объема данных: По прогнозам IDC, объем мировых данных к 2025 году достигнет 175 зеттабайт, что требует значительных мощностей для хранения и обработки.
2. Развитие технологий искусственного интеллекта и Интернета вещей: Системы ИИ и IoT генерируют большие объемы данных, которые требуют надежной инфраструктуры для анализа и хранения.
3. Глобальная цифровизация: Пандемия COVID-19 ускорила переход к облачным решениям, увеличив спрос на инфраструктуру ЦОД.
4. Геополитический фактор: в необходимости увеличения количества ЦОД также связана с обратным процессом деглобализации. Ряд стран, в число которых входит Россия и Китай, стремятся повысить свой суверенитет в условиях нестабильной геополитической обстановки и обрести свои собственные цифровые сервисы тем самым значительно снижая зависимость от ведущих стран в области цифровизации.

В условиях "информационного взрыва" роль ЦОД объектов становится критически важной для глобальной ИТ-инфраструктуры. С развитием технологий и увеличением объема данных спрос на высокопроизводительные и энергоэффективные ЦОД объекты продолжает расти. Ожидается, что ближайшие годы станут периодом активной модернизации и расширения мощностей, особенно в странах с высоким спросом на облачные и аналитические услуги.

\*\*\*

1. Statista. "Data centers worldwide by country 2024".
2. Data Centre Magazine, "Top 10: Countries with the Most Data Centres", 2024.
3. Reinsel, D., Gantz, J., & Rydning, J. (2020). The Digitization of the World: From Edge to Core. IDC. "Пандемия ускорила развитие рынка больших данных" (2021). Российская газета, 26 января.

**Орехов В.В., Лытнев Н.Н.**

**Обзор технологий и сервисов для разработки и запуска облачных игр**

*Кубанский государственный аграрный университет  
(Россия, Краснодар)*

doi: 10.18411/trnio-12-2024-884

#### **Аннотация**

В статье рассматриваются ключевые технологии и сервисы для разработки и запуска облачных игр, включая международные и отечественные решения. Облачные игры становятся всё более популярными благодаря своей доступности на устройствах с минимальными техническими характеристиками. Особое внимание уделено платформам

Google Stadia, NVIDIA GeForce Now, Xbox Cloud Gaming, а также российским проектам, таким как SberPlay, Playkey и FogPlay. Рассматриваются преимущества, вызовы и перспективы облачного гейминга с учётом глобальных и локальных факторов.

**Ключевые слова:** облачные игры, технологии облачного гейминга, игровые платформы, потоковая передача, разработка игр.

### Abstract

This article explores key technologies and services for developing and deploying cloud games, including international and domestic solutions. Cloud gaming is gaining popularity due to its accessibility on devices with minimal technical specifications. Special attention is given to platforms like Google Stadia, NVIDIA GeForce Now, Xbox Cloud Gaming, and Russian projects such as SberPlay, Playkey, and FogPlay. The article examines the advantages, challenges, and prospects of cloud gaming in the context of global and local factors.

**Keywords:** cloud gaming, cloud gaming technologies, gaming platforms, streaming, game development.

Возможность насладиться любой игрой, не имея при этом дорогого оборудования является значительным шагом вперед в игровой индустрии, представленным в виде облачных игр и связанных с ними сервисов. Облачные сервисы – это услуга по предоставлению вычислительных мощностей удаленно от пользователя так, что потребитель хранит и обрабатывает информацию не на своем оборудовании, а дистанционно [1]. Техническому отделу сервиса требуется произвести обновление лишь одного экземпляра программы, вместо нескольких десятков или сотен [2] со стороны пользователей. В силу своей специфики облачные сервисы призваны сократить расходы на ИТ инфраструктуру, а по мере развития технологий, доступность будет лишь повышаться.

Среди международных сервисов лидирующие позиции занимают G NVIDIA GeForce Now и Xbox Cloud Gaming. Nvidia GeForce Now не привязывает пользователя к какой-то конкретной игровой платформе с доступными на ней играми и приложениями, вместо этого позволяя пользователю запускать их из сторонних сервисов, где они уже были куплены. Xbox Cloud Gaming имеет интеграцию с Xbox Game Pass, что дает доступ к обширной библиотеке игр, входящих в подписку. Кроме вышеперечисленных также существовала платформа Google Stadia, которая предлагала мощную серверную инфраструктуру и инструменты, однако крайне сильно ограничивала пользователя в доступной библиотеке специально адаптированных игр, что привело к малой ее популярности и в итоге к закрытию.

Однако, начиная с 2022 года, в связи с неблагоприятными отношениями России с другими странами российские организации и пользователи постепенно стали переходить на отечественные облачные сервисы. Введение санкций США и другими странами послужило огромным стимулом к развитию отечественных продуктов [3]. Среди них выделяются SberPlay, Playkey и FogPlay. SberPlay использует мощности СберКлауда и предлагает интеграцию с экосистемой Сбера. Playkey, будучи одной из первых российских платформ облачного гейминга, использует распределённые серверы по всей стране, минимизируя задержки и обеспечивая стабильное подключение.

FogPlay является одним из наиболее интересных примеров отечественного подхода к облачному геймингу. Платформа акцентирует внимание на качестве сервиса и гибкости. Она предоставляет пользователям возможность оплаты времени, проведенного в играх. FogPlay активно сотрудничает с локальными интернет-провайдерами, что помогает минимизировать задержки в передаче данных. Технологическая основа платформы включает поддержку современных стандартов кодирования видео, таких как H.265, и использование серверов в крупных дата-центрах России, что обеспечивает высокую производительность и стабильность работы.

Крепкую позицию занимает также VK Play Cloud, принадлежащий корпорации VK. Они предоставляют либо вариант с почасовой оплатой игр на мощном удаленном устройстве, либо приобретение подписки, которая оплачивается раз в месяц и дает безлимит на время в игре или ограниченное количество часов, с безлимитом в ночное время. Как и Geforce Now, VK Cloud позволяет запускать игры из сторонних библиотек пользователей, что делает его крайне привлекательным для владельцев обширных

библиотек, однако за это приходится платить более высокой платой, нежели в том же FogPlay. VK Cloud является частью более широкого сервиса VK Play, который представляет собой магазин игр, из которого в том числе можно запускать игры в облаке, кроме того под их началом проводятся турниры по различным дисциплинам, спонсируются разработки конкурентноспособных отечественных продуктов, которые позже также могут появиться в облаке.

Главным привлекательным фактором облачных игр является их доступность для пользователей с устаревшим оборудованием, но достаточно хорошим интернетом, мгновенном запуске, без необходимости скачивания и установки, что экономит как время, так и память устройства

Несмотря на все плюсы, облачные сервисы все еще сильно зависят от качества интернет-соединения, что делает их неэффективными в регионах с малой скоростью интернета. Для российских платформ главным вызовом все еще является конкуренция с международными сервисами, что, однако, способствует их развитию и предоставлению пользователям более удобных и специфичных для страны сервисов.

Будущее облачных игр связано с развитием технологий 5G, улучшением алгоритмов сжатия данных и увеличением доступности высокоскоростного интернета. Российские проекты, такие как FogPlay, могут сыграть важную роль в этом процессе, предлагая адаптированные решения для местных пользователей и способствуя развитию игровой индустрии.

В заключение можно сказать, что облачные игры — это перспективное направление, открывающее новые возможности для пользователей и разработчиков. Благодаря широкой доступности и гибким тарифам позволить себе мощные вычислительные мощности может практически любой человек или студия, давая больше пространства и ресурсов для работы над качеством самого проекта[4]. Международные платформы продолжают развивать технологическую базу, а российские проекты, такие как SberPlay, Playkey и FogPlay, перенимают мировые тренды и демонстрируют конкурентоспособность на локальном рынке, применяя самые современные технологии. Однако для ускорения роста необходимо способствовать распространению высокоскоростного интернет-соединения, лицензировать контент и адаптировать игры к различным инфраструктурам.

\*\*\*

1. Дышкант, С. С. Перспективы облачных вычислений на российском рынке / С. С. Дышкант, А. М. Кумратова, К. Д. Лещенко // Мировые тенденции развития науки и техники: пути совершенствования : Материалы X Международной научно-практической конференции. В 3-х частях, Москва, 29 декабря 2022 года / Автономная некоммерческая организация «Национальный исследовательский институт дополнительного профессионального образования» (АНО «НИИ ДПО»). Том Часть 1. – Москва: Общество с ограниченной ответственностью "Пресс-центр", 2022. – С. 42-45.
2. Гайфуллин, А. О. Технология разделения данных в облачных сервисах / А. О. Гайфуллин, Д. Н. Савинская, К. Д. Лещенко // Вопросы современной науки: проблемы, тенденции и перспективы : Материалы VII международной научно-практической конференции, приуроченной к Году педагога и наставника, Новокузнецк, 08 декабря 2023 года. – Кемерово: Кузбасский государственный технический университет имени Т.Ф. Горбачева, 2023. – С. 248-250.
3. Романюк, А. С. Облачная миграция в России: с западного на отечественный. Развитие отечественных облачных технологий / А. С. Романюк, С. М. Самойлик, Д. Н. Савинская // Виртуозы науки : Сборник тезисов Международной научно-практической конференции студентов и молодых учёных за 2023 г, Краснодар, 06–15 ноября 2023 года. – Краснодар: Кубанский государственный аграрный университет им. И.Т. Трубилина, 2024. – С. 509-511.
4. Егоров, И. А. Применение технологий облачных вычислений в области 3D графики / И. А. Егоров, Д. Н. Савинская, К. Д. Лещенко // Современные тенденции развития фундаментальных и прикладных наук : Материалы VII Всероссийской научно-практической конференции, Брянск, 25 января 2024 года. – Брянск: Брянский государственный инженерно-технологический университет, 2024. – С. 269-272.

Падаев Р.Р.

**Выбор сетевых протоколов для эффективного взаимодействия в Интернет вещей***Санкт-Петербургский государственный университет телекоммуникаций**им. проф. М.А. Бонч-Бруевича**(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-885

**Аннотация**

В статье рассматриваются особенности применения сетевых протоколов в системах Интернета вещей (IoT). Проанализированы уровни TCP/IP, подходящие технологии для IoT, включая IPv6, Bluetooth Low Energy, LoRaWAN, а также протоколы прикладного уровня HTTP и MQTT. Особое внимание уделено сравнению протоколов с точки зрения энергоэффективности, безопасности и производительности для различных IoT-приложений.

**Ключевые слова:** интернет вещей, IoT, сетевые протоколы, MQTT, HTTP, LoRaWAN.

**Abstract**

The article discusses the features of the application of network protocols in Internet of Things (IoT) systems. TCP/IP layers, suitable technologies for IoT, including IPv6, Bluetooth Low Energy, LoRaWAN, as well as HTTP and MQTT application layer protocols are analyzed. Special attention is paid to the comparison of protocols in terms of energy efficiency, security and performance for various IoT applications.

**Keywords:** internet of Things, IoT, network protocols, MQTT, HTTP, LoRaWAN.

Интернет вещей (IoT, Internet of Things) будет построен на базе существующей сетевой инфраструктуры, технологий и протоколов, используемых в настоящее время в домах/офисах и в Интернете, и предложит много другого. Сети IoT будут функционировать на базе существующей четырехуровневой модели TCP/IP, где на каждом уровне применяются определенные протоколы. Диаграмма, изображенная на рисунке 1, ниже показывает сравнение текущих протоколов с теми, которые, вероятно, будут использоваться для IoT.

Current Internet Protocols		Expected IOT Protocols
HTTP FTP,SMTP,IMAP	Application	MQTT COAP,AMQP
TCP and UDP	Transport	UDP and TCP
IPv4 and IPv6	Networking	IPv6 and IPv4
Ethernet,Wi-Fi, GSM	Data Link	Ethernet,Wi-Fi, GSM, LTE-M, Lora, SigFox
Protocol Level TCP/IP Model		

IOT and Internet Protocols

Рисунок 1. Сравнение текущих интернет-протоколов и ожидаемых протоколов для IoT.

Размер шрифта отражает популярность протокола. Например, слева шрифт для IPv4 больше, так как этот протокол широко используется в современном Интернете. Справа его размер уменьшен, так как в IoT ожидается большее распространение IPv6. Диаграмма не включает все существующие протоколы. Наибольшие изменения ожидаются на канальном (уровни 1 и 2) и прикладном уровнях (уровень 4), тогда как сетевой и транспортный уровни, вероятно, останутся почти без изменений.

На канальном уровне (Data Link) вам нужно соединить устройства между собой. Они могут находиться как недалеко, например, в локальных сетях (local networks) так и на большом расстоянии друг от друга: в городских (metropolitan area networks) и глобальных сетях (wide area networks).

В настоящее время на этом уровне в домашних и офисных сетях (LAN) используются Ethernet и Wi-Fi, а в мобильных (WAN) — 3G / 4G. Однако многие IoT-устройства маломощные, например, датчики, и питаются только от батарей. В этих случаях Ethernet не подходит, но можно использовать low powered Wi-Fi и low powered Bluetooth.

Хотя для подключения этих устройств по-прежнему будут использоваться существующие беспроводные технологии, такие как Wi-Fi, Bluetooth и 3G/4G, стоит обратить внимание на новые технологии, специально разработанные для IoT-приложений, которые, вероятно, будут набирать популярность. К таким технологиям относятся Bluetooth Low Energy (BLE), LoRaWAN (Long Range WAN), SigFox и LTE-M.

На сетевом уровне (Networking) в долгосрочной перспективе будет доминировать протокол IPv6. Маловероятно, что будет использоваться IPv4, но он может играть определенную роль на начальных этапах. Большинство IoT-устройств для дома, например, умные лампочки, в настоящее время используют IPv4.

На транспортном уровне (Transport) в Интернете и вебе доминирует TCP. Он используется как в HTTP, так и во многих других популярных протоколах Интернета (SMTP, POP3, IMAP4 и т. д.). MQTT, который, как я ожидаю, станет одним из основных протоколов прикладного уровня для обмена сообщениями, в настоящее время использует TCP. Однако в будущем из-за более низких накладных расходов, я ожидаю, что UDP будет более популярен для IoT. Вероятно, более широкое распространение получит MQTT-SN, работающий поверх UDP.

Для протоколов IoT важны такие характеристики, как скорость передачи данных, задержка (время, необходимое для доставки сообщения), потребляемая мощность, безопасность и доступность программных средств.

На этом уровне сегодня активно используются два основных протокола: HTTP и MQTT.

HTTP — один из самых известных протоколов прикладного уровня, лежащий в основе веба (WWW). Он продолжит играть важную роль в IoT, особенно для REST API, которые обеспечивают взаимодействие веб-приложений и сервисов. Однако из-за высоких накладных расходов HTTP вряд ли станет основным протоколом для IoT, хотя все равно останется востребованным в Интернете.

MQTT (Message Queuing Telemetry Transport), напротив, приобрел статус основного протокола обмена сообщениями в IoT благодаря своей легкости и простоте использования.

MQTT быстро становится стандартом де-факто для IoT-приложений. Это происходит из-за его легкости и скорости по сравнению с HTTP и того, что он является протоколом "один ко многим", а не "один к одному" (HTTP).

Многие современные веб-приложения, с радостью использовали бы MQTT вместо HTTP, если бы он был доступен на момент их разработки.

Хороший пример — отправка информации множеству клиентов, например, о прибытии и отправлении поездов / автобусов / самолетов. В этом сценарии протокол "один-к-одному", такой как HTTP, имеет большие накладные расходы и создает большую нагрузку на веб-серверы. Масштабирование этих веб-серверов может быть затруднено. С

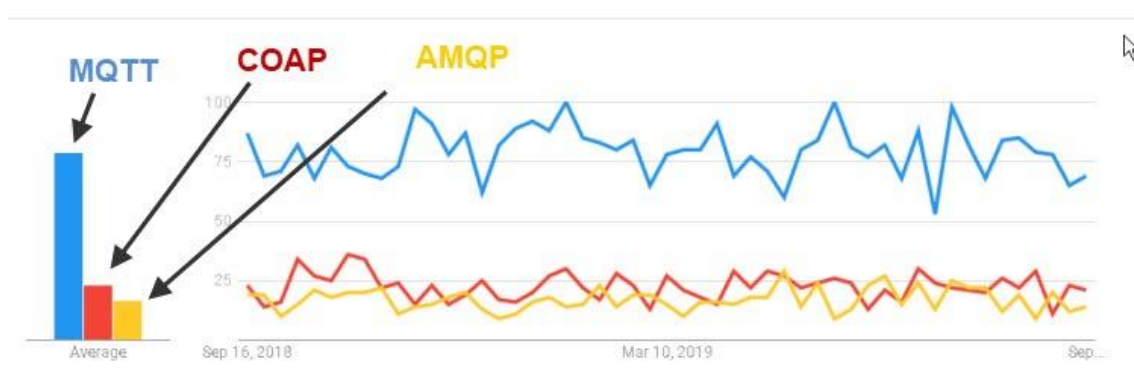
MQTT клиенты подключаются к брокеру, которые легко можно добавлять для балансировки нагрузки.

HTTP не был разработан для IoT-приложений, но, как уже упоминалось, он будет широко использоваться в течение какого-то времени благодаря его широкому использованию в API. Почти все платформы IoT поддерживают как HTTP, так и MQTT.

В первые годы существования Интернета протокол HTTP, который впоследствии стал доминирующим, был лишь одним из множества протоколов. Хотя HTTP изначально не был разработан для передачи файлов и электронной почты, сегодня он используется и для этих целей.

Вероятно, в IoT с протоколами обмена сообщениями произойдет нечто похожее: большинство сервисов будут использовать один преобладающий протокол.

Ниже представлен рисунок, 2 на котором изображены графики Google Trends, демонстрирующие, как менялась популярность протоколов MQTT, COAP и AMQP в последние годы.



## Google Trends MQTT, COAP and AMQP

Рисунок 2. Популярность протоколов MQTT, COAP и AMQP по данным Google Trends.

Для успешного внедрения IoT-систем важно учитывать, какие протоколы обмена данными поддерживаются различными облачными платформами. Большинство платформ поддерживают протокол MQTT, что подчеркивает его популярность и актуальность для IoT-приложений. Протокол HTTPS также широко распространен, поскольку обеспечивает высокий уровень безопасности. Некоторые платформы, такие как Microsoft Azure и Thingworx, поддерживают AMQP, что может быть полезно для специфических сценариев передачи данных.

\*\*\*

1. Волкогонов В. Н. и др. Применение физически неклонированных функций для выполнения аутентификации в среде интернета вещей // Актуальные проблемы инфотелекоммуникаций в науке и образовании. – 2021. – С. 409-414.
2. Миняев А. А. Метод оценки эффективности системы защиты информации территориально-распределенных информационных систем персональных данных // (АПИНО 2020). – 2020. – С. 716-719.
3. Петрова Т. В. и др. Подходы обнаружения беспроводной точки доступа злоумышленника в локальной вычислительной сети // Региональная информатика (РИ-2022). – 2022. – С. 572-573.
4. Абрамова Е. А., Красов А. В., Поляничева А. В. Тенденции развития и безопасность IP-телефонии // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). – 2023. – С. 23-28.

Панова Ю.А.

**Интеграция ЕПП и ALD с PostgreSQL для управления доступом и безопасности в корпоративной сети**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-886

**Аннотация**

В статье рассматривается интеграция Единого пространства пользователей (ЕПП) и Astra Linux Directory (ALD) с PostgreSQL для централизованного управления учетными записями и обеспечения безопасности данных. Описаны архитектура ALD, применение протоколов для защиты, а также основные этапы настройки и оптимизации взаимодействия с PostgreSQL.

**Ключевые слова:** Astra Linux, Единое пространство пользователей, ALD, PostgreSQL, управление доступом, безопасность данных.

**Abstract**

The article discusses the integration of the Unified User Space (EPP) and Astra Linux Directory (ALD) with PostgreSQL for centralized account management and data security. The ALD architecture, the use of protocols for protection, as well as the main stages of configuring and optimizing interaction with PostgreSQL are described.

**Keywords:** Astra Linux, Unified User Space, ALD, PostgreSQL, access control, data security.

Единое пространство пользователей (ЕПП) и Astra Linux Directory (ALD) являются ключевыми компонентами системы управления пользователями в операционной системе Astra Linux. Их совместная работа обеспечивает централизованное управление учетными записями, упрощает процессы аутентификации и авторизации, а также повышает общую безопасность информационной инфраструктуры.

ЕПП представляет собой концепцию, направленную на создание единой среды для управления учетными записями и их атрибутами. В рамках ЕПП пользователи могут использовать одну учетную запись для доступа к различным сервисам и приложениям внутри корпоративной сети. Это значительно сокращает время на настройку и администрирование прав доступа, а также минимизирует ошибки, возникающие при использовании нескольких учетных записей для одного пользователя. В Astra Linux ЕПП служит базой для консолидации всех пользователей и ресурсов в одной точке, что особенно важно для крупных корпоративных систем, требующих строгого контроля за доступом к данным и ресурсам.

Astra Linux Directory (ALD) — это специализированная служба, разработанная для управления ЕПП в среде Astra Linux. ALD является надстройкой над технологиями LDAP, Kerberos 5 и CIFS, обеспечивая автоматическую настройку всех необходимых файлов конфигурации служб, реализующих перечисленные технологии, а также предоставляя интерфейс управления и администрирования. Основные компоненты ALD включают клиентскую часть (ald-client), утилиту администрирования (ald-admin) и серверную часть (ald-server). Клиентская часть устанавливается на все клиентские компьютеры, входящие в ЕПП, утилита администрирования — на компьютеры, с которых

осуществляется администрирование базы данных ALD, а серверная часть — на сервер домена.

Чтобы понять, как Единое пространство пользователей (ЕПП) и Astra Linux Directory (ALD) работают вместе, важно рассмотреть архитектуру ALD, его основные компоненты и протоколы, которые поддерживают управление пользователями. В Astra Linux ALD является фундаментальным компонентом, который выполняет функции каталога и хранения данных о пользователях, необходимых для централизованного управления учетными записями и доступом. Наглядная информация об основных компонентах ALD представлена на рисунке 1 ниже.



Рисунок 1. Основные компоненты ALD.

Архитектура ALD строится на трех основных компонентах, которые изображены на рисунке выше:

1. ALD Сервер (ald-server) — серверная часть, расположенная на сервере домена. Этот компонент отвечает за управление базой данных пользователей и их правами доступа. ALD Сервер хранит учетные записи и их атрибуты, обрабатывает запросы на аутентификацию и авторизацию, обеспечивая связь с клиентскими устройствами.
2. ALD Администратор (ald-admin) — утилита для администрирования, установленная на компьютерах, с которых администраторы управляют базой данных ALD. Она предоставляет интерфейс для выполнения административных задач, таких как добавление и удаление учетных записей, настройка прав доступа и управление группами пользователей.
3. ALD Клиент (ald-client) — клиентская часть, установленная на всех устройствах, которые должны входить в домен ЕПП. ALD Клиент взаимодействует с ALD Сервером, передавая запросы на аутентификацию и проверку прав доступа, а также обеспечивая пользователям доступ к ресурсам сети в зависимости от их прав.

Для работы ЕПП в Astra Linux используются различные протоколы, обеспечивающие защиту данных и безопасный доступ к ресурсам. В основе ALD лежат такие протоколы, как LDAP (Lightweight Directory Access Protocol), Kerberos и CIFS, каждый из которых выполняет свою роль в процессе аутентификации и авторизации пользователей, что наглядно отображено на рисунке 2.

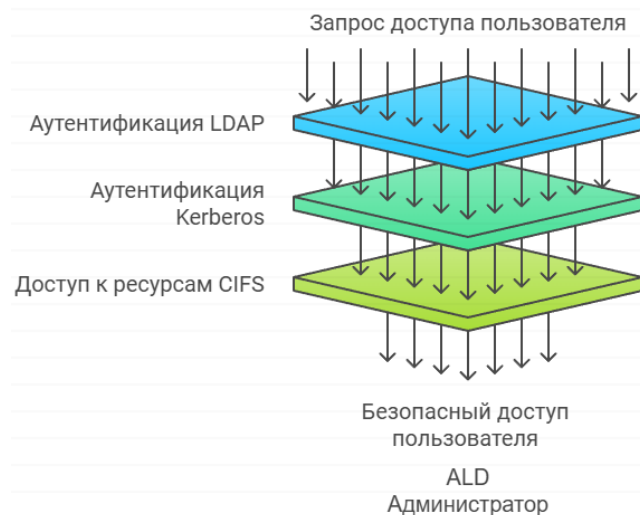


Рисунок 2. Процесс многоуровневой аутентификации и защиты данных в ALD.

LDAP — отвечает за структуру каталога и управление учетными данными. Этот протокол позволяет централизованно хранить данные о пользователях, их атрибутах и правах доступа. Когда пользователь отправляет запрос на доступ, система сначала обращается к LDAP для проверки данных учетной записи и первичной аутентификации.

Kerberos — используется для многофакторной аутентификации и подтверждения личности пользователя. После начальной проверки учетной записи через LDAP, Kerberos выполняет второй этап аутентификации, защищая данные от несанкционированного доступа и обеспечивая безопасность при передаче данных.

CIFS — протокол, который обеспечивает доступ к файлам и другим ресурсам в сети. После успешной аутентификации через LDAP и Kerberos пользователь получает доступ к необходимым ресурсам с помощью CIFS, что позволяет обмениваться данными между устройствами в рамках единой сети.

Данные протоколы интегрированы в Astra Linux и автоматически настраиваются при конфигурации ALD, что значительно упрощает работу администраторов. При каждом запросе пользователя ALD проходит через все уровни защиты, представленные на рисунке, обеспечивая тем самым многоуровневую проверку и безопасный доступ к корпоративным ресурсам.

ALD интегрируется в Astra Linux для централизованного управления учетными записями и безопасного доступа к ресурсам. Настройка начинается с серверной части: задается статический IP-адрес и синхронизация времени через NTP, что важно для корректной работы протоколов аутентификации, таких как Kerberos. Затем конфигурируется файл `/etc/hosts`, который заменяет DNS-сервер, позволяя всем устройствам домена находить друг друга по именам. Для этого файл должен быть идентичен на всех машинах, указав IP-адреса и имена устройств, например:

```
192.168.1.1 dc1.local dc1
```

Установка ALD включает пакеты `ald-server-common`, `fly-admin-ald-server` и `smolensk-security-ald`, после чего сервер инициализируется командой:

```
ald-init init
```

После инициализации необходимо перезагрузить сервер. В процессе инициализации домена еще раз будет запрос на создание пароля администратора домена, а также запрос на создание пароля на базу данных kerberos. Если все прошло удачно, то на экране появится сообщение как на рисунке 3.

```
Перезапуск сервиса postgresql...
Перезапуск сервиса aldd...

Astra Linux Directory сконфигурирована.
Сервер ALD активен.
Клиент ALD Включен.

Astra Linux Directory сервер успешно инициализирован.
root@dc1:/etc/aldd#
```

Рисунок 3. Успешная инициализация сервера ALD.

После настройки ALD в Astra Linux следующим важным шагом является интеграция с системой управления базами данных PostgreSQL. Эта интеграция обеспечивает надежное хранение и управление данными пользователей, необходимыми для аутентификации и авторизации в домене ALD. Понимание процесса интеграции позволяет обеспечить безопасность и эффективность работы системы.

В архитектуре ALD PostgreSQL выполняет функцию хранилища данных, где сохраняются учетные записи пользователей, их атрибуты и права доступа. Использование PostgreSQL обусловлено его надежностью, масштабируемостью и поддержкой необходимых функций для безопасного хранения данных. ALD взаимодействует с PostgreSQL для выполнения операций чтения и записи данных, обеспечивая централизованное управление пользователями в домене.

Интеграция ALD с PostgreSQL включает несколько этапов, каждый из которых важен для обеспечения корректной работы системы:

1. Добавление сервиса PostgreSQL в ALD.

Для начала необходимо зарегистрировать сервис PostgreSQL в домене ALD. Это позволяет ALD управлять аутентификацией и авторизацией для PostgreSQL. Команда `ald-admin service-add` добавляет сервис с указанием имени хоста:

```
ald-admin service-add postgres/${hostname}
```

Здесь `$(hostname)` возвращает имя текущего хоста, на котором установлен PostgreSQL. Эта команда регистрирует сервис PostgreSQL в базе данных ALD, позволяя системе управлять его учетными данными.

2. Назначение сервиса в группу безопасности.

После добавления сервиса необходимо назначить его в соответствующую группу безопасности. Это определяет уровень доступа и привилегий для сервиса. Команда `ald-admin sgroup-svc-add` добавляет сервис в группу `mac`:

```
ald-admin sgroup-svc-add postgres/${hostname} --sgroup=mac
```

Группа `mac` используется для управления мандатным контролем доступа, обеспечивая соответствие политики безопасности.

3. Обновление ключей сервиса.

Для обеспечения безопасного взаимодействия между ALD и PostgreSQL необходимо обновить ключи сервиса. Команда `ald-client update-svc-keytab` создает файл ключей (`keytab`) для сервиса:

```
keytab="/etc/postgresql/11/main/keytab"
ald-client update-svc-keytab postgres/${hostname} --ktfile="$keytab"
```

Файл `keytab` содержит ключи, используемые для аутентификации сервиса в домене ALD. После создания файла необходимо установить правильные права доступа:

```
chown postgres:postgres "$keytab"
```

Это гарантирует, что только пользователь postgres имеет доступ к файлу ключей, обеспечивая безопасность.

4. Настройка конфигурационных файлов PostgreSQL.

Для интеграции с ALD необходимо внести изменения в конфигурационные файлы PostgreSQL.

- postgresql.conf: Укажите путь к файлу ключей:

```
krb_server_keyfile = '/etc/postgresql/11/main/keytab'
```

Это позволяет PostgreSQL использовать указанный файл для аутентификации через Kerberos.

- pg\_hba.conf: Настройте метод аутентификации для клиентов:

```
host all all <адрес> gss include_realm=0
```

Здесь <адрес> замените на соответствующий IP-адрес или диапазон адресов клиентов. Метод gss указывает на использование Kerberos для аутентификации. Параметр include\_realm=0 исключает домен из имени пользователя при аутентификации.

5. Перезапуск сервиса PostgreSQL.

После внесения изменений в конфигурационные файлы необходимо перезапустить сервис PostgreSQL для применения настроек:

```
systemctl restart postgresql
```

Это обеспечивает загрузку новых настроек и готовность сервиса к работе в домене ALD.

После завершения интеграции ALD и PostgreSQL взаимодействие между этими компонентами обеспечивает централизованное и безопасное управление доступом к базам данных. Когда пользователь пытается подключиться к PostgreSQL, процесс начинается с аутентификации, которая проходит через протокол Kerberos. PostgreSQL передает запрос на аутентификацию в ALD, который выполняет проверку учетных данных пользователя. ALD обращается к своей базе данных, где хранятся данные о пользователях, и подтверждает личность, используя заранее созданные ключи и параметры безопасности.

Если учетные данные пользователя успешно подтверждены, ALD возвращает результат аутентификации PostgreSQL. На этом этапе, помимо аутентификации, выполняется еще и авторизация доступа. PostgreSQL, получив подтверждение от ALD, проверяет права пользователя, опираясь на информацию, предоставленную ALD. Это позволяет системе оценить, какие действия пользователь может выполнять в базе данных, и какие ресурсы ему доступны.

Таким образом, ALD выполняет две ключевые функции: подтверждает личность пользователя и предоставляет информацию о его правах. Этот подход обеспечивает надежный и централизованный контроль над доступом, соответствующий политике безопасности организации. В результате взаимодействие ALD и PostgreSQL позволяет унифицировать управление доступом и повысить безопасность корпоративных данных, исключая необходимость дублирования учетных записей и их разрозненного администрирования.

После изучения основ взаимодействия ALD и PostgreSQL, мы можем перейти к практической части — настройке ALD для работы с PostgreSQL. Этот процесс требует настройки обеих систем, чтобы обеспечить корректное взаимодействие, а также выполнить инициализацию PostgreSQL для хранения данных ALD. Следующие шаги помогут выполнить эту настройку, гарантируя, что оба компонента работают в унисон и безопасно обрабатывают данные пользователей.

Шаг 1: Инициализация PostgreSQL для работы с ALD. Прежде, чем ALD сможет использовать PostgreSQL для хранения данных, необходимо создать специальные роли и назначить привилегии, обеспечивая тем самым доступ ALD к базе данных.

1. Необходимо создать роль PostgreSQL, которая будет выполнять операции от имени ALD:

Эта команда создает пользователя `ald_user` с паролем, который ALD будет использовать для подключения к PostgreSQL. Рекомендуется использовать сложный пароль, поскольку он отвечает за защиту данных ALD.

2. Затем нужно задать привилегии для роли `ald_user`, чтобы дать ей права на создание базы данных и управление схемами данных:

```
ALTER ROLE ald_user CREATEDB;
```

Привилегия `CREATEDB` позволяет роли `ald_user` управлять базой данных, что критически важно для хранения пользовательских данных и поддержки структуры базы для ALD.

Шаг 2: Настройка ALD для подключения к PostgreSQL. После создания необходимых ролей следует настроить ALD для подключения к PostgreSQL с нужными параметрами. Для этого нужно внести изменения в конфигурационные файлы ALD, указав параметры подключения.

1. Необходимо открыть файл конфигурации ALD (например, `/etc/ald/ald.conf`) и указать данные для подключения к PostgreSQL, включая имя пользователя, пароль, хост и порт:

```
db_user = ald_user
db_password = strong_password
db_host = 127.0.0.1
db_port = 5432
```

Здесь `db_user` и `db_password` соответствуют учетным данным, созданным на первом этапе. Поле `db_host` указывает на локальный сервер (если PostgreSQL установлен на том же сервере, что и ALD), а `db_port` — на порт, по которому PostgreSQL принимает подключения.

2. Для применения новых параметров подключения необходимо перезагрузить ALD:

```
systemctl restart ald
```

Это действие позволяет ALD загрузить обновленные настройки и начать взаимодействие с PostgreSQL.

Шаг 3: Настройка схем данных в PostgreSQL. На этом этапе необходимо подготовить базу данных PostgreSQL к работе с ALD, настроив схемы данных для поддержки хранения и извлечения информации.

1. Необходимо создать базу данных для ALD, которая будет использоваться для хранения данных о пользователях:

```
CREATE DATABASE ald_data OWNER ald_user;
```

Эта команда создает базу данных `ald_data`, где `OWNER` указывает на роль `ald_user`, которая будет контролировать все операции ALD.

2. Затем следует подключиться к базе данных и создать таблицы для хранения информации о пользователях. Например, таблица users может включать такие поля, как id, username, password\_hash и created\_at:

```
CREATE TABLE users (  
    id SERIAL PRIMARY KEY,  
    username VARCHAR(50) UNIQUE NOT NULL,  
    password_hash VARCHAR(255) NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

Эта структура позволяет хранить основные учетные данные пользователей, которые ALD будет использовать для аутентификации. Поле password\_hash хранит зашифрованные пароли, а created\_at автоматически фиксирует дату и время создания записи.

3. Дополнительно необходимо создать таблицы для хранения информации о группах пользователей и их атрибутах. Например, таблицы groups и user\_groups поддерживают данные о принадлежности пользователей к группам:

```
CREATE TABLE groups (  
    id SERIAL PRIMARY KEY,  
    group_name VARCHAR(50) UNIQUE NOT NULL  
);  
  
CREATE TABLE user_groups (  
    user_id INT REFERENCES users(id),  
    group_id INT REFERENCES groups(id),  
    PRIMARY KEY (user_id, group_id)  
);
```

Эти таблицы позволяют ALD связывать пользователей с определенными группами, обеспечивая контроль за доступом и распределением полномочий.

Завершив настройку таблиц и схем данных, PostgreSQL будет готов к использованию ALD для централизованного хранения учетных данных и управления доступом. Теперь ALD сможет взаимодействовать с PostgreSQL, запрашивая учетные данные пользователей и их группы, что обеспечивает высокую надежность и удобство управления доступом в корпоративной сети.

Таким образом, интеграция ALD с PostgreSQL в Astra Linux требует внимания к настройкам производительности и безопасности. При работе системы могут возникать такие проблемы, как таймауты соединений или ошибки прав доступа. Решение этих проблем включает оптимизацию параметров PostgreSQL, таких как увеличение памяти и настройка пула соединений для улучшения производительности. Важнейший аспект — безопасность взаимодействия ALD и PostgreSQL. Для защиты данных применяются шифрование, контроль доступа на основе ролей и механизмы аудита, что позволяет контролировать целостность данных и соответствует требованиям корпоративной безопасности. Такой подход обеспечивает высокую надежность, управляемость и безопасность в корпоративной инфраструктуре на базе Astra Linux.

\*\*\*

1. Анализ и управление рисками информационной безопасности объекта критической информационной инфраструктуры / А. М. Гельфанд, В. В. Сигачева, А. В. Архипов, Л. К. Сиротина // Вестник Санкт-

- Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. – 2023. – № 3. – С. 21-27.
2. Горбань С. А., Красов А. В., Цветков А. Ю. Оценка эффективности механизмов контроля правами доступа в ОС Linux // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2023). – 2023. – С. 345-348.
  3. Красов А. В., Сахаров Д. В., Тасюк А. А. Проектирование системы обнаружения вторжений для информационной сети с использованием больших данных // Научные технологии в космических исследованиях Земли. – 2020. – Т. 12. – №. 1. – С. 70-76.
  4. Лаврова Д. С. и др. Предупреждение Dos-атак путем прогнозирования значений корреляционных параметров сетевого трафика // Проблемы информационной безопасности. Компьютерные системы. – 2018. – №. 3. – С. 70-77.

**Протасова Е.А., Переверзев А.Л., Силантьев А.М.,  
Примаков Е.В., Солодовников А.П.**  
**Способы аппаратной реализации функции активации в нейронных сетях**

*Национальный исследовательский университет «МИЭТ»  
(Россия, Москва)*

doi: 10.18411/trnio-12-2024-887

**Аннотация**

Функция активации является неотъемлемой частью нейронных сетей. Выбор метода аппроксимации функции активации и его аппаратная реализация — основные аспекты, от которых зависит скорость и точность работы алгоритма. В этой статье рассматриваются способы аппаратной реализации различных функций активации.

**Ключевые слова:** функция активации, нейронная сеть, аппаратная реализация, аппроксимация.

**Abstract**

The activation function is an integral part of neural networks. The choice of the activation function approximation method and its hardware implementation are the main aspects on which the speed and accuracy of the algorithm depend. This article discusses ways to implement various activation functions in hardware.

**Keywords:** activation function, neural network, hardware implementation, approximation.

Выбор метода аппроксимации функции активации и его аппаратная реализация — основные аспекты, от которых зависит скорость и точность работы алгоритма. Низкая точность аппроксимации позволяет получить высокую производительность, а уменьшение погрешности приводит к увеличению аппаратных затрат в ПЛИС и снижению скорости обработки данных.

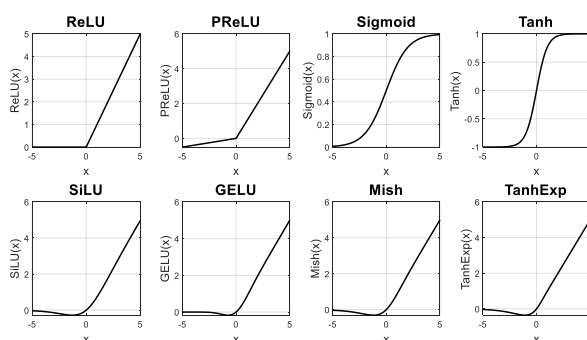


Рисунок 1. Виды функции активации.

### ReLU — Rectified Linear Unit

$$ReLU(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

ReLU является одной из самых часто используемых функцией активации из-за простоты программной и аппаратной реализации и возможности достижения высокой точности модели. Реализация данной функции проста: если входное значение является положительным, то результатом функции будет значение со входа, если отрицательным — ноль. Такой подход позволяет достичь большой частоты при минимальных затратах аппаратных ресурсов.

### PReLU — Parameteric Rectified Linear Unit

$$PReLU(x) = \begin{cases} p * x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Данная функция устраняет проблему исчезающих нейронов, которой обладает ReLU из-за того, что значение функции равно нулю в отрицательной области. Для PReLU в отрицательной области входное значение функции умножается на небольшой коэффициент, например, на 0.01, как в случае с Leaky ReLU. Это позволяет получить ненулевые значения функции в отрицательной области, тем самым избежав обнуления нейронов. В положительной области функция ведет себя так же, как ReLU, то есть выводит на выход значение, которое было подано на вход.

Для простоты реализации в ПЛИС коэффициент в отрицательной области функции должен быть равен степени числа 2, например  $2^{-6} = 0.015625$ , чтобы избежать использования умножителя, так как это увеличит критический путь и аппаратные затраты.

Умножение на  $2^{-6}$  легко осуществить сдвигом входного значения вправо на 6 позиций.

### Логистическая функция или сигмоида

$$sigm(x) = \frac{1}{1 + e^{-x}}$$

Это еще одна функция активации, которая относится к наиболее часто используемым. Как можно видеть из формулы, реализация данной функции несколько сложнее, чем описанные выше. Стоит отметить, что, так как функция симметрична, то можно реализовать, например, только её отрицательную область. Значение функции в положительной области в таком случае можно получить по формуле:

$$sigm(x) = 1 - sigm(-x) \quad (1)$$

Существует множество способов реализации такой функции. Рассмотрим наиболее популярные из них, они так же могут быть применены и к другим нелинейным функциям.

#### 1. Кусочно-линейная аппроксимация

Суть данной реализации заключается в том, чтобы выбрать начальную и конечную точки функции для аппроксимации (например, от -8 до 0), а также количество частей, на которые будет разбита функция. Каждая такая часть аппроксимируется линейной функцией первого порядка [1]. От количества отрезков будет зависеть точность вычисления данной функции. Стоит отметить, что в области, где производная функции достигает максимального значения (например, на интервале от -1 до 0), можно разбить функцию на более мелкие отрезки, чтобы достичь большей точности.

Для достижения лучшего быстродействия функции может использоваться линейная аппроксимация с коэффициентами, равными степени числа 2 [2]. Такая реализация позволит избежать использования умножителя, что также может

уменьшить количество занимаемых ресурсов в ПЛИС. Недостатком такого подхода является низкая точность вычислений.

## 2. Кусочно-нелинейная аппроксимация

В данном методе функция так же разбивается на отрезки, которые аппроксимируются полиномом второго порядка. Такой подход требует большего количества ресурсов, чем описанный ранее, из-за использования двух умножителей, что также увеличит время вычисления функции. Однако во многом это зависит от конкретной аппроксимации и количества отрезков.

## 3. Таблица поиска

Суть использования таблицы поиска заключается в том, чтобы по определенному адресу обращаться к ячейкам, в которые занесены значения функции. Размер таблицы поиска сильно возрастает с увеличением разрядности чисел, поэтому наиболее оптимально использовать этот метод для чисел с меньшей разрядностью, например, 8 бит. В таком случае, функция будет вычисляться с максимальной для 8 бит точностью, притом довольно быстро.

Для экономии ресурсов в таблице могут храниться не все необходимые значения, а с определенным шагом. Промежуточные значения в таком случае могут вычисляться с использованием интерполяции. Это может сократить аппаратные затраты, однако усложнит реализацию функции.

## 4. CORDIC

Вычисление функции алгоритмом CORDIC позволяет получить её значение с максимальной точностью, что является главным преимуществом данного алгоритма. Однако при использовании такого подхода может потребоваться большое число тактов. Возможно сгладить этот недостаток конвейеризацией CORDIC, однако это увеличит количество используемых ресурсов.

## 5. Комбинированные методы

Описанные выше способы реализации можно комбинировать между собой, сглаживая недостатки каждого. Например, можно использовать кусочно-линейную аппроксимацию, а в таблице поиска хранить разницу между вычисленным и истинным значениями функции [3]. Либо вычислять значения функции с помощью линейной аппроксимации, но в области с большим значением производной функции (для сигмоиды это может быть область от -1.3 до 0.8) использовать таблицу поиска, чтобы добиться большей точности.

### Гиперболический тангенс

Гиперболический тангенс легко можно выразить через логистическую функцию, притом без использования умножителей:

$$\tanh_{\text{sigm}}(x) = 2\text{sigmoid}(2x) - 1$$

Однако необходимо учесть, что точность вычисляемой функции в таком случае будет зависеть от точности сигмоиды.

### SiLU

$$\text{SiLU}_{\text{sigm}}(x) = \text{sigmoid}(x) \cdot x$$

SiLU является частным случаем функции Swish. Существует реализация с кусочно-линейной аппроксимацией данной функции без использования умножителей [4] с максимальной ошибкой, равной 0.088, и среднеквадратической ошибкой, равной 0.00295.

### TanhExp

Является не очень популярной, но достаточно эффективной функцией активации [5]. TanhExp можно реализовать при помощи кусочно-линейной аппроксимации (с максимальной ошибкой 0.03105) и аппроксимации второго порядка (с максимальной ошибкой 0.0074) [6].

### GeLU

$$GELU(x) = x \cdot 0.5 \left( 1 + \operatorname{erf} \left( \frac{x}{\sqrt{2}} \right) \right)$$

Популярная функция активации, однако является довольно сложной для реализации в аппаратуре. Данную функцию можно аппроксимировать двумя способами.

1. Через гиперболический тангенс (максимальная ошибка 0.00047)

$$GELU_{\tanh}(x) = 0.5x \left( 1 + \tanh \left( \sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right) \right)$$

2. Через сигмоиду (максимальная ошибка 0.0063)

$$GELU_{\text{sigm}}(x) = x \cdot \text{sigmoid}(1.702x)$$

Второй способ является менее точным, но также и менее ресурсозатратным.

### Mish

$$Mish(x) = x \cdot \tanh(\ln(1 + \exp(x)))$$

Вычисление Mish может осуществляться следующим образом (максимальная ошибка равна 0.025):

$$Mish_{\text{sigm}}(x) = \begin{cases} x \cdot \text{sigm}(x), & x < -2 \\ x \cdot \text{sigm}(x) - 0.03, & -2 \leq x < -1 \\ x \cdot \text{sigm}(x) - 0.02, & -1 \leq x < 0 \\ x \cdot \text{sigm}(2x), & x \geq 0 \end{cases}$$

В таблице 1 приведены значения точности для реализуемых функций за исключением ReLU и PReLU, так как их точность не зависит от вычислений. Для реализации сигмоиды предлагается применить кусочно-линейную аппроксимацию, разделив область определения функции на 16 отрезков от -8 до 0. Значения функции в положительной области можно получить, используя формулу (1). Гиперболический тангенс, SiLU, GELU, Mish реализуются при помощи сигмоиды, поэтому от неё также зависит точность этих функций.

Таблица 1

Сравнение реализаций функции активации.

Функция	Реализация	Максимальная ошибка	Среднеквадратическая ошибка (MSE)
Sigmoid	Линейная аппроксимация	0.0075	$4.58 \times 10^{-6}$
Tanh	$\tanh_{\text{sigm}}$	0.015	$9.17 \times 10^{-6}$
SiLU	$\text{SiLU}_{\text{sigm}}$	0.013	$1.38 \times 10^{-5}$
GELU	$\text{GELU}_{\text{sigm}}$	0.02	$7.63 \times 10^{-5}$
Mish	$\text{Mish}_{\text{sigm}}$	0.029	$7.66 \times 10^{-5}$
TanhExp	Линейная аппроксимация	0.031	$1.24 \times 10^{-4}$

\*\*\*

3. C. Yu and Y. -C. Hung, Generic Hardware Integrating Multiple Activation Functions for Deep Learning Applications // 2024 International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), Taichung, Taiwan, 2024. S. 333-334.
4. Tsmots, O. Skorokhoda and V. Rabyk, Hardware Implementation of Sigmoid Activation Functions using FPGA // 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems. Polyana, Ukraine, 2019. S. 34-38.
5. H. Namin, K. Leboeuf, R. Muscedere, H. Wu and M. Ahmadi, Efficient hardware implementation of the hyperbolic tangent sigmoid function // 2009 IEEE International Symposium on Circuits and Systems. Taipei, Taiwan, 2009. S. 2117-2120.
6. K. Choi, S. Kim, J. Kim and I. -C. Park, Hardware-Friendly Approximation for Swish Activation and Its Implementation // IEEE Transactions on Circuits and Systems II: Express Briefs. 2024. vol. 71, no. 10, S. 4516-4520.
7. Xinyu Liu, Xiaoguang Di1, TanhExp: A Smooth Activation Function with High Convergence Speed for Lightweight Neural Networks // IET Research Journals, 2015. S. 1-7.

8. S. Bouguezzi, H. Faiedh and C. Souani, Hardware Implementation of Tanh Exponential Activation Function using FPGA // 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD). Monastir, Tunisia, 2021. S. 1020-1025.

**Ребницкая И.В., Никифорова Е.М., Скляренко А.А., Туренко Е.В.**  
**Инфографика и другие информационные системы в аэропортах**

*Санкт-Петербургский государственный университет гражданской авиации  
имени Главного маршала авиации. А.А. Новикова  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-888

**Аннотация**

Аэропорты в значительной степени полагаются на информационные системы для обеспечения бесперебойной работы своей повседневной деятельности. Эти системы играют решающую роль в управлении различными аспектами работы аэропорта, включая обработку пассажиров, обработку багажа, планирование рейсов и безопасность. В этой статье мы рассмотрим наиболее широко используемые информационные системы в аэропортах.

**Ключевые слова:** информационные системы, инфографика, управление аэропортом, аэропорт Внуково, автоматизация, оптимизация, эффективность, безопасность.

**Abstract**

Airports rely heavily on information systems to ensure the smooth operation of their day-to-day operations. These systems play a crucial role in managing various aspects of airport operations, including passenger handling, baggage handling, flight planning and security. In this article, we will look at the most widely used information systems at airports.

**Keywords:** information systems, airport management, Vnukovo airport, automation, optimization, efficiency, safety.

**Введение**

Рост экономики, улучшение благосостояния граждан и расширение внешнеэкономических связей способствовали развитию авиационной отрасли России. Многие отечественные аэропорты модернизируют инженерную инфраструктуру и внедряют амбициозные ИТ-проекты.

Современные аэропорты стремятся повысить комфорт и безопасность пассажиров, улучшить взаимодействие с авиаперевозчиками. Драйверами этих процессов выступают крупные государственные мероприятия и желание повысить конкурентоспособность.

**Инфраструктурная взаимосвязь**

Примером успешного развития аэропортов является модернизация ИТ-инфраструктуры. Это происходит одновременно с улучшением инженерной инфраструктуры, что позволяет эффективно управлять пассажиропотоком.

**Специализированные решения**

Для решения задач контроля пассажиропотока и улучшения сервисного обслуживания аэропорты используют различные системы. В настоящее время становятся актуальными способы представления информации с помощью инфографики. Инфографика – это визуализация данных (статичная и интерактивная информация, выполненная в виде схем, таблиц, указателей, электрифицированных табло, экранов, располагаемых в местах скопления пассажиров).

**1. Системы управления аэропортами (AMS).**

Системы управления аэропортами представляют собой комплексные программные решения, которые включают различные операции аэропорта, включая обработку

пассажиров, обработку багажа и планирование рейсов. AMS предоставляет централизованную платформу для администрации аэропорта для эффективного управления своими операциями.

Некоторые из ключевых функций AMS включают:

- обработка и регистрация пассажиров;
- отслеживание и обработка багажа;
- планирование и управление рейсами;
- распределение и оптимизация ресурсов;
- безопасность и контроль доступа.

## 2. Системы управления воздушным движением (ATCS).

Системы управления воздушным движением являются особенно важными информационными системами, используемыми в аэропортах для управления воздушным движением. ATCS предоставляет авиадиспетчерам информацию в реальном времени о движении рейсов, погодных условиях и потоке воздушного движения. Система позволяет диспетчерам принимать обоснованные решения, обеспечивая безопасное и эффективное управление воздушным движением.

Основные характеристики ATCS включают:

- отслеживание и мониторинг рейсов;
- информация и прогноз погоды;
- управление потоками воздушного движения;
- системы предотвращения столкновений.

## 3. Системы обработки пассажиров (PPS).

Системы обработки пассажиров предназначены для оптимизации обработки пассажиров, включая регистрацию, сдачу багажа и досмотр. PPS сокращает время ожидания, улучшает качество обслуживания пассажиров и повышает безопасность аэропорта.

Основные характеристики PPS включают:

- автоматизированная регистрация и сдача багажа;
- киоски самообслуживания и мобильная регистрация;
- биометрическая идентификация и досмотр безопасности;
- отслеживание и мониторинг пассажиров в реальном времени.

## 4. Системы обработки багажа (BHS).

Системы обработки багажа предназначены для оптимизации обработки багажа и сокращения случаев неправильного обращения. BHS использует автоматизированные системы для отслеживания и транспортировки багажа от регистрации до загрузки самолета.

Ключевые особенности BHS включают:

- автоматизированная сортировка и отслеживание багажа;
- сверка и отслеживание багажа;
- мониторинг и отчетность о багаже в реальном времени.

## 5. Информационные системы безопасности (SIS).

Информационные системы безопасности имеют решающее значение для обеспечения безопасности аэропорта. SIS предоставляет информацию в реальном времени об угрозах безопасности.

Ключевые особенности SIS включают:

Системы наблюдения и мониторинга;

- контроль доступа и биометрическая идентификация;
- системы обнаружения и реагирования на угрозы;
- отчеты и аналитика по безопасности в реальном времени.

## 6. Системы отображения информации о рейсах (FIDS).

Системы отображения информации о рейсах используются для отображения информации о рейсах в реальном времени для пассажиров.

Ключевые особенности FIDS включают:

- отображение информации о рейсах в реальном времени;
- автоматизированное управление расписанием рейсов;
- системы уведомления и оповещения пассажиров.

Комплекс систем управления аэропортом Внуково

Информационное обеспечение аэропорта Внуково основано на решении системы Inform GroundStar и включает:

- Операционную базу данных аэропорта (AODB) с шиной обмена данными.
- Систему формирования, управления и контроля исполнения расписания (FIP).
- Систему формирования расписания и подготовки плана полётов (SFP).
- Систему управления ресурсами (RMS).
- Систему визуального мониторинга местоположения мобильных ресурсов аэропорта (GFM).
- Систему обмена информацией с перронным персоналом (ComControl).
- Систему управления задержками (HubControl).
- Подсистему формирования шаблонов «плавающих» смен персонала (ShiftPattern).
- Подсистему планирования и управления сменами персонала (Roster Control).
- Систему оценки времени посадки воздушных судов (VKOAir).

AODB и шина данных MB являются центральным банком данных, обеспечивающим хранение и систематизацию всей информации об аэропортовой деятельности. Сбор информации осуществляется с использованием специализированной шины данных, что упрощает процедуры информационной интеграции.

FIP: обеспечивает точное состояние суточного плана полетов и необходимого количества ресурсов. Система автоматизации позволяет минимизировать влияние человеческого фактора и оперативно принимать решения на основе данных.

SFP: система для подготовки плана полетов на горизонте планирования полугода-год.

RMS Planning: система для долгосрочного и краткосрочного планирования ресурсов.

RMS RealTime: система для оперативного управления ресурсами, включая стоянки воздушных судов, терминальные ресурсы и мобильные ресурсы.

ComControl: система для обмена информацией с помощью мобильных устройств.

Shiftpattern: модуль для формирования шаблонов «плавающих» смен.

Roster Control: планирование и управление сменами персонала на основании суточного плана.

HubControl: управление процессами и предупреждения о задержках.

GroundFleet: мониторинг ресурсов и воздушных судов на перроне аэропорта, что позволяет оптимизировать работу диспетчера и контролировать их местоположение.

Ключевые преимущества системы:

- повышение оперативности и эффективности диспетчерского управления;
- повышение безопасности обслуживания;
- объективный контроль за местоположением и состоянием спецтранспорта;
- статистическая обработка данных для оптимизации работы.

В заключение следует отметить, что рассмотренные информационные системы имеют важное значение для обеспечения бесперебойной работы аэропортов. Используя эти системы, аэропорты могут повысить эффективность работы, улучшить качество обслуживания пассажиров и обеспечить безопасность.

\*\*\*

2. Отчеты авиационной отрасли: международная ассоциация воздушного транспорта (IATA), международный совет аэропортов (ACI).
3. Федеральное агентство воздушного транспорта (Росавиация), Министерство транспорта Российской Федерации могут: правила, стандарты и рекомендации по управлению аэропортами в России.
4. Журнал Journal of Air Transport Management
5. International Journal of Aviation Management.

**Токарев Е.В.**

**Выбор протоколов TCP, UDP и WebSocket для различных типов приложений**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-889

**Аннотация**

В статье рассматриваются протоколы TCP, UDP и WebSocket и их применимость для различных приложений. Описаны особенности протоколов и критерии выбора в зависимости от требований к передаче данных и скорости отклика.

**Ключевые слова:** TCP, UDP, WebSocket, сетевое взаимодействие, передача данных, сетевые протоколы.

**Abstract**

The article discusses TCP, UDP and WebSocket protocols and their applicability to various applications. The features of the protocols and the selection criteria depending on the requirements for data transmission and response speed are described.

**Keywords:** TCP, UDP, WebSocket, networking, data transfer, network protocols.

В системах реального времени и современных сетевых приложениях выбор протокола передачи данных играет ключевую роль, определяя скорость и надежность соединения. Основными протоколами, которые применяются в контексте CPBT, являются TCP, UDP и WebSocket. Ниже рассмотрен каждый из них, их особенности и применимость.

Для технологии CPBT (Position-Based Tick Rate) протокол TCP (Transmission Control Protocol) является важным компонентом, обеспечивающим надежную передачу данных в сетевом взаимодействии. TCP используется в ситуациях, когда необходима высокая точность и гарантия доставки данных, что особенно важно для передачи критически значимых данных. На рисунке 1 ниже представлена структура TCP-сегмента и его инкапсуляция в IP-пакет. TCP работает поверх IP, добавляя свои механизмы для управления соединением и контроля целостности данных.

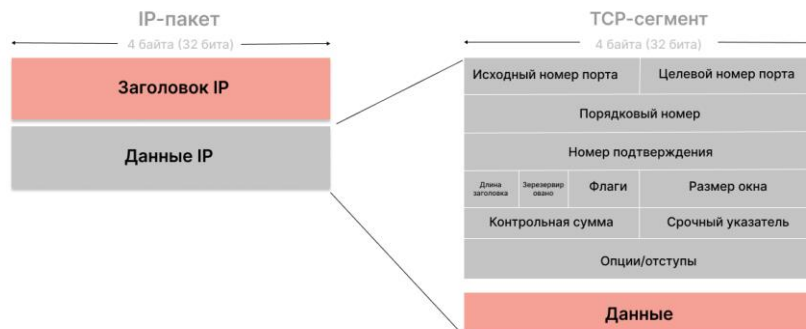


Рисунок 1. Структура пакета при работе протокола TCP/IP.

TCP-сегмент включает несколько ключевых полей:

- Исходный номер порта и целевой номер порта: Эти поля идентифицируют порты отправителя и получателя, определяя, какое приложение на каждом устройстве должно получить данные.
- Порядковый номер: это число используется для упорядочивания сегментов и обеспечения их передачи в правильной последовательности, даже если они приходят в разном порядке.
- Номер подтверждения: указывает, какой следующий сегмент ожидает получить отправитель, подтверждая успешное получение предыдущих данных.
- Флаги: Управляющие флаги, такие как SYN, ACK и FIN, управляют состоянием соединения и помогают в установке, поддержании и завершении соединения.
- Контрольная сумма: Поле для проверки целостности данных, которое используется для выявления ошибок при передаче.
- Размер окна: указывает объем данных, который может быть отправлен до получения подтверждения, что позволяет оптимизировать скорость передачи и регулировать поток данных.

TCP-сегмент инкапсулируется в IP-пакет, который содержит заголовок IP, включающий адреса отправителя и получателя. Этот уровень инкапсуляции позволяет TCP-сообщениям передаваться через сети с разными маршрутизаторами и промежуточными устройствами, поддерживая связь между отправителем и получателем.

В CPBT протокол TCP используется для передачи данных, где требуется высокая точность и надежность. Например, TCP может быть полезен для передачи данных, которые должны быть доставлены в полном объеме и в правильной последовательности, таких как команды или настройки, критически важные для работы системы. Благодаря надежному механизму подтверждений и восстановления потерянных данных TCP гарантирует, что такие данные будут корректно получены, даже если передача потребует повторных отправок из-за ошибок.

Использование TCP обеспечивает целостность и безопасность передачи данных, делая его подходящим выбором для компонентов CPBT, где недопустимы потери данных и критична корректность их доставки.

В технологии CPBT (Position-Based Tick Rate) протокол UDP играет ключевую роль, обеспечивая быструю передачу данных с минимальными задержками. UDP отличается от TCP тем, что не требует подтверждений и установления соединения, что позволяет отправлять данные без дополнительных накладных расходов на проверку целостности и последовательности пакетов. Это особенно важно в системах, где требуется высокая скорость и низкая задержка передачи данных. На рисунке 2 представлен заголовок UDP.



Рисунок 2. Заголовок UDP.

Заголовок состоит из следующих полей:

1. Порт отправителя и порт получателя: Эти поля указывают на начальную и конечную точки передачи, определяя порты на устройствах, между которыми осуществляется обмен данными.
2. Длина сообщения UDP: показывает общий размер сообщения, включая заголовки и данные, что позволяет сети определить объем передаваемой информации.
3. Контрольная сумма: используется для быстрой проверки целостности данных. Она необязательна в UDP, и некоторые системы могут ее не учитывать, чтобы еще больше сократить время передачи.

Данные инкапсулируются в заголовок UDP и передаются по IP-сети, что делает передачу максимально быстрой и легкой. В контексте CPBT, UDP подходит для отправки информации о состоянии системы или объекта, где небольшие потери данных не критичны, но важна высокая скорость и минимальная задержка. Например, при передаче данных о позиции движущихся объектов UDP позволяет обеспечить более плавное отображение, даже если отдельные пакеты теряются. Использование UDP в CPBT позволяет системе передавать данные без задержек, связанных с подтверждением и пересылкой потерянных пакетов, что улучшает скорость реакции системы и её отзывчивость.

WebSocket — это протокол прикладного уровня, который устанавливает постоянное двустороннее соединение между клиентом и сервером поверх TCP. В отличие от классических HTTP-запросов, WebSocket позволяет поддерживать постоянное соединение, что делает его полезным для приложений, где требуется постоянная передача данных в реальном времени. WebSocket используется в CPBT для взаимодействия клиента и сервера, позволяя передавать данные об игровых событиях с минимальной задержкой и высокой производительностью. WebSocket также может быть использован совместно с TCP и UDP в зависимости от задач. Протокол позволяет передавать различные типы данных и легко адаптируется к условиям реального времени, что делает его важной частью CPBT для обеспечения минимальных задержек и стабильной передачи данных.

Выбор сетевого протокола в системах реального времени (CPBT) зависит от специфических требований приложения к скорости, надежности и объему передаваемых данных. В таблице 1 ниже представлено, как TCP, UDP и WebSocket применяются в разных типах приложений и почему тот или иной протокол является предпочтительным выбором.

Таблица 1

*Выбор протоколов передачи данных для различных типов приложений в системах реального времени.*

<b>Протокол</b>	<b>Типы приложений</b>	<b>Причины выбора протокола</b>
<i>TCP</i>	<i>Финансовые системы и биржи</i>	<i>Гарантирует полную и корректную передачу данных, что критично для финансовых операций и предотвращения потерь.</i>
	<i>Передача файлов и документов</i>	<i>Обеспечивает целостность передаваемых файлов за счет подтверждения получения всех сегментов.</i>
	<i>Веб-приложения (HTTP/HTTPS)</i>	<i>Стабильная и защищенная передача данных, необходимая для корректного отображения и функционирования веб-страниц.</i>
<i>UDP</i>	<i>Онлайн-игры и мультимедийные приложения</i>	<i>Минимальные задержки для быстрого отклика, где допустимы небольшие потери пакетов (например, обновление позиции).</i>
	<i>Видеоконференции и VoIP</i>	<i>Позволяет передавать аудио и видео с минимальной задержкой, обеспечивая плавную связь, несмотря на возможные потери.</i>

	<i>Системы мониторинга и IoT</i>	<i>Снижает нагрузку на сеть, обеспечивает быструю передачу данных для оперативного мониторинга состояния объектов.</i>
<i>WebSocket</i>	<i>Чаты и мессенджеры</i>	<i>Постоянное соединение для мгновенной доставки сообщений, необходимое для интерактивного общения.</i>
	<i>Интерактивные веб-приложения</i>	<i>Обеспечивает мгновенное обновление данных на интерфейсе, создавая бесшовное взаимодействие с пользователем.</i>

Таким образом, выбор протокола зависит от требований приложения к надежности, скорости и задержке. TCP подходит для приложений, где важна корректная передача данных (финансовые системы, передача файлов). UDP предпочтителен для задач с минимальной задержкой, таких как онлайн-игры и видеоконференции. WebSocket используется для постоянного двустороннего обмена данными в реальном времени, например, в чатах и интерактивных веб-приложениях.

\*\*\*

1. Сахаров Д. В. и др. Использование математических методов прогнозирования для оценки нагрузки на вычислительную мощность IoT-сети // Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России. – 2020. – № 2. – С. 86-94.
2. Волкогонов В. Н., Гельфанд А. М., Карамова М. Р. Обеспечение безопасности персональных данных при их обработке в информационных системах персональных данных // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). – 2019. – С. 266-270.
3. Цветков А. Ю., Шалаева М. Е., Юрченко М. А. Обеспечение безопасности в клиент-серверном Java приложении для учета и автоматической проверки лабораторных работ // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). – 2019. – С. 756-761.
4. Котенко И. В. и др. Модель человеко-машинного взаимодействия на основе сенсорных экранов для мониторинга безопасности компьютерных сетей // Региональная информатика "РИ-2018". – 2018. – С. 149-149.

**Токарев Е.В.**

**Основы сетевого программирования и роль протоколов и сокетов в сетевом взаимодействии**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

*doi: 10.18411/trnio-12-2024-890*

**Аннотация**

В статье рассматриваются основные принципы сетевого программирования, типы сетевых приложений, а также роль протоколов и сокетов в обеспечении взаимодействия между ними. Особое внимание уделено клиент-серверной и распределенной архитектурам, а также функциям и применению протоколов TCP/IP.

**Ключевые слова:** сетевое программирование, клиент-серверная архитектура, распределенные системы, протокол TCP/IP, сокет, Python.

**Abstract**

The article discusses the basic principles of network programming, types of network applications, as well as the role of protocols and sockets in ensuring interaction between them. Special attention is paid to client-server and distributed architectures, as well as the functions and application of TCP/IP protocols.

**Keywords:** network programming, client-server architecture, distributed systems, TCP/IP protocol, sockets, Python.

Сетевые приложения отличаются от обычных тем, что они обмениваются данными через компьютерную сеть. Под сетевыми приложениями можно понимать любые программы, взаимодействующие с другими системами или компонентами, используя каналы передачи данных. Этот обмен, может быть, как с внешними сервисами, так и между компонентами одной системы, созданной одной командой.

Возможность обмена данными по сети позволяет разработчикам создавать приложения с широким функционалом. Например, можно обращаться к сторонним сервисам через их открытое API для получения данных, таких как информация о погоде. Также можно разработать собственный сервис, доступный для других пользователей, предоставив публичный API.

Сетевые приложения также позволяют создавать распределенные системы, где множество компонентов взаимодействуют, находясь на разных компьютерах. Такой подход используется в пиринговых сетях, распределенных вычислительных системах и ботнетах. Другой популярный вариант - централизованная схема клиент-сервер, где клиентская часть приложения взаимодействует с серверной, расположенной на сервере разработчика. Клиент-серверная архитектура позволяет делить логику приложения на две части и организовывать централизованное хранение данных, обеспечивая доступ к ним с различных устройств.

Например, банки активно используют омниканальные схемы взаимодействия, позволяя пользователю начать операцию в мобильном приложении и продолжить её в веб-кабинете без потери данных. Это создает ощущение бесшовного взаимодействия, так как информация о пользователе и его действиях сохраняется на сервере.

Сетевые приложения можно разделить на несколько типов:

1. Обмен данными с внешними сервисами: Использование API для получения данных.
2. Публичные сервисы: Предоставление данных через собственный API.
3. Распределенные системы: Взаимодействие независимых компонентов.
4. Автоматическое обновление: Программа проверяет сервер разработчика на наличие новой версии.
5. Клиент-серверные системы: Разделение приложения на клиентскую и серверную части.
6. Централизованное хранилище данных: Организация общего доступа к данным для пользователей.
7. Омниканальные системы: Сохранение данных о действиях пользователя в разных каналах для бесшовного взаимодействия.

На рисунке 1 ниже показан пример клиент-серверной архитектуры, где веб-сервер обменивается данными с различными клиентскими устройствами через браузеры. Такая схема взаимодействия является основой для большинства современных веб-приложений, где данные централизованно обрабатываются и предоставляются пользователям по сети.

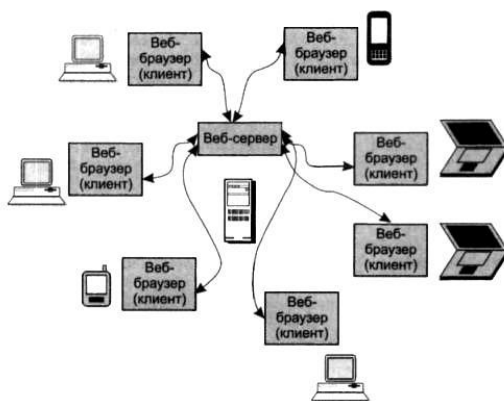


Рисунок 1. Пример клиент – серверной архитектуры.

При создании сетевых приложений важно определить архитектуру, которая будет использоваться для организации обмена данными между компонентами. Сетевые приложения обычно реализуются в одной из двух архитектурных моделей: клиент-серверной или распределенной.

Клиент-серверная архитектура — наиболее популярный подход в сетевом программировании. В такой схеме приложение разделяется на две основные части: сервер и клиенты. Серверный модуль постоянно работает, ожидая запросов от клиентов, обрабатывает их и отправляет ответы. Клиенты, взаимодействуя с пользователем, инициируют соединения с сервером для выполнения операций. На рисунке 2 ниже показано, как сервер обрабатывает запросы от нескольких клиентов, посылая им ответы.

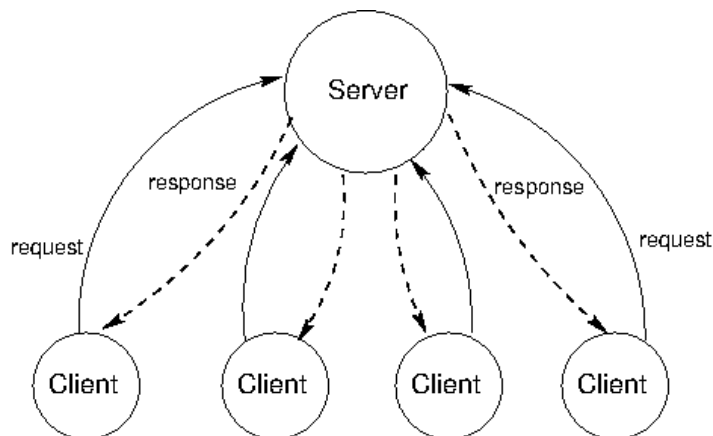


Рисунок 2. Схема клиент-серверной архитектуры.

Данная архитектура проста в понимании и позволяет четко разделить функции между сервером и клиентом. Однако у нее есть недостаток: если сервер выходит из строя, приложение перестает функционировать, поскольку сервер является единой точкой отказа. Несмотря на это, клиент-серверная архитектура широко используется в большинстве сетевых приложений и поддерживается множеством фреймворков и библиотек.

В распределенной архитектуре приложение состоит из множества одинаковых модулей, каждый из которых может выполнять функции как клиента, так и сервера. Эти модули соединяются друг с другом и обмениваются данными произвольным образом, образуя гибкую сеть. На рисунке 3 ниже показана схема распределенной системы, в которой каждый узел может обмениваться данными с другими узлами напрямую.

## Distributed Applications

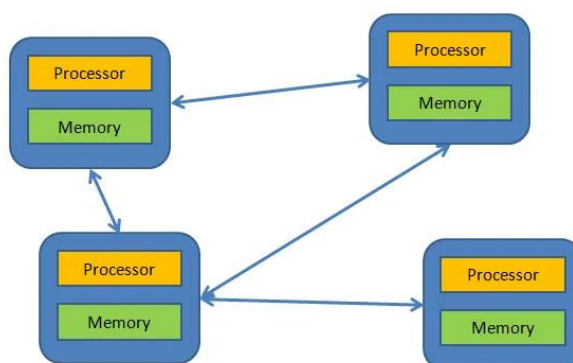


Рисунок 3. Схема распределенной системы.

Примером распределенной архитектуры являются пиринговые сети, в которых каждый узел способен подключаться к любому другому и передавать информацию. Такая система продолжает функционировать даже при недоступности некоторых узлов, так как не зависит от единого центрального сервера. Распределенные приложения более устойчивы к сбоям, но их проектирование и реализация сложнее, так как требуют продуманного подхода к организации обмена данными и взаимодействию узлов.

Протоколы — это наборы правил и стандартов, которые определяют, как данные передаются, принимаются и интерпретируются в сетевом окружении. Они устанавливают форматы данных, процедуры и методы, используемые для обмена информацией между разными системами, и обеспечивают согласованность и надежность в передаче данных.

Наиболее распространенные протоколы, используемые в сетевом программировании, принадлежат к семейству TCP/IP, включающему такие важные протоколы, как TCP (Transmission Control Protocol), UDP (User Datagram Protocol), HTTP (Hypertext Transfer Protocol) и FTP (File Transfer Protocol).

Каждый сетевой интерфейс IP-сети имеет уникальный в этой сети адрес (IP-адрес). Упрощенно можно считать, что каждый компьютер в сети Интернет имеет собственный IP-адрес. При этом в рамках одного сетевого интерфейса может быть несколько (до 65536) сетевых портов. Для установления сетевого соединения приложение клиента должно выбрать свободный порт и установить соединение с серверным приложением, которое слушает (listen) порт с определенным номером на удаленном сетевом интерфейсе. Пара IP-адрес и порт характеризуют сокет (гнездо) - начальную (конечную) точку сетевой коммуникации.

Сокеты могут применяться для связи процессов как на удаленной машине, так и на локальной. В первом случае, естественно, необходимо, чтобы удаленная машина была доступна по сети. Это можно проверить при помощи команды пинг. Во втором случае сокеты могут выступать как механизм межпроцессного взаимодействия. Или вы можете использовать одну машину для всех процессов-компонентов вашей программной системы, например, для отладки в процессе разработки.

Для создания соединения TCP/IP необходимо два сокета: один на локальной машине, а другой - на удаленной. Таким образом, каждое сетевое соединение имеет IP-адрес и порт на локальной машине, а также IP-адрес и порт на удаленной машине. Как правило, порт локальной машины (исходящий порт) не так важен и его номер не особенно используется в практике. Но порт серверного сокета — это важная информация.

Сокеты могут быть клиентские и серверные. Серверный сокет — это функция в программе, которая сидит на определенном порту и “слушает” входящие соединения. Процедура создания серверного сокета аналогична вводу текста из консоли: программа блокируется до тех пор, пока пользователь не ввел что-то. Когда это случилось, программа разблокируется и может продолжать выполнение и обработку полученных данных. Также и серверный сокет: ждет, когда к нему подключится клиент и тогда продолжает выполнение программы и может считывать данные из сокета (которые послал клиент) и отправлять данные в сокет. Клиентский же сокет, наоборот, сразу пытается подключиться к определенному узлу сети (это может быть локальная машина, или, чаще, удаленный компьютер) и на определенный сетевой порт. Если на этой машине на этом порту “сидит” серверный сокет, то подключение происходит успешно. Если же данный сокет никем не прослушивается, то процедура подключения возвращает ошибку.

В языке программирования Python существует стандартный модуль `socket`, который реализует все необходимые функции для организации обмена сообщениями через сокеты. Для его использования его достаточно импортировать (так как это модуль стандартной библиотеки, устанавливать его не нужно, он идет в поставке с дистрибутивом Python):

## 1 | import socket

Для начала построения сетевого взаимодействия необходимо создать сокет:

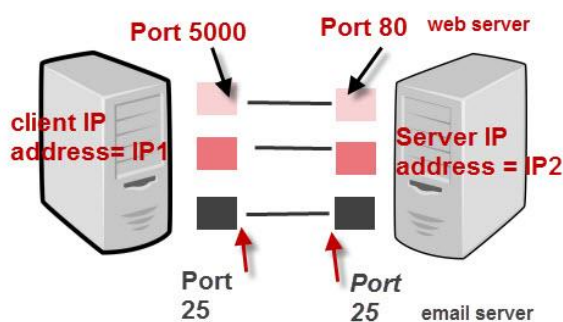
```
1 | sock = socket.socket()
```

Здесь ничего особенного нет, и данная часть является общей и для клиентских, и для серверных сокетов. Далее мы будем писать код отдельно для сервера и для клиента.

Существует несколько видов сокетов, которые немного различаются по сфере применения и деталях реализации. Самыми распространенными являются Интернет-сокеты. Они используются для пересылки информации между процессами. Есть еще сокеты UNIX, они не используют Интернет-протоколы для обмена сообщениями, и используются для организации межпроцессного взаимодействия.

Также среди Интернет сокетов существуют потоковые и датаграммные сокеты. Датаграммные сокеты называют “сокеты без соединения”, они используют протокол UDP вместо TCP. Потоковые сокеты обеспечивают гарантированную доставку, очередность сообщений, они более надежны. Протокол HTTP использует именно потоковые сокеты для соединения клиента с сервером. UDP обычно используется для передачи потокового медиа, когда скорость критичнее риска потери единичных пакетов.

Для эффективного использования сетевых сокетов необходимо вспомнить концепцию сетевых портов, что изображено на рисунке 4, так как сокеты их активно используют.



IP Address + Port number = Socket

## TCP/IP Ports And Sockets

Рисунок 4. Схема работы TCP/IP портов и сокетов в сетевом взаимодействии.

IP-адрес или любой другой способ адресации хоста позволяет нам идентифицировать узел сети. Номер порта позволяет указать конкретное приложение на этом хосте, которому предназначен пакет. Номер порта нужен, так как на любом компьютере может быть одновременно запущено множество приложений, осуществляющих обмен данными по сети. Если использовать аналогию с почтовыми адресами, то IP-адрес — это номер дома, а порт — это номер квартиры в этом доме.

Номер порта — это всего лишь 16-битное число, которое указывается в пакете, передающемся по сети. Не нужно путать сетевой порт с физическими разъемами, это чисто программная концепция.

Так как на номер порта отведено 16 бит, существует всего 65536 возможных портов. Причем, номера портов отдельно считаются по протоколам TCP и UDP. Таким

образом, на компьютере одновременно может существовать более 130 тысяч процессов, обменивающихся данными. На практике, свободных портов всегда в избытке и хватает даже для работы множества высоконагруженных серверов.

Но не все номера портов созданы равными. Первые 1024 являются “системными” и используются в основном стандартными приложениями. Существует общепринятое соглашение, какие сетевые службы используют системные порты. Например, служба веб-сервера по умолчанию использует 80 порт для соединений по протоколу HTTP и 443 - для протокола HTTPS. Служба SSH использует порт номер 22. И так далее. Любая стандартная сетевая служба имеет некоторый порт по умолчанию. Кстати, хорошим показателем практикующего администратора является запоминание часто используемых номеров стандартных портов. Специально это учить не нужно, только если вы не хотите блеснуть знаниями на собеседовании.

Для использования системных портов обычно требуются повышенные привилегии. Это сделано для того, чтобы обычные пользователи не “забивали” стандартные порты и не мешали работе системных сетевых служб. Мы вообще не рекомендуем использовать системные порты. Остальные могут использоваться совершенно произвольно и их более чем достаточно для повседневных нужд.

Кстати, хоть сетевые службы используют определенные стандартные порты, они вполне могут быть переназначены на свободные. Служба не “привязана” к номеру порта, это легко регулируется настройками. Например, строго рекомендуется при настройке службы SSH менять стандартный 22 порт на случайный для повышения безопасности.

Порты назначаются процессу при попытке открыть серверный сокет. В этот момент происходит связывание сокета с номером порта, который выбрал программист. Помните, что если вы пытаетесь использовать занятый порт, то это спровоцирует программную ошибку. Поэтому в реальных приложениях стоит сначала проверять, свободен ли порт или нет, либо (что гораздо проще) обрабатывать исключение при попытке связаться с занятым портом.

Сетевые администраторы могут в целях безопасности блокировать соединения на некоторые порты. Так же поступают и программы-файрволлы. Это требуется для повышения безопасности сервера. Вообще, по умолчанию, все порты “закрыты”, то есть подключения к ним блокируется файрволлом. При необходимости системный администратор может “открыть” обмен данными по определенному номеру порта в настройках файрволла. Это следует учитывать при попытках подключения к удаленным машинам.

\*\*\*

1. Котенко И. В. и др. Модель человеко-машинного взаимодействия на основе сенсорных экранов для мониторинга безопасности компьютерных сетей //Региональная информатика" РИ-2018". – 2018. – С. 149-149.
2. Сахаров Д. В. и др. Использование математических методов прогнозирования для оценки нагрузки на вычислительную мощность IoT-сети //Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». – 2020. – №. 2. – С. 86-94.
3. Гельфанд А. М. и др. ОБЛАСТИ ПРИМЕНЕНИЯ АНАЛИТИКИ БОЛЬШИХ ДАННЫХ В КРИТИЧЕСКИХ ИНФОРМАЦИОННЫХ ИНФРАСТРУКТУРАХ //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2022). – 2022. – С. 438-440.
4. Волкогонов В. Н., Гельфанд А. М., Карамова М. Р. Обеспечение безопасности персональных данных при их обработке в информационных системах персональных данных //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). – 2019. – С. 266-270.
5. Катасонов А. И., Штеренберг С. И., Цветков А. Ю. Оценка стойкости механизма, реализующего... Мандатную сущностно-ролевую модель разграничения прав доступа в операционных системах

- семейства `gnu linux` //Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. – 2020. – №. 2. – С. 50-56.
6. Цветков А. Ю., Шалаева М. Е., Юрченко М. А. Обеспечение безопасности в клиент-серверном Java приложении для учета и автоматической проверки лабораторных работ //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019). – 2019. – С. 756-761.

**Цаплин В.И., Панов М.А.**  
**Технология распределенного реестра**

*Уральский государственный экономический университет  
(Россия, Екатеринбург)*

*doi: 10.18411/trnio-12-2024-891*

**Аннотация**

В статье описывается технология распределенного реестра, что это такое, основные виды, какими особенностями обладает. Самая популярная реализация такого реестра – технология блокчейн. Рассматриваем особенности, функционал, методы, которые помогут улучшить уже существующие решения или создать новые. Реальные кейсы внедрения в бизнес и государство.

**Ключевые слова:** распределенные реестры, блокчейн, смарт-контракты, алгоритм консенсуса, Bitcoin, криптовалюта.

В последние десятилетия мир переживает глобальный переход к цифровизации, затрагивающий все сферы человеческой деятельности, от повседневных операций до систем государственного управления. Уже сегодня происходит четвертая промышленная революция.

**Abstract**

The article describes the technology of distributed registry, what it is, the main types, what features it has. The most popular implementation of such a registry is blockchain technology. We consider the features, functionality, methods that will help improve existing solutions or create new ones. Real cases of implementation in business and government.

**Keywords:** distributed ledgers, blockchain, smart contracts, consensus algorithm, Bitcoin, cryptocurrency.

В условиях быстрорастущего объема и потока данных, увеличения информационных процессов появляется необходимость в надежных, бесперебойных и безопасных информационных системах, которые будут способны оперативно справляться со своими задачами. Одним из важнейших цифровых трендов является технология распределенных реестров, которая способна реализовать выполнение вышеперечисленных требований.

Технология распределенного реестра (от английского – Distributed Ledger Technology, сокращенно DLT) – это электронная система баз данных, которая распределена между несколькими сетевыми узлами (нодами) или устройствами.

Узел (нода) – это точка в сети, которая распределяет данные между другими узлами (нодами).

Появившаяся в 2008 году криптовалюта Bitcoin, созданная неким Сатоши Накамото, как раз основана на технологии распределенного реестра. На сегодняшний день, курс одной монеты Bitcoin, который продолжает расти, составляет более 60 тысяч долларов, что говорит о высоком уровне доверия к данной валюте. Помимо этого, монета ни разу не давала технического сбоя, не смотря на то, что с ней происходили различного рода изменения, например, халвинг, постоянные майнинги и огромное количество торгов.

Поговорим о ключевых особенностях технологии распределенного реестра:

1. отсутствие центрального администратора. В данной системе нет единого центра хранения информации;
2. распределённость. Данные распределены между несколькими узлами или серверами;
3. децентрализация. Узлы (ноды) взаимодействуют между собой, а информация между ними синхронизируется в определенном порядке.

Благодаря этим особенностям, в отличие от традиционных баз данных, распределенные реестры гораздо лучше защищены от вмешательства и изменения данных. Но почему и насколько хорошо? Чтобы была возможность что-то с чем-то сравнить, рассмотрим все типы информационных систем более подробно.

Всего выделяют три типа реестров.

Первый, это централизованный реестр (рисунок 1, типы реестров). Это тип реестра, в котором все данные и управление находится в одном центральном узле, это может быть сервер или компьютер. Для получения доступа к данным и операциям, клиенту или устройству необходимо подключиться к этому узлу.

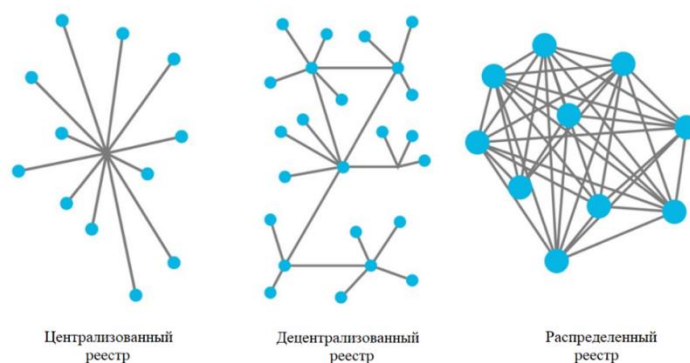


Рисунок 1. Типы реестров.

В качестве примера можно предложить систему управления предприятия, банковские системы или классические серверные приложения.

Из преимуществ такого реестра хотелось бы выделить централизованное администрирование, которое упрощает мониторинг, контроль и другие процессы.

Главный недостаток: атаки, помехи или иные угрозы влекут за собой выход из строя центрального узла, в таком случае вся система выйдет из строя.

Далее, идет децентрализованные реестры (рисунок 1, типы реестров). В этом реестре несколько узлов работают независимо друг от друга, каждый из которых отвечает за свои обязанности. Они способны обмениваться данными или замещать друг друга.

Пример: сеть банкоматов. Представим, что в каждом квартале стоит по одному банкомату. Если один из них выйдет из строя – то остальные способны заменить его. Например, банкомат без наличных средств способен дать клиенту направление до ближайшего доступного банкомата, тем самым переложив свои обязанности на другой. Преимущества видны сразу – реестр тяжело вывести из строя, но из этого и выявляется недостаток: нагрузка с одного узла распределяется между другими узлами, это необходимо учитывать при проектировании.

И, наконец, распределенные реестры (рисунок 1, типы реестров). В такой системе все узлы работают как единое целое, а задачи, данные и вычислительные ресурсы распределены между ними. Как правило, пользователи могут взаимодействовать с любым узлом, а система автоматически управляет распределением данных и задачами между всеми узлами.

В качестве примера: всё та же система распределенных реестров, на которой работает криптовалюта Bitcoin, облачные сервисы (например, Amazon Web Services или Google Cloud).

Главным преимуществом является:

1. высокая масштабируемость. Можно очень просто и легко добавлять новые узлы;
2. отказоустойчивость. Отказ одного узла для системы – не приведет к сбою всей системы;
3. равномерное распределение нагрузки между узлами;
4. актуальные копии базы данных. В распределенном реестре каждый узел способен хранить информацию о базе данных, обеспечивая высокую операционную устойчивость.

Из недостатков всё же следует отметить достаточно сложную инфраструктуру, требующую правильной организации, взаимодействия и согласия между узлами сети.

Поговорим о видах распределенных реестров. На сегодняшний день выделяют следующие варианты:

1. открытые (публичные) реестры. Такие реестры доступны для всех желающих, любой пользователь может подключиться к сети, просматривать данные и участвовать в системных процессах (например, добавление новых узлов). Прозрачность и децентрализация – главные характеристики такого вида. Из реальных кейсов – распределенные реестры сети криптовалюты Bitcoin;
2. закрытые (частные) реестры. В таком реестре доступ к сети получают только определённые пользователи. Приватность – главная характеристика частной сети. Такой реестр используют корпоративные компании, например, банки;
3. гибридные (смешанные) реестры. Такой вид сочетает в себе черты как открытых, так и закрытых реестров. Особенность – частичная прозрачность и децентрализация, имеется возможность установить ограничения на доступ и управление для определенных участников. Пример – распределенный реестр платформы Ripple, которая используется для международных переводов, обработкой транзакций в банковской сфере.

Распределенные реестры не имеют единого центра управления или контроля, в этом и есть его ключевое отличие от, например, централизованного подхода. Отсутствие “начальника” не означает анархию, для контроля состояния реестра существует алгоритм консенсуса.

Алгоритм консенсуса – это механизм, который помогает участникам распределенной сети прийти к единому мнению относительно состояния данных, хранящихся в реестре, без необходимости доверять друг другу.

Основными задачи алгоритма консенсуса являются:

1. обеспечение согласованности данных. Это означает, что все узлы сети должны получать одно и то же представление и состояние реестра;
2. защита от атак. Консенсус защищает сеть от различных атак;
3. обеспечение достоверности. Алгоритмы обеспечивают гарантию о том, что в систему попадает только достоверная информация.

Существует немало различных видов алгоритмов консенсуса, о них поговорим на примерах вариантах реализации сетей распределенных реестров.

Самым популярным вариантом реализации сети с использованием технологии распределенного реестра является блокчейн, о котором и продолжим разговор.

Блокчейн – это база данных, состоящая из непрерывных и взаимосвязанных между собой цепочкой блоков, которая одновременно хранится на множестве соединенных через

интернет компьютеров. Блок – уникальная структура данных, хранящая набор записей (транзакций). Каждый блок имеет свой уникальный номер (хеш). Так как все блоки хранятся последовательно, то существует необходимость запоминать эту последовательность, для этого и нужны хеши, чтобы записывать предыдущий блок. Такой подход обеспечивает целостность и неизменность данных.

В традиционных системах (централизованных), например, банки или нотариальные учреждения, пользователи должны полагаться на посредников, чтобы обеспечить подлинность и безопасность сделок. В случае с блокчейном, система автоматически проверяет и подтверждает каждую транзакцию, что сводит к минимуму вероятность ошибок или манипуляций с данными. Но каким образом? Всё благодаря смарт-контрактам.

Смарт-контракты — это самовыполняющиеся контракты, где все условия прописаны в коде. Эти контракты автоматически выполняются при наступлении заранее определенных условий. Таким образом, применение такой технологии позволяет нам избавиться от посредников, что в свою очередь влечет снизить финансовые издержки, а также повышение автоматизации любой операции, ведь необходимости в ожидании ответа от посредника нет, контракт сразу вступает в силу при выполнении всех условий и наоборот.

Теперь поговорим об алгоритмах консенсуса. В блокчейне они необходимы для того, чтобы между участниками сети было согласие по поводу того, какой блок добавлять в цепочку. Один из них это Proof of Work (PoW), он же называется “Доказательство работы”, он же используется в технологии криптовалюты Bitcoin. Участниками выступают майнеры, каждый из которых решает сложные математические задачи, чтобы подтвердить добавление нового блока в блокчейн. Данный процесс требует большого объема вычислительных ресурсов, взамен обеспечивая высоким уровнем безопасности сети и, в качестве мотивации для участников, майнеров, предлагается вознаграждение. В блокчейне криптовалюты Bitcoin наградой служит сама монета.

Блокчейн – это лишь одна из реализаций распределенного реестра, от потребностей может меняться алгоритм консенсуса, поэтому, он не един для всех. Существует и другой, это Proof of Stake (PoS), в переводе – “Доказательство долей владения”. Здесь участниками сети выступают валидаторы, которые получают право добавлять блоки в зависимости от количества монет, которыми они владеют, а затем, монеты замораживаются. Монеты – это валюта, которой пользуется сеть. В качестве примера – криптовалюта Ethereum, вторая по популярности, после Bitcoin. Существенный плюс – экономичность, меньшее потребление энергии, из недостатков – высокая централизация, так как владельцы большого количества монет могут получить большее влияние.

Из похожего есть ещё Delegated Proof of Stake (DPoS), делегированное доказательство доли. Основные участники – это делегаты, которые выбираются по итогам голосования. Недостаток – такая же высокая централизация и возможная манипуляция голосами. Примером платформы, которая использует блокчейн с таким алгоритмом консенсуса является TRON. На август 2021 года на площадке такого вида распределенного реестра существует более 500 проектов, среди них игры, социальные сервисы, криптообменники.

Теперь поговорим о применении блокчейна на уровне государства. На сегодняшний день идет реализация третьей формы национальной валюты с помощью технологии блокчейн – цифровой рубль, который будет выпускаться Банком России. Каждый рубль будет обладать уникальным номером (токеном), благодаря чему платежи по цифровой форме валюты станут. Преимущества цифрового рубля:

1. офлайн платежи;

2. снижение издержек для бизнеса и государства, благодаря низкой стоимости каждой транзакции (отсутствие посредников, как говорилось ранее);
3. контроль денежных потоков;
4. перспективы в развитии трансграничных платежей.

Что на счёт бизнеса? Внедрение такой технологии в бизнес кажется целесообразным, например, такие компании как СБЕР, МТС, которые создают собственные эко-системы, или промышленные компании, например, Норильский Никель, Газпром, уже используют блокчейн в своих продуктах.

Таким образом, блокчейн – это технология, основанная на распределенных реестрах, которая гарантирует неизменность и сохранность данных. С помощью смарт-контрактов можно автоматизировать процесс транзакций, в следствие чего получаем повышенную скорость операций, а также снижение финансовых издержек. Алгоритмы консенсуса и виды реестров (открытые, закрытые, гибридные) позволяют сделать блокчейн очень гибким, под любые требования и нужды, даже сделать его централизованным.

\*\*\*

1. Леонтьев С.М. Блокчейн и криптовалюты: применение технологии блокчейн за пределами финансовой сферы и новые модели бизнеса / Леонтьев С.М. // Вестник магистратуры. -2023. -№ 7 (142). -С. 100-101.
2. Миценко, Д. Е. Актуальность информационной безопасности в условиях цифровой экономики / Д. Е. Миценко, М. А. Панов // Цифровая трансформация общества и информационная безопасность: Материалы II Всероссийской научно-практической конференции, Екатеринбург, 19 мая 2023 года / Отв.за выпуск А.Ю. Коковихин, отв. редактор М.А. Панов. – Екатеринбург: Уральский государственный экономический университет, 2023. – С. 21-24. – EDN ERUEMQ.
3. Бутко, Г. П. Пути совершенствования цифровых инструментов в деятельности предприятий / Г. П. Бутко, М. А. Меньшикова, М. А. Панов // Цифровые модели и решения. – 2024. – Т. 3, № 1. – С. 39-48. – DOI 10.29141/2949-477X-2024-3-1-4. – EDN PWUVVD.

## Чэнь И.

### Применение методов машинного обучения для анализа больших данных в информационных системах: вызовы и перспективы

*Южно-Уральский государственный университет  
(Россия, Челябинск)*

doi: 10.18411/trnio-12-2024-892

#### Аннотация

В статье рассматриваются подходы к применению методов машинного обучения для анализа больших данных в информационных системах, с акцентом на ключевые вызовы и перспективы их использования. С увеличением объемов данных компании сталкиваются с проблемами обработки, хранения и анализа информации. Алгоритмы машинного обучения предоставляют значительные возможности для трансформации данных в полезные бизнес-инсайты и улучшения процессов принятия решений. В работе обсуждаются современные подходы к интеграции методов МЛ, рассматриваются существующие барьеры и предлагаются перспективные пути решения проблем.

**Ключевые слова:** большие данные, машинное обучение, информационные системы, анализ данных, алгоритмы.

#### Abstract

This paper explores the application of machine learning methods for the analysis of big data in information systems, addressing key challenges and opportunities. With the exponential growth of data, businesses face critical issues related to data processing, storage, and analysis.

Machine learning algorithms offer significant advantages in transforming data into valuable insights for decision-making. This paper delves into the state-of-the-art approaches in machine learning, discusses existing barriers to their implementation, and highlights potential strategies to overcome these challenges.

**Keywords:** big data, machine learning, information systems, data analysis, algorithms.

С увеличением объемов данных возникает необходимость интеграции методов машинного обучения (МО) в существующие информационные системы. Однако на практике такая интеграция сталкивается с рядом трудностей. Одной из основных проблем является качество данных. Большие данные часто бывают неструктурированными, неполными и противоречивыми. Это создает препятствия для применения эффективных алгоритмов МО, которые требуют предварительной подготовки данных. Кроме того, обработка больших массивов данных требует значительных вычислительных ресурсов. Машинное обучение подразумевает использование сложных моделей, которые нуждаются в мощных процессорах и больших объемах памяти. В результате многие компании сталкиваются с высокими затратами на инфраструктуру. Не менее важной проблемой является обеспечение безопасности данных. В современных информационных системах защита данных становится ключевым фактором, особенно в свете многочисленных случаев утечек информации. Применение методов машинного обучения требует тщательной работы с данными и соблюдения всех стандартов защиты информации.

Машинное обучение (МО) в информационных системах поднимает важные вопросы этики и его влияния на общество, что становится все более актуальным в современном мире. Использование алгоритмов, которые могут принимать решения, вызывает серьезную обеспокоенность относительно справедливости и прозрачности этих решений. Часто алгоритмы принимают решения на основе исторических данных, которые могут быть неравномерно распределены или содержать предвзятые паттерны. Это может привести к дискриминации и несправедливым результатам, особенно в таких областях, как здравоохранение, правоприменение и подбор персонала. Например, алгоритмы, которые предвзято оценивают кандидатов на работу или допускают дискриминацию на основе пола, расы или возраста, могут существенно усугубить социальное неравенство.

Одной из главных задач в решении этих этических проблем является обучение алгоритмов на сбалансированных и репрезентативных наборах данных, которые учитывают разнообразие общества. Для этого важно разрабатывать и применять методы, которые могут выявлять и устранять предвзятость в данных и решениях. Постоянный мониторинг работы алгоритмов и корректировка их поведения также способствуют снижению риска несправедливых решений. Создание прозрачных и объяснимых алгоритмов, которые могут обосновывать свои действия, становится обязательным для повышения доверия пользователей и уменьшения рисков.

Еще одним важным аспектом является влияние МО на рынок труда. Внедрение автоматизированных систем с использованием методов машинного обучения может привести к замещению рабочих мест, особенно в сферах, где выполняются рутинные задачи. Это вызывает общественное беспокойство, так как потенциальные потери рабочих мест могут повлиять на целые отрасли и социальноэкономические слои населения. Переход к автоматизации требует разработки программ по переподготовке кадров и внедрения новых возможностей для людей, чьи профессии подвергаются трансформации. Активное обсуждение и регулирование последствий автоматизации помогают минимизировать негативные эффекты на общество, делая возможным плавный переход к использованию технологий в новых контекстах и созданию рабочих мест, требующих человеческого творчества и аналитического мышления.

Таким образом, развитие МО в информационных системах должно сопровождаться этическими рамками и прозрачным подходом к его применению. Важно

обеспечивать, чтобы технологии служили улучшению качества жизни и созданию более справедливого общества, а не углубляли существующие проблемы.

Таблица 1

*Основные вызовы применения машинного обучения в анализе больших данных.*

Вызовы	Описание
Качество данных	Необходимость предварительной очистки и подготовки данных.
Высокие затраты на инфраструктуру	Высокие расходы на вычислительные мощности и оборудование.
Обеспечение безопасности данных	Соблюдение стандартов защиты информации.

## Глава 2: Перспективы и решения применения машинного обучения

Современные исследования в области машинного обучения сосредоточены на разработке оптимизированных алгоритмов, которые могут обрабатывать большие объемы данных с минимальными затратами ресурсов. Такие алгоритмы, как глубокие нейронные сети, решающие деревья и алгоритмы ансамблевого обучения, уже нашли широкое применение. Одним из перспективных направлений является использование гибридных моделей, которые объединяют преимущества нескольких методов для повышения точности и устойчивости предсказаний. Это особенно важно в условиях динамично изменяющейся информации и сложных структур данных.



Рисунок 1: Архитектура обработки больших данных с использованием методов машинного обучения.

С развитием облачных технологий обработка больших данных и использование методов машинного обучения становятся более доступными для предприятий. Облачные платформы предоставляют мощные инструменты для анализа данных, включая готовые решения для машинного обучения. Это позволяет компаниям быстрее внедрять инновации и сокращать затраты на инфраструктуру. В то же время интеллектуальные системы автоматизации, построенные на основе машинного обучения, помогают улучшать процессы принятия решений, снижать ошибки и повышать производительность.

Таблица 2

*Перспективные решения в применении машинного обучения для анализа больших данных.*

Решения	Описание
Гибридные модели	Объединение различных алгоритмов для улучшения качества предсказаний.
Использование облачных технологий	Обеспечение высокой доступности и масштабируемости.
Умные системы автоматизации	Оптимизация бизнес-процессов на основе ИИ.

Внедрение методов машинного обучения для анализа больших данных открывает новые горизонты для развития информационных систем. Однако это направление сталкивается с вызовами, требующими интеграции современных технологий, оптимизации алгоритмов и обеспечения защиты данных. Машинное обучение способно

не только повысить точность и эффективность анализа данных, но и облегчить процессы автоматизации. Информационные системы, основанные на использовании больших данных и алгоритмов ИИ, позволяют принимать более точные решения, разрабатывать новые продукты и повышать конкурентоспособность бизнеса. Будущее машинного обучения и анализа больших данных лежит в постоянной инновации и сотрудничестве между академическими институтами, бизнесом и государственными органами.

\*\*\*

1. Иванов, А.В. Машинное обучение и большие данные // Информационные системы. М., 2020. С. 12-20.
2. Козлов, Б.П. Применение методов анализа данных в бизнесе // Технологии данных. СПб., 2019. С. 33-45.
3. Смирнова, Е.Н. Влияние больших данных на экономику // Наука и технологии. Казань, 2021. С. 48-60.

**Шахбиев Д.О.-Х., Магомедов И.И.**

**Искусственный интеллект в образовании**

*ФГБОУ ВО «Чеченский государственный университет им. А.А. Кадырова»  
(Россия, Грозный)*

*doi: 10.18411/trnio-12-2024-893*

#### **Аннотация**

В данной статье рассматриваются понятие искусственного интеллекта, основные методы его использования в образовании. Перечислены перспективы развития данной технологии и ее недостатки. Также показаны особенности и отрицательные стороны подобных методов. К каждому методу приведены примеры, где описываются готовые ИИ-решения для образовательных целей. На основе собранных данных сформулированы ограничения и возможности развития данных технологий в сфере образования.

**Ключевые слова:** искусственный интеллект, образование, нейросети, персонализация обучения.

#### **Abstract**

The article discusses the concept of artificial intelligence, the main methods of its use in education. The prospects for the development of this technology and its disadvantages are listed. The features and negative sides of such methods are also shown. Examples are provided for each method, which describe ready-made AI solutions for educational purposes. Based on the collected data, the limitations and opportunities for the development of these technologies in the field of education are formulated.

**Keywords:** artificial intelligence, education, neural networks, personalization of learning.

Сегодня искусственный интеллект плотно проникает в сферу образования и начинает ее менять. В мире уже есть примеры успешной реализации ИИ в образовательные программы, а также множество проектов, которые находятся на стадии разработки. Подобные проекты также реализуются и в России, например, платформа «Высшая математика» использует алгоритмы ИИ в процессе создания индивидуальных программ обучения по математике.

Искусственный интеллект (ИИ) представляет собой реализацию компьютерных и информационных систем, способных выполнять задачи, которые обычно требуют человеческого интеллекта, такие как распознавание речи, принятие решений, визуальное восприятие, задачи творческого характера, перевод с одного языка на другой и др.

В образовании ИИ используется преимущественно для автоматизации, оптимизации и персонализации учебных программ и процесса обучения, а также для работы с аналитическими данными.

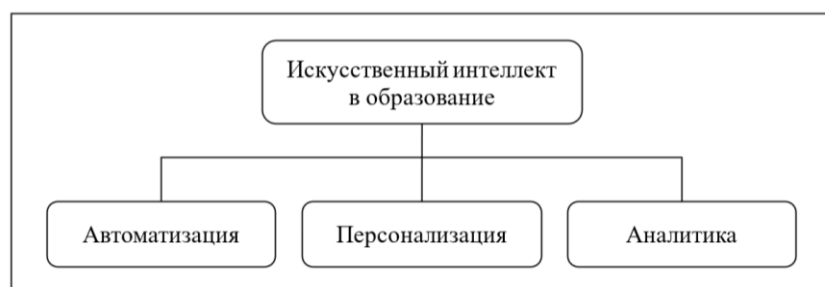


Рисунок 1. Искусственный интеллект в образовании.

Автоматизация и оптимизация учебного процесса. На подготовку учебного и контрольного материала и его проверку уходит много времени и сил, поэтому оптимальным решением для облегчения этих рутинных задач является их автоматизация. Использование ИИ для автоматизации стало распространенной практикой и получила широкую популярность. Например, для автоматизации процесса подготовки тестов для проверки остаточных знаний учащихся был разработан сервис PrepAI.

Сервис позволяет генерировать тесты на основе вашего учебного материала, который предварительно вносится в систему, или на основе интернет-ресурсов (в поисковую строку вводится название темы и сервис автоматический подбирает тестовые вопросы по ней). Встроенные в систему алгоритмы ИИ распознают контент, группируют его и генерирует тестовые задания и ответы к ним. На PrepAI можно создавать различные тесты для школьников и студентов университетов. Данный сервис во многом облегчает работу педагога и позволяет уделять больше внимания творческому подходу в учебном процессе.

Важным аспектом работы преподавателя является проверка домашних заданий или ответов по контрольным работам. Все сводится к тому, что преподаватель сверяет ответы учащихся с верными ответами, проверяет рукописи и выставляет баллы. Автоматизация данного процесса облегчит работу преподавателя и позволит правильно оценивать возможности учащихся.

Уже давно существуют программы позволяющие проверять выполнения заданий требующих четко сформулированного ответа в виде слова, числа или их последовательности. Например, подобная программа используется для оценки первой части единого государственного экзамена (ЕГЭ). Но основной проблемой является проверка рукописных ответов, здесь с трудностями сталкиваются даже преподаватели и эксперты, поэтому целесообразно будет использовать ИИ. Такой программой на основе нейросети является Gradescore, которая позволяет проверят тестовые и письменные задания по различным предметам. В самом начале работы, преподаватель загружает в систему задания и правильные ответы, а затем сканирует листы бумаги с выполненными заданиями от обучающихся. Нейросеть распознает рукописный текст, сопоставляет с правильными ответами, находит ошибки и группирует их, чтобы преподаватель мог дать обратную связь по слабым местам студентов.

Персонализация обучения. Каждый учащийся имеет разный уровень подготовки и разные способности к обучению, кто-то обладает аналитическим складом ума, а кто-то имеет уклон к творчеству. У любого учащегося есть его индивидуальные характеристики, поэтому для достижения наилучшего результата от процесса обучения необходимо осуществлять персональный подход к обучению. Персонализация обучения – это процесс разработки индивидуальных образовательных программ, которые создаются с учетом опыта, интересов, предпочтительных темпов и способов освоения знаний для конкретного обучающегося.

Индивидуальный подход к учащимся требует анализа их знаний и предпочтений, а это требует от преподавателя немалые усилия по оценке этих знаний и подготовке индивидуальных программ обучения для каждого студента или ученика. В данной работе

хорошо себя показал искусственный интеллект, он позволяет создавать уникальные учебные программы, подходящие для каждого обучающегося. Алгоритмы машинного обучения анализируют данные обучающего процесса и способности учащегося, определяя оптимальные методы обучения и темп усвоения материала.

ИИ позволяет определять уровень подготовки ученика и создавать для него персонализированный план обучения. Например, Duolingo – приложение для изучения иностранных языков. До начала обучения вы проходите тест по знанию языка, и приложение собирает упражнения подходящей для вас сложности. По мере прохождения обучения Duolingo анализирует ваш прогресс и корректирует программу обучения, помогает заполнять пробелы и уделяет больше внимания интересующим вас темам. Также в приложение встроена функция распознавания речи для практических занятий навыков произношения.

Аналитика и прогнозирование. Для выявления закономерностей в успеваемости учащихся и правильной оценки их уровня подготовки необходимо анализировать большие объемы данных об учебном процессе. Использование ИИ позволяет осуществлять анализ с невероятной скоростью и повышенной точностью. Подобное решение встречается в рассмотренных ранее сервисе для создания тестов PrepAI и программе для проверки выполненных заданий Gradescope.

Обе программы анализируют выполненные задание, группируют ответы и выявляют часто допускаемые ошибки. Преподаватель получает доступ к готовым аналитическим данным и может правильно оценивать способности каждого учащихся. Также на основе данных можно уделять больше времени изучению тех тем, которые вызвали наибольшие затруднения у обучающихся.

Отрицательная сторона использования ИИ в образовании. Искусственный интеллект помогает улучшать качество образования, повышает эффективность и ускоряет все процессы. Но есть вероятность возникновения некоторых проблем, которые могут проявляться при использовании искусственного интеллекта:

- Проблема безопасности персональных данных. Защита персональных данных, несмотря на актуальность проблемы, остается сложным и чувствительным процессом. Использование информационных технологий для обработки персональных данных открывает новые возможности для злоумышленников, которые хотят их получить, поэтому нужно уделять особое внимание безопасности. Искусственный интеллект является относительно новой технологией, что делает ее наиболее привлекательной для взлома.
- Опасность потери контроля над образовательным процессом. Использование ИИ для создания индивидуальных учебных программ может привести к тому, что учащиеся будут получать знания только в определенных предметных областях, а это может нивелировать разнообразие в учебном процессе [6]. Также это может пагубно повлиять на мотивацию обучающихся и их интерес к получению новых знаний. Автоматизация некоторых аспектов процесса обучения может негативно влиять на отношения преподавателя и учащихся, которые могут потерять доверие к компетентности педагога.
- Недостаточная прозрачность искусственного интеллекта. ИИ эффективен в решении определенных задач, но он может привести и к негативным последствиям для обучения. Например, преподаватели могут перестать оценивать своих учеников и полностью довериться оценке ИИ, что может привести к некорректным итоговым баллам и дискриминации некоторых учеников.

Ограничения и возможности ИИ в образовании. Нейросети (разновидность искусственного интеллекта) получили широкую популярность среди студентов и

школьников, они позволяют генерировать уникальные тексты и писать красивые сочинения, выполнять сложные домашние задания, создавать презентации или даже писать курсовые и дипломные работы. Например, студент Александр Жадан успешно защитил дипломную работу, которую он написал с использованием текстовой нейросети ChatGPT, что вызвало большой резонанс среди общественности.

В некоторых странах уже принимаются меры на законодательном уровне, ограничивающие использование нейросетей студентами. В нескольких штатах США не разрешают использовать нейросети в учебных работах, т.к. существует вероятность негативного воздействия на обучение. Аналогичное решение приняли в Японии и Италии, где нейросети запретили полностью.

Решения на основе искусственного интеллекта могут использоваться далеко не для всех задач. В ходе опроса, который провела HolonIQ – исследовательская группа в сфере обучения и образования, выяснилось, что обучение школьников и дошкольников на основе ИИ-решений не имеет достаточного потенциала.

Искусственный интеллект в образовании может стать толчком к созданию гибких и доступных образовательных систем, что должно привести к повышению уровня образования и обучаемости в мире. Также ИИ-системы прогнозирования откроют возможности анализировать рынок труда и перестраивать учебную программу, чтобы студенты могли получить востребованную будущую профессию.

\*\*\*

1. А. Р. Гапсаламов. Экономика и образование в эпоху цифровизации: монография / А. Р. Гапсаламов, Т. Н. Бочкарева, В. Л. Васильев, Э. М. Ахметшин. — Москва: Ай Пи Ар Медиа, 2022. — 111 с. — ISBN 978-5-4497-1749-8. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/122650.html>.
2. Азевич, А. И. Информационные технологии обучения. Теория. Практика. Методика: учебное пособие по курсам «Использование современных информационных и коммуникационных технологий в учебном процессе» и «Аудиовизуальные технологии обучения» для студентов, обучающихся по специальностям «Логопедия», «Олигофренопедагогика», «Сурдопедагогик» / А. И. Азевич. — Москва: Московский городской педагогический университет, 2010. — 216 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/26492.html>.
3. Джеймс Баррат. Последнее изобретение человечества: искусственный интеллект и конец эры Homo sapiens / Баррат Джеймс; перевод Н. Лисова; под редакцией А. Никольского. — Москва: Альпина нон-фикшн, 2019. — 312 с. — ISBN 978-5-91671-436-4. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/86821.html>.
4. Жасмина Вана. Искусственный интеллект отвечает на величайшие вопросы человечества. Что делает нас людьми? — Жасмина Вана, Иэн С. Томас, GPT-3; перевод:

**Шевко Н.Р., Миронова Е.А.**

**Особенности обеспечения кибербезопасности в современных условиях**

*Всероссийский государственный университет юстиции (РПА)  
(Россия, Казань)*

*doi: 10.18411/trnio-12-2024-894*

#### **Аннотация**

Вопросы обеспечения информационной безопасности являются актуальными в современном обществе. В условиях компьютеризации всего мира, перехода большинства ресурсов в цифровую среду требует особого подхода к защите информации. Переход подавляющего большинства предприятий на удаленную работу в период пандемии COVID-19 обозначил необходимость более надежной защиты информационных ресурсов от реальных и потенциальных угроз. В данной статье авторами дано понятие кибербезопасности, выделены ее аспекты, а также сформулированы базовые требования к обеспечению надежной защиты информационных ресурсов.

**Ключевые слова:** защита информации, кибербезопасность, информационные ресурсы, удаленная работа, угрозы, информационные риски.

### Abstract

Issues of ensuring information security are relevant in modern society. In the context of computerization of the whole world, the transition of most resources to the digital environment requires a special approach to information protection. The transition of the vast majority of enterprises to remote work during the COVID-19 pandemic has highlighted the need for more reliable protection of information resources from real and potential threats. In this article, the authors give the concept of cybersecurity, highlight its aspects, and formulate the basic requirements for ensuring reliable protection of information resources.

**Keywords:** information protection, cybersecurity, information resources, remote work, threats, information risks.

Кибербезопасность – это комплекс мер и технологий, направленных на защиту компьютерных систем, сетей, программ и данных от несанкционированного доступа, изменения или уничтожения. В современном мире кибератаки становятся все более частыми и сложными, поэтому важность кибербезопасности постоянно возрастает.

Основные аспекты кибербезопасности включают:

1. защита информации - обеспечение конфиденциальности, целостности и доступности данных;
2. сетевая безопасность - защита сетевых ресурсов и коммуникаций от внешних угроз;
3. безопасность приложений - разработка и внедрение безопасных программных решений;
4. операционная безопасность - управление доступом к системам и контроль за действиями пользователей;
5. физическая безопасность - охрана серверов, компьютеров и других физических компонентов инфраструктуры;
6. обучение персонала - повышение осведомленности сотрудников о рисках и методах защиты;
7. мониторинг и реагирование - постоянный мониторинг активности в сети и оперативное реагирование на инциденты;
8. соответствие стандартам и законодательству - соблюдение нормативных требований и стандартов (например, GDPR).

Для обеспечения кибербезопасности используются различные инструменты и методы, такие как антивирусные программы, межсетевые экраны, шифрование данных, двухфакторная аутентификация и регулярные обновления программного обеспечения.

Важно помнить, что кибербезопасность требует комплексного подхода и постоянного внимания со стороны всех участников процесса – от разработчиков до конечных пользователей.

Удалённая работа (или дистанционная работа) — это форма организации труда, при которой сотрудники выполняют свои обязанности вне офиса, часто находясь дома или в любом другом удобном для них месте. Эта модель работы стала особенно популярной после пандемии COVID-19, когда многие компании были вынуждены перевести своих сотрудников на удалённый режим.

В условиях удалённой работы кибербезопасность становится особенно важной задачей для компаний и сотрудников. В основном сотрудники, которые работают удаленно, используют свой личный компьютер для выполнения тех или иных корпоративных задач. Многие не уделяют должного внимания обеспечению информационной безопасности и антивирус, установленный на компьютере, не обновляющийся под новые информационные угрозы, подвергает компьютер пользователя, а, следовательно, и работу всей компании угрозам.

К примеру, компьютер, который используется сотрудником для выполнения задач компании не должен быть использован для личных нужд ввиду уязвимости

информационной системы при подключении с сети общего пользования и скачивания различных материалов. Встречаются и такие моменты, когда пользователя просят пройти по ссылке в мессенджере под предлогом проголосовать в конкурсе за какого-то родственника. При переходе по ссылке можно потерять доступ к компьютеру или потерять информацию на нем.

Если при скачивании файла стоит формат в виде .exe файла, скорее всего это вредоносное программное обеспечение, при установке которого пользователь может столкнуться со следующими проблемами:

- на рабочий стол выйдет баннер с требованием перевести денежные средства для разблокировки системы;
- при распаковке данного .exe-файла злоумышленник может получить доступ к данным вашего компьютера;
- злоумышленник может получить доступ на управление вашим компьютером удаленно.

Еще одним наиболее известным видом киберпреступления является спам-рассылка от неизвестного (и даже несуществующего) отправителя. При переходе по ссылке в подобной ситуации осуществится переход на фишинговый сайт либо установится вредоносное программное обеспечение.

Для того, чтобы избежать приведенных выше проблем, стоит уделять особое внимание информационной защите. В частности, современная антивирусная программа с актуальной вирусной базой помогает защитить компьютер от новых вредоносных программ. Причем современные антивирусные программы, например, EsetSmartSecurityPremium, является и антивирусом, способным блокировать установку подозрительных программ и изолировать файлы, и фаерволом (защита) файлов на компьютере в режиме реального времени. Кроме того, имеется функция проактивной защиты компьютера, а именно анализируемые файлы блокируются, пока не будет подтверждена их безопасность. При посещении веб-сайтов данный антивирус блокирует подозрительные сайты. Имеется функция защиты веб-камеры и система предотвращения вторжений на узел (Hips – нежелательное поведение приложений).

Удаленная работа увеличивает риски утечек данных, атак злоумышленников и других киберугроз.

Защитить компьютерную систему от угроз просто необходимо. Для этого:

1. должно быть безопасное подключение к Интернету. А именно – наличие надежного пароля, выполнение всех протоколов шифрования, которые указаны в настройках роутера производителем;
2. не стоит подключаться к бесплатным точкам доступа в кафе, аэропортах и т.д. Такие сети являются потенциальной угрозой безопасности;
3. антивирусное программное обеспечение должно быть лицензированным, а антивирусные базы актуальными. Регулярные обновления закрывают уязвимости, которыми могут воспользоваться злоумышленники;
4. необходимо хранить файлы служебных данных на внешнем носителе, и закрывать доступ к нему, после окончания работы.

Таким образом, удаленная работа требует особого внимания к вопросам кибербезопасности. Важно не только обеспечить техническую защиту, но и регулярно обучаться правилам безопасного поведения в Интернете. Нужно осознавать, к каким рискам может привести халатность в данной теме.

\*\*\*

1. Год красного локдауна — как COVID-19 повлиял на кибербезопасность [Электронный ресурс] // Официальный сайт Лаборатории Касперского – Режим доступа: <https://www.kaspersky.ru/blog/pandemic-year-in-infosec/30316/> (дата обращения: 14.11.2024).
2. Горач Н. Н., Филатова И. В. Вызовы и угрозы информационной безопасности преступлениями, совершаемыми в условиях пандемии COVID-19 // Вестник Московского университета МВД России. 2020. №8. [Электронный ресурс] – Режим доступа: <https://cyberleninka.ru/article/n/vyzovy-i-ugrozy->

- informatisionnoy-bezopasnosti-prestupleniyami-sovershaemyimi-v-usloviyah-pandemii-covid-19 (дата обращения: 14.11.2024).
3. Гюльтабаев Т.А., Алтухов С.А. Проблемы обеспечения кибербезопасности в условиях борьбы с пандемией COVID-19 // Сборник научных трудов. Том 14. ВГУЮ (РПА Минюста России). Ростов-на-Дону, 2021. С. 391-398.
  4. Марков А. Информационная безопасность в условиях пандемии COVID-19 [Электронный ресурс] // Официальный сайт Российского совета по международным делам (РСМД) – Режим доступа: <https://russiancouncil.ru/analytics-and-comments/analytics/informatisionnaya-bezopasnost-v-usloviyakh-pandemii-covid-19/> (дата обращения: 14.11.2024).
  5. Шевко Н.Р. Проблемы защиты персональных данных в условиях цифровой трансформации общества // Образование. Наука. Научные кадры. 2023. № 3. С. 164-166.
  6. Шевко Н.Р. Проблемы обеспечения кибербезопасности // Ученые записки Казанского филиала "Российского государственного университета правосудия". 2023. Т. 19. С. 181-185.

**Эркенова М.У., Аджиев И.Р.**

**Будущее автоматизации в туристическом бизнесе: прогноз развития технологий и тенденций**

*ФГБОУ ВО «Северо-Кавказская государственная академия»  
(Россия, Черкесск)*

*doi: 10.18411/trnio-12-2024-895*

**Аннотация**

Данная статья анализирует текущее состояние автоматизации в туристическом бизнесе и прогнозирует развитие технологий и тенденций в ближайшем будущем. Рассматриваются ключевые технологические драйверы, такие как искусственный интеллект, большие данные, блокчейн и расширенная реальность, а также их влияние на бизнес-процессы и пользовательский опыт. В заключении представлены выводы и рекомендации для участников туристического рынка.

**Ключевые слова:** автоматизация, туризм, искусственный интеллект, большие данные, блокчейн, расширенная реальность, пользовательский опыт, прогнозирование.

**Abstract**

This article analyzes the current state of automation in the tourism business and predicts the development of technologies and trends in the near future. Key technological drivers such as artificial intelligence, big data, blockchain and augmented reality are considered, as well as their impact on business processes and user experience. In conclusion, the conclusions and recommendations for the participants of the tourism market are presented.

**Keywords:** automation, tourism, artificial intelligence, big data, blockchain, augmented reality, user experience, forecasting.

Туристический бизнес всегда был одной из самых динамично развивающихся отраслей экономики. В последние десятилетия наблюдается стремительное внедрение технологий автоматизации, которые меняют подход к организации путешествий, взаимодействию с клиентами и управлению ресурсами. Автоматизация в туристическом бизнесе охватывает широкий спектр технологий, включая системы управления бронированием, чат-боты, искусственный интеллект, анализ больших данных и многое другое. А, в условиях глобализации и цифровизации туристическая отрасль сталкивается с необходимостью адаптироваться к новым реалиям. Современные путешественники становятся все более требовательными, ожидая персонализированного подхода и мгновенного доступа к информации. В ответ на эти вызовы компании в туристическом бизнесе начинают активно внедрять автоматизированные решения, которые позволяют оптимизировать процессы, сократить время ожидания и улучшить качество обслуживания.

Однако автоматизация не только открывает новые горизонты, но и ставит перед туристическим бизнесом ряд вызовов и проблем. Внедрение новых технологий требует значительных инвестиций, а также изменения организационной структуры и корпоративной культуры. Не менее важным является вопрос безопасности данных и защиты личной информации клиентов, что становится особенно актуальным в свете растущего числа кибератак и утечек данных.

Важным аспектом автоматизации является также ее влияние на трудовые ресурсы. С одной стороны, автоматизация может привести к сокращению рабочих мест, особенно в тех областях, где рутинные задачи могут быть выполнены машинами. С другой стороны, она открывает новые возможности для создания высококвалифицированных рабочих мест, связанных с управлением и развитием технологий. В этом контексте необходимо рассмотреть, каким образом компании могут адаптироваться к изменениям на рынке труда и какие стратегии могут быть использованы для повышения квалификации сотрудников.

Текущее состояние автоматизации в туристическом бизнесе:

На сегодня, менеджер по туризму выполняет широкий спектр задач, включая поиск и бронирование билетов, отелей, трансферов, оформление виз, составление туристических программ и коммуникацию с клиентами. Высокая конкуренция на туристическом рынке и растущие требования клиентов требуют оптимизации рабочих процессов и повышения эффективности работы менеджеров. Автоматизация становится ключевым фактором успеха в современной туристической индустрии.

Задачи менеджера по туризму можно разделить на несколько категорий:

В настоящее время к задачи менеджера по туризму можно разделить на несколько категорий, включая:

- Системы бронирования: Онлайн-платформы для бронирования авиабилетов, отелей, автомобилей и других туристических услуг.
- CRM-системы: Управление взаимодействием с клиентами, персонализация предложений, автоматизация маркетинговых кампаний.
- Системы управления контентом: Создание и управление контентом на веб-сайтах и в мобильных приложениях.
- Чат-боты: Автоматизированное общение с клиентами, предоставление справочной информации и обработка простых запросов.

В ближайшем будущем ожидается дальнейшее развитие автоматизации в туристическом бизнесе, обусловленное следующими технологическими трендами:

- Искусственный интеллект (ИИ): ИИ будет играть ключевую роль в персонализации туристических предложений, прогнозировании спроса, оптимизации маршрутов и автоматизации сложных задач. Например, ИИ-системы смогут анализировать предпочтения клиентов и предлагать индивидуальные туристические маршруты, учитывая их интересы, бюджет и время.
- Большие данные: Анализ больших данных позволит туристическим компаниям лучше понимать поведение клиентов, выявлять новые тренды и оптимизировать свои услуги. Анализ данных о бронированиях, предпочтениях и отзывах позволит прогнозировать спрос, оптимизировать ценообразование и улучшать качество обслуживания.
- Блокчейн: Технология блокчейн обеспечит безопасность и прозрачность туристических транзакций, упростит процесс обмена данными между различными участниками туристической экосистемы и защитит персональные данные клиентов.

- Расширенная реальность (XR): XR-технологии (VR, AR, MR) позволят туристам "посетить" места, которые они планируют посетить, виртуально ознакомиться с отелями и достопримечательностями, что повысит уровень вовлеченности и доверия.
- Интеграция различных систем: Дальнейшая интеграция различных систем бронирования, CRM-систем и других платформ обеспечит бесшовный пользовательский опыт и автоматизацию всех этапов туристического процесса.

Развитие автоматизации приведет к:

- Повышению эффективности: Автоматизация рутинных задач освободит время менеджеров для решения более сложных вопросов и общения с клиентами.
- Персонализации пользовательского опыта: ИИ-системы позволят создавать персонализированные туристические предложения, учитывая индивидуальные потребности каждого клиента.
- Улучшению качества обслуживания: Автоматизация позволит туристическим компаниям оперативно обрабатывать запросы клиентов, предоставлять актуальную информацию и решать проблемы.
- Сокращению издержек: Автоматизация позволит оптимизировать бизнес-процессы и сократить затраты на персонал и другие ресурсы.

Будущее автоматизации в туристическом бизнесе связано с активным внедрением ИИ, больших данных, блокчейна и XR-технологий. Успешное использование этих технологий требует инвестиций в инновационные решения, развитие компетенций персонала и адаптацию бизнес-моделей к изменяющимся условиям рынка. Туристическим компаниям необходимо активно внедрять автоматизированные системы, чтобы оставаться конкурентоспособными и удовлетворять растущие потребности клиентов. Важным аспектом является обеспечение безопасности данных и конфиденциальности информации клиентов.

Автоматизация будет играть ключевую роль в формировании будущего туристического бизнеса. Успешное внедрение новых технологий позволит туристическим компаниям повысить свою эффективность, персонализировать услуги и улучшить пользовательский опыт. Однако, необходимо учитывать возможные риски и вызовы, связанные с внедрением инновационных решений, и уделять особое внимание этическим аспектам использования новых технологий.

\*\*\*

1. Богомазова Ирина Викторовна, Климова Татьяна Брониславовна ЦИФРОВЫЕ СЕРВИСЫ И ТУРИСТСКАЯ ЭКОСИСТЕМА В РАЗВИТИИ ВНУТРЕННЕГО ТУРИЗМА // Экономика. Информатика. 2022. №4. URL: <https://cyberleninka.ru/article/n/tsifrovye-servisy-i-turistskaya-ekosistema-v-razviti-i-vnutrennego-turizma> (дата обращения: 04.12.2024).
2. Герман У., Калишевский С. Перспективы использования искусственного интеллекта в управлении персоналом туристского предприятия // Российские регионы: взгляд в будущее. – 2024. – Т. 11. – №. 1-2. – С. 82-91. URL: <https://cyberleninka.ru/article/n/perspektivy-ispolzovaniya-iskusstvennogo-intellekta-v-upravlenii-personalom-turistskogo-predpriyatiya> (дата обращения: 04.12.2024).
3. Девяткина Д. Г. и др. исследование влияния искусственного интеллекта на трудовые рынки // журнал прикладных исследований. – 2023. – С. 51. URL: [https://admin.pegaspress.ru/uploads/Zh\\_PI\\_2023\\_S2\\_53745b45dd. =51](https://admin.pegaspress.ru/uploads/Zh_PI_2023_S2_53745b45dd. =51) (дата обращения: 04.12.2024).

Эркенова М.У., Джибабов М.М.

**Создание персонализированных учебных планов с помощью машинного обучения для изучения родного языка**

ФГБОУ ВО «Северо-Кавказская государственная академия»

(Россия, Черкесск)

doi: 10.18411/trnio-12-2024-896

**Аннотация**

В данной статье рассматриваются возможности использования машинного обучения для создания персонализированных учебных планов для обучения родному языку. Анализируются различные подходы, включая адаптивное обучение и рекомендательные системы. Актуальность исследования обусловлена необходимостью адаптации образовательных процессов к индивидуальным потребностям и особенностям учащихся. Статья подчеркивает значимость интеграции технологий машинного обучения в учебный процесс и предлагает рекомендации по их применению для улучшения качества обучения родному языку.

**Ключевые слова:** персонализированное обучение, машинное обучение, родной язык, адаптивное обучение, системы обучения.

**Abstract**

This article discusses the possibilities of using machine learning to create personalized curricula for teaching the native language. Various approaches to the construction of such trajectories, including adaptive learning and recommendation systems, are analyzed. The relevance of the research is due to the need to adapt educational processes to the individual needs and characteristics of students. The article highlights the importance of integrating machine learning technologies into the learning process and offers recommendations on their application to improve the quality of native language learning.

**Keywords:** personalized learning, machine learning, native language, adaptive learning, learning systems.

Современное общество сталкивается с множеством вызовов в области образования, особенно когда речь идет о изучении родного языка. В условиях глобализации и стремительного развития технологий, традиционные методы обучения часто оказываются недостаточно эффективными для удовлетворения индивидуальных потребностей учащихся. В этом контексте важным направлением становится создание персонализированных учебных планов, которые могут адаптироваться к уникальным особенностям каждого обучающегося.

Обучение родному языку – сложный процесс, требующий индивидуального подхода. Традиционные методы обучения часто не учитывают специфические потребности и особенности каждого учащегося, что может приводить к снижению эффективности обучения. Актуальность данной статьи обусловлена необходимостью поиска новых подходов к обучению родного языка с использованием инновационных технологий. Машинное обучение предоставляет новые возможности для персонализации учебного процесса, позволяя создавать индивидуальные учебные планы, адаптированные к уровню знаний, стилю обучения и предпочтениям каждого учащегося.

Традиционные учебные планы разрабатывались с акцентом на единые стандарты и требования, что зачастую не учитывало индивидуальные потребности учащихся. Стандартизированные подходы к обучению, основанные на фиксированных программах и оценках, не могли обеспечить необходимую гибкость для разнообразия обучающихся.

Подходы к построению персонализированных учебных траекторий:

Существуют различные подходы к построению персонализированных учебных траекторий с помощью машинного обучения:

- Адаптивное обучение: Один из наиболее распространенных методов индивидуализации образования — это адаптивное обучение. Этот подход предполагает использование технологий для создания учебных программ, которые автоматически подстраиваются под уровень знаний и навыков учащегося. Такой подход позволяет не только улучшить понимание материала, но и повысить мотивацию учащихся, так как они получают возможность работать в своем собственном темпе и сосредоточиться на тех аспектах, которые требуют дополнительного внимания. Система непрерывно отслеживает успеваемость учащегося и динамически корректирует сложность заданий и темпу обучения. Алгоритмы машинного обучения анализируют ответы учащегося и определяют его сильные и слабые стороны, адаптируя дальнейший учебный материал.
- Рекомендательные системы: Система рекомендует учащемуся конкретные учебные материалы, упражнения и задания на основе его предыдущей активности и предпочтений. Рекомендации могут быть основаны на коллаборативной фильтрации (рекомендации на основе предпочтений других учащихся с похожим уровнем знаний) или контент-based фильтрации (рекомендации на основе характеристик учебных материалов).
- Гибридные подходы: Сочетание адаптивного обучения и рекомендательных систем позволяет создавать более эффективные и персонализированные учебные траектории.

Архитектура системы персонализированного обучения:

Предлагаемая архитектура системы персонализированного обучения включает в себя следующие компоненты:

1. Система сбора данных: Собирает данные о действиях учащегося, включая ответы на задания, время выполнения заданий, использование различных функций платформы.
2. Модуль обработки данных: Предварительная обработка данных, очистка данных от шума и преобразование данных в формат, пригодный для использования алгоритмами машинного обучения.
3. Модуль построения модели: Обучение моделей машинного обучения на основе собранных данных. Можно использовать различные алгоритмы, такие как деревья решений, случайный лес, нейронные сети.
4. Модуль генерации рекомендаций: На основе обученной модели генерируются рекомендации по учебным материалам и заданиям для каждого учащегося.
5. Модуль оценки эффективности: Оценка эффективности персонализированных учебных траекторий путем сравнения успеваемости учащихся, обучающихся по персонализированным и традиционным программам.

К основным преимуществам можно отнести:

- Повышение эффективности обучения: Персонализированные учебные траектории позволяют учащимся учиться в оптимальном темпе и фокусироваться на наиболее сложных для них аспектах языка.
- Повышение мотивации: Индивидуальный подход и адаптация к предпочтениям учащихся повышают их мотивацию и вовлеченность в учебный процесс.

- Улучшение качества знаний: Более целенаправленное обучение приводит к более глубокому усвоению материала.
- Качество данных: Эффективность системы зависит от качества и количества данных, собранных о действиях учащихся.
- Выбор алгоритмов МО: Выбор подходящего алгоритма машинного обучения зависит от конкретных задач и доступных данных.
- Интерпретация результатов: Результаты работы алгоритмов машинного обучения должны быть интерпретированы педагогами для корректировки учебного процесса.
- Обеспечение конфиденциальности данных: Необходимо обеспечить безопасность и конфиденциальность данных учащихся.

На сегодняшний день можем с уверенностью сказать, что машинное обучение открывает новые возможности для создания персонализированных учебных планов в обучении родному языку. Разработка эффективных систем требует тщательного анализа данных, выбора подходящих алгоритмов машинного обучения и учёта этических аспектов.

Также стоит отметить, что индивидуализация образования должна учитывать культурные и социальные аспекты. В контексте изучения родного языка важно учитывать культурные особенности и традиции, которые могут влиять на процесс обучения. Это требует от преподавателей глубокого понимания контекста, в котором они работают, и способности адаптировать свои учебные планы в соответствии с потребностями и интересами учащихся.

Дальнейшие исследования должны быть направлены на улучшение качества данных, разработку новых алгоритмов машинного обучения и создание интуитивно понятных интерфейсов для учащихся и педагогов. Персонализированное обучение с помощью машинного обучения имеет большой потенциал для повышения эффективности и качества обучения родному языку.

\*\*\*

1. Андреева А. А., Калмыкова С. В. Практические аспекты применения средств искусственного интеллекта в онлайн-курсах // Педагогический журнал. – 2023. – Т. 13. – №. 8А. – С. 305.
2. Бритвина П. В. ВЛИЯНИЕ МАШИННОГО ОБУЧЕНИЯ НА ПЕРСОНАЛИЗАЦИЮ ОБРАЗОВАТЕЛЬНЫХ ПРОЦЕССОВ // Вестник науки. 2024. №1 (70). URL: <https://cyberleninka.ru/article/n/vliyanie-mashinnogo-obucheniya-na-personalizatsiyu-obrazovatelnyh-protsessov> (дата обращения: 04.12.2024).
3. ВИШНЯКОВ Ю. С. и др. СОДЕРЖАНИЕ ОБРАЗОВАНИЯ. URL: <https://elibrary.ru/item.asp?id=49914817> (дата обращения: 04.12.2024).

**Эркенова М.У., Хачиров И.А.**

### **Сравнительный анализ существующих CRM-систем и возможности их адаптации на 1С: Предприятие 8.3**

*ФГБОУ ВО «Северо-Кавказская государственная академия»  
(Россия, Черкесск)*

*doi: 10.18411/trnio-12-2024-897*

#### **Аннотация**

В данном исследовании проведён сравнительный анализ существующих CRM систем с акцентом на возможность их адаптации и интеграции с платформой 1С: Предприятие 8.3. В условиях растущей конкуренции на рынке бизнеса, эффективное управление взаимоотношениями с клиентами становится ключевым фактором успеха. Анализ включает в себя как облачные, так и локальные CRM-системы, учитывающие различные аспекты, такие как функциональные возможности, стоимость внедрения, простота использования и уровень интеграции с другими бизнес-процессами. В ходе

работы изучены лучшие практики адаптации CRM-систем к платформе 1С. Обращено внимание на преимущества и недостатки каждого решения, а также рассмотрены потенциальные трудности, возникающие в процессе интеграции. Результаты исследования показывают, что успешная адаптация CRM на базе 1С: Предприятие 8.3 требует комплексного подхода, включающего кастомизацию бизнес-процессов и обучение пользователей.

**Ключевые слова:** CRM-системы, 1С: Предприятие 8.3, адаптация CRM.

### Abstract

This study provides a comparative analysis of existing CRM systems with an emphasis on the possibility of their adaptation and integration with the 1С: Enterprise 8.3 platform. In the context of growing competition in the business market, effective customer relationship management is becoming a key success factor. The analysis includes both cloud and on-premises CRM systems that take into account various aspects such as functionality, cost of implementation, ease of use and the level of integration with other business processes. In the course of the work, the best practices of adapting CRM systems to the 1С platform were studied. Attention is drawn to the advantages and disadvantages of each solution, as well as the potential difficulties encountered in the integration process. The results of the study show that the successful adaptation of CRM based on 1С: Enterprise 8.3 requires an integrated approach, including customization of business processes and user training.

**Keywords:** CRM systems, 1С: Enterprise 8.3, CRM adaptation.

### Введение

Рынок CRM-систем предлагает широкий выбор решений, от облачных сервисов до сложных систем. Выбор оптимальной системы зависит от специфики бизнеса, размера компании и бюджета. Однако, многие организации, уже работающие на платформе 1С:Предприятие 8.3, рассматривают возможность интеграции или адаптации CRM-функционала именно в рамках этой платформы. В данной статье мы проведем сравнительный анализ популярных CRM-систем и оценим потенциал их адаптации на 1С:Предприятие 8.3.

Для начала рассмотрим, определение CRM. CRM (Customer Relationship Management) можно рассматривать с различных точек зрения. В первую очередь, это стратегия управления взаимоотношениями с клиентами, которая основывается на анализе данных о клиентах и их взаимодействии с компанией. CRM-системы позволяют собирать и хранить информацию о клиентах, их предпочтениях, истории покупок и взаимодействиях с компанией, что в свою очередь позволяет более точно настраивать маркетинговые кампании, улучшать качество обслуживания и предлагать персонализированные решения. Таким образом, CRM становится неотъемлемой частью бизнес-стратегии, направленной на создание долгосрочных отношений с клиентами [1].

При сравнении CRM-систем следует учитывать такие параметры, как стоимость, функциональность, степень интеграции, удобство использования и возможность кастомизации. Например, системы, такие как Salesforce, предлагают мощные инструменты для крупных корпораций, но могут быть излишне сложными для малых и средних предприятий. В то же время, Vitrix24 могут предложить более доступные решения, но могут не предоставить всех необходимых функций для более сложных бизнес-процессов.

Для сравнения выберем несколько популярных CRM-систем, представляющих разные подходы:

Salesforce: Лидер рынка, облачная CRM-система с широким функционалом, высокой масштабируемостью и гибкой настройкой. Однако, обладает высокой стоимостью и требует определенных навыков для настройки и администрирования.

- Плюсы: Высокая степень кастомизации, мощные аналитические инструменты, интеграция с большим количеством сервисов.

- Минусы: Высокая стоимость, необходимость обучения сотрудников.

**Bitrix24:** Популярная облачная CRM-система с хорошим соотношением цены и функционала, удобная для малого и среднего бизнеса. Предлагает широкий спектр инструментов, включая автоматизацию маркетинга и коммуникаций. Интеграция с другими системами, в том числе и 1С, возможна, но может потребовать дополнительных настроек.

- Плюсы: Бесплатная версия, комплексное решение (включает задачи, проекты, чат и др.), доступный интерфейс.
- Минусы: Ограниченные возможности в бесплатной версии, недостаточная гибкость для крупных компаний.

**AmoCRM:** Еще одна облачная CRM, отличающаяся интуитивно понятным интерфейсом и удобством использования. Хорошо подходит для команд продаж, ориентированных на воронку продаж. Интеграция с 1С также возможна, но требует отдельной настройки.

- Плюсы: Доступная цена, наличие множества встроенных инструментов для аналитики.
- Минусы: Ограниченные возможности при интеграции с другими системами

**HubSpot CRM:** Бесплатная облачная CRM-система с расширенным бесплатным функционалом, ориентированная на привлечение и взаимодействие с лидами. Хорошо интегрируется с другими сервисами HubSpot, но интеграция с 1С потребует дополнительных решений.

На рисунке 1 видим основные возможности адаптации CRM-функционала на 1С:Предприятие 8.3:

Характеристика	Salesforce	Bitrix24	AmoCRM	HubSpot CRM	1С:Предприятие 8.3 (с доработками)
Стоимость	Высокая	Средняя	Средняя	Бесплатный вариант с платными расширениями	Зависит от масштаба доработок
Функционал	Очень широкий	Широкий	Средний	Средний, с упором на маркетинг	Зависит от доработок, может быть любым
Масштабируемость	Высокая	Средняя	Средняя	Средняя	Высокая
Облачная/Локальная	Облачная	Облачная	Облачная	Облачная	Локальная
Интеграция с 1С	Возможна, но сложна	Возможна, требует настройки	Возможна, требует настройки	Требует дополнительных решений	Нативная интеграция
Удобство использования	Среднее	Высокое	Высокое	Высокое	Зависит от доработок

Рисунок 1. Возможности адаптации CRM-функционала на 1С:Предприятие 8.3.

1С:Предприятие 8.3 обладает мощными возможностями для разработки собственных CRM-решений. Это позволяет создавать системы, полностью соответствующие специфическим требованиям бизнеса. Преимущества такого подхода:

Преимущества адаптации CRM на 1С:

1. Интеграция с существующими бизнес-процессами: Адаптация CRM на 1С позволяет использовать данные из других систем и упростить обмен информацией между отделами.
2. Кастомизация: Пользователи могут создавать собственные модули, отвечающие специфическим требованиям бизнеса.
3. Управление данными: Вся информация о клиентах и процессах может храниться в одной системе, что упрощает доступ и аналитику.
4. Автоматизация процессов: 1С позволяет автоматизировать многие задачи, такие как выставление счетов, отправка напоминаний клиентам и анализ продаж.

5. Облачные решения: 1С также предлагает облачные решения, что позволяет получать доступ к системе из любого места.

Возможные недостатки:

Требования к разработке: Адаптация CRM требует наличия технических специалистов, что может увеличить затраты.

Время на внедрение: Процесс адаптации может занять значительное количество времени, в зависимости от сложности бизнес-процессов.

В результате проведенного сравнительного анализа существующих CRM-систем стало очевидно, что выбор подходящего решения зависит от множества факторов, включая требования бизнеса, бюджет, масштабируемость и интеграционные возможности. Многие CRM-системы предлагают обширный функционал для управления отношениями с клиентами, автоматизации бизнес-процессов, а также создания отчетности и аналитики. Однако каждая система имеет свои уникальные особенности и сильные стороны.

Тем не менее, при выборе CRM для адаптации на 1С важно учитывать не только технические возможности, но и потребности конечных пользователей. Удобный интерфейс, поддержка мобильных устройств и наличие обучающих материалов сыграют ключевую роль в успешной имплементации. Также стоит отметить, что дальнейшее развитие системы и возможность быстрого реагирования на изменения в бизнесе будут определяющими факторами ее успешности.

Заключение

Таким образом, критерии выбора CRM-системы должны быть основаны на глубоких знаниях бизнес-процессов компании и стратегических целях. Адаптация в 1С: Предприятие 8.3 позволяет значительно повысить эффективность работы с клиентами, однако требует тщательного анализа и планирования. В конечном итоге, правильно выбранная и адаптированная CRM-система станет не только инструментом для повышения продаж, но и незаменимым помощником в построении долгосрочных отношений с клиентами, что является основой успешного бизнеса.

\*\*\*

1. 1С: Предприятие 8.3. Документация разработчика. [Ссылка на ресурс]
2. Базаров, В. И. "Управление клиентскими отношениями на основе CRM-систем". Издательство: Научный мир, 2020.
3. Петров, С. А. "Автоматизация бизнес-процессов на платформе 1С". Издательство: ЭконоТех, 2021.

**Юданов Р.С.**

### **Вопросы информационной безопасности web-приложений**

*Санкт-Петербургский государственный университет телекоммуникаций  
им. проф. М.А. Бонч-Бруевича  
(Россия, Санкт-Петербург)*

doi: 10.18411/trnio-12-2024-898

#### **Аннотация**

В статье рассматриваются актуальные вопросы безопасности веб-приложений. С ростом сложности и функциональности веб-сайтов, а также с увеличением их популярности, обеспечение безопасности становится критически важным аспектом, требующим внимания как со стороны разработчиков, так и со стороны пользователей. Основные меры защиты должны быть на стороне сервера, но не следует забывать и о знаниях по безопасности, которые необходимы фронтенд-разработчикам.

**Ключевые слова:** Веб-приложение, фронтенд, сервер, вредоносный код, уязвимость, атака, SQL, XSS, CSRF, DoS.

**Abstract**

The article discusses current issues of web application security. With the increasing complexity and functionality of websites, as well as their increasing popularity, security is becoming a critical aspect that requires attention from both developers and users. The main security measures should be on the server side, but we should not forget about the security knowledge that front-end developers need.

**Keywords:** Web application, frontend, server, malicious code, vulnerability, attack, SQL, XSS, CSRF, DoS.

Веб-приложение – это программа, которая доступна через веб-браузер, связана с веб-сервером и позволяет пользователям взаимодействовать с данными и функционалом приложения.

В современном мире работа большинства компаний тесно связана с веб-приложениями. Раньше в области безопасности больше акцент ставился на безопасность сетей и серверов. Сейчас, когда web-приложения становятся сложнее и функциональнее, требуется уделять все большее внимание их безопасности.

Виды и число атак растут в геометрической прогрессии. Атаки можно условно поделить на две категории угроз ИБ:

- нарушение конфиденциальности информации;
- нарушение доступности информации.

Наиболее распространенная угроза безопасности web-приложений — это эксплуатация уязвимостей, а при популяризации приложения в интернете — не избежать и DDoS-атак. Для взлома и вывода приложения из строя могут использоваться различные инструменты как любительские, так и профессиональные, так же могут использоваться автоматические системы сканирования для эксплуатации уязвимостей [1].

Список возможных уязвимостей веб-приложения, которыми могут воспользоваться злоумышленники:

1. инъекции SQL (внедрение SQL-кода);
2. XSS (Cross-Site Scripting — межсайтовый скриптинг);
3. DoS (denial-of-service attack — отказ в обслуживании);
4. CSRF (Cross-Site Request Forgery — межсайтовая запросная подделка).

Веб-приложения осуществляют запросы к данным, хранящимся в базах данных, в то время как пользователи взаимодействуют с этими данными через различные формы для ввода и отображения информации. Инъекции SQL возможны, когда разработчики не учли того, что в поля ввода могут быть введены данные, которые сервер может воспринять как команду для выполнения определенных действий. В результате это может нарушить стандартную работу сервера [2].

XSS уязвимости позволяют внедрить вредоносный код, чаще всего JavaScript, в веб-приложение, которое потом выполняется в браузере других пользователей. Атака не затрагивает серверную часть, но напрямую влияет на клиентскую. С ее помощью нельзя повлиять на данные, хранящиеся на сервере, но можно, например, украсть cookies (небольшой набор данных, отправляемый веб-сервером и хранящийся в веб-браузере пользователя без изменений и какой-либо обработки), которые часто разработчики используют для хранения идентификатора пользователя, позволяющего отличить одного авторизованного пользователя от другого. Получив такой идентификатор, злоумышленник может подставить его себе. Если разработчик не учел дополнительную защиту, то злоумышленник может войти в приложение под учетной записью другого пользователя.

Наиболее распространены 3 типа XSS уязвимостей:

- Stored XSS — вредоносный код загружается на странице из базы данных сайта. Например, если злоумышленник оставит комментарий с

- вредоносным кодом, который попадет в базу данных и будет выполнен у всех пользователей, посетивших сайт;
- Reflected XSS — вредоносный код попадает жертве через ссылку. Например, если злоумышленник передаст ссылку лично, через почту или в переписке, в которой будут содержаться данные, используемые сервером для формирования страницы;
  - DOM-based XSS — вредоносный код встраивается в DOM дерево веб-приложения. Так как современные приложения в своем большинстве, это SPA, то XSS на основе DOM встречается чаще всего. Например, вредоносный код попадает жертве через ссылку, js достает query параметры и добавляет их на страницу пользователя.

DoS-атаки и DDoS-атаки — не одно и то же. Они обе направлены на перегрузку системы, с помощью сетевых запросов. Разница лишь в том, что при DoS-атаке используется один хост для перегрузки системы и это недостаточно эффективно, так как сервера имеют ограничения на отправку одновременных запросов. При DDoS-атаке злоумышленники организуют специальную сеть с вредоносным ПО, называемую “ботнет”. Входящие в инфраструктуру “ботнет” компьютеры не связаны между собой. Они используются для генерации избыточного трафика, способного перегрузить систему [3].

CSRF уязвимость позволяет злоумышленнику выполнить запрос к серверу на авторизованном пользователем сервисе, отправив ему ссылку на свой вредоносный сайт. Например, пользователь выполняет банковский перевод, оставил транзакцию незавершенной и проверяет свою почту на новой вкладке. Там он увидел URL с привлекательной акцией, отправленной злоумышленником. Теперь, когда пользователь открывает данный URL в новой вкладке, там его ждет форма захвата. Страница транзакции осталась на первой вкладке, а эта вредоносная — открыта на второй. Как только пользователь нажимает на кнопку “Submit”, выполняется запрос. Указанная сумма будет списана со счета без его ведома. Хотя CSRF-атаки не предназначены для кражи каких-либо конфиденциальных данных, поскольку злоумышленник не получает никакого ответа, они отличаются тем, что могут изменять данные на сервере.

Не только сам разработчик веб-приложения может упустить уязвимость, но и тот чью библиотеку он использовал. В последнее время остро встал вопрос о внедрении вредоносного кода в популярные библиотеки. К сожалению, много примеров того, как те или иные библиотеки попадают в списки с высоким уровнем риска. Зачастую разработчик не может исправить код, который предоставляет библиотека. Избежать опасных ситуаций, связанных с использованием общедоступных библиотек, несложно [4]. Достаточно:

1. проверять надежность источников;
2. актуальность версий библиотек и
3. сами библиотеки на наличие уже выявленных уязвимостей на различных ресурсах.

Встречаются проблемы с невалидными перенаправлениями, хоть в большинстве современных браузеров уже встроена защита от этой уязвимости. Как показывает практика, не все пользователи своевременно обновляют свои браузеры, поэтому эта уязвимость все еще может быть актуальной, хоть уже и не входит в OWASP Top-10 [4].

В заключение стоит сказать, что большинство современных фреймворков и библиотек уже имеет встроенные механизмы защиты от уязвимостей. И если учитывать, что основная защита выстраивается преимущественно на стороне сервера, что логично, ведь именно там хранятся конфиденциальные данные пользователей и именно серверная сторона отправляет на клиентскую HTTP-заголовки безопасности, у которых функционал гораздо шире, чем тот, что можно выстроить на фронтенде. Всё же frontend-разработчику

никогда не будет лишним знание и понимание базовых механизмов защиты, хоть и часто тема безопасности обходится frontend-разработчиками стороной.

\*\*\*

1. ГАРДА. Информационная Безопасность Web-Приложений - Современные Решения [Электронный ресурс]. 2024. URL: <https://gardatech.ru/articles/smi/informatsionnaya-bezopasnost-web-prilozheniy-sovremennyye-resheniya>.
2. Habr. Безопасность веб-приложений: анализ методов защиты от атак на уровне Backend. [Электронный ресурс]. 2024. URL: <https://habr.com/ru/articles/800017>.
3. Selectel. DDoS-атаки — что это: происхождение, виды и способы защиты. [Электронный ресурс]. 2024. URL: <https://selectel.ru/blog/ddos-attacks>.
4. Habr. Фантастические веб-уязвимости и где они обитают. [Электронный ресурс]. 2022. URL: <https://habr.com/ru/companies/simbirsoft/articles/659847>.

**Dudak A.**

### **Optimization of loading and performance in SPA on React**

*Tomsk State University of Control Systems and Radioelectronics  
(Russia, Tomsk)*

*doi: 10.18411/trnio-12-2024-899*

#### **Abstract**

This article focuses on performance optimization methods and loading time improvements for single page applications (SPA) applications built with React. Key approaches are discussed, including lazy loading of components, code splitting, the use of service workers for caching, and asynchronous data fetching. Methods to optimize rendering, such as React.memo and PureComponent, are analyzed, as well as list virtualization for efficient handling of large data sets. The article offers practical recommendations for improving app responsiveness and reducing load times, thereby enhancing the user experience.

**Keywords:** optimization, performance, single page applications (SPA), React, lazy loading, code splitting.

#### **Аннотация**

Статья посвящена методам оптимизации производительности и времени загрузки для SPA-приложений, разработанных с использованием React. Рассматриваются ключевые подходы, такие как ленивое подключение компонентов, разделение кода, использование сервис-воркеров для кеширования и асинхронная загрузка данных. Анализируются методы оптимизации рендеринга, включая использование React.memo и PureComponent, а также виртуализация списков для эффективной работы с большими объемами данных. Статья предоставляет рекомендации для улучшения отклика приложения и снижения времени загрузки, что способствует лучшему пользовательскому опыту.

**Ключевые слова:** оптимизация, производительность, single page applications (SPA), React, ленивое подключение, разделение кода.

#### **Introduction**

One of the primary approaches to modern web application design, known as single page applications (SPA), is widely used for its ability to provide a more interactive and seamless user experience. One of the key advantages of SPA is that the application is loaded only once, and all user interactions occur without page reloads. However, despite its merits, the performance of SPA presents significant challenges, particularly regarding load time and response time. This is especially true for applications built using React, which, despite its high performance, requires additional optimization efforts to ensure the best user experience.

Performance optimization in the context of SPA on React covers a wide range of aspects, including resource loading management, component rendering, asynchronous data handling, and application state management. Since such applications are dynamically loaded and contain significant amounts of data, load and response times can vary considerably depending on the complexity of the architecture and the implemented approaches.

The goal of this research is to analyze existing methods of performance optimization for SPA on React, considering the specifics of their operation and potential issues related to asynchronicity and state management. The article discusses key approaches such as lazy loading of components, code splitting, and the use of service workers for caching, as well as tools for measuring performance.

### **Main part. Overview of SPA and design features**

Web applications built with SPA technology allow user interactions to occur on a single page without requiring a browser reload. Instead, when transitioning between different parts of the application, data is dynamically loaded through asynchronous requests, and the content is updated on the client side. This significantly improves the perceived speed of the application, making it more fluid and responsive. React, one of the most popular frameworks for SPA development, is used to create dynamic interfaces with the ability to update data without reloading the page [1].

A key feature of SPA is that all the HTML (HyperText Markup Language), CSS (Cascading Style Sheets), and JavaScript required for the application to function are loaded from the server only once. After this, interaction with the server is limited to loading necessary data in formats like JSON or other lightweight formats, which accelerates the application's performance compared to traditional multi-page websites. However, this architecture has its drawbacks, with the main issue being the first-load time. In SPA, all the necessary code must be loaded upfront, which can lead to significant delays during the initial user visit, especially if the application consists of many components and resources.

The rendering process in React-based SPA also has its peculiarities. React uses a virtual DOM, which helps efficiently update the user interface while minimizing the number of actual changes to the DOM tree. This optimizes rendering, but with a large number of components and complex state interactions, performance issues may arise. For example, re-rendering components that are not optimized with methods like `React.memo` or `PureComponent` can lead to unnecessary reprocessing and slow down the application [2].

Additionally, managing state is a crucial aspect of SPA. Unlike traditional server-side applications where state is stored on the server, SPA often manage state on the client side. This requires implementing effective solutions for state synchronization and storage, such as using state management libraries (e.g., `Redux` or `React Context`). However, improper state management can make the application overly complex and difficult to maintain, which will affect performance.

### **Principles and methods of optimization**

Performance optimization in React-based SPA applications includes several key areas aimed at improving load time, rendering speed, and responsiveness to user actions. One of the first challenges developers face when building SPA is the application's load time. Since all the necessary resources (HTML, CSS, JavaScript) are loaded on the first request, it is crucial to minimize the size of these files. Methods like lazy loading, code splitting, and the use of service workers are applied to address this issue.

Lazy loading involves loading components or modules only when they are needed for the application's operation. In React, this can be implemented using `React.lazy` and the `Suspense` component, which allow components to be loaded only when required [3]. This approach reduces the initial load time, as the user does not need to wait for the entire application to load.

Code splitting allows breaking the entire JavaScript into smaller files, which are loaded as needed. In React, this is often achieved using `Webpack`, which supports dynamic module loading. This reduces the amount of code initially loaded and speeds up the application's startup.

Service workers, on the other hand, enable caching of static resources such as JavaScript and CSS on the client side. This significantly reduces load times during subsequent visits, as most resources do not need to be reloaded from the server (fig.1).

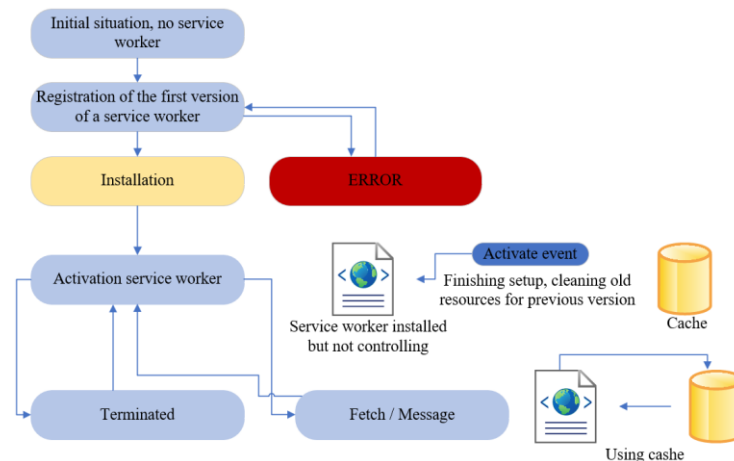


Figure 1. The architecture of the service worker [4].

The diagram illustrates the lifecycle and workflow of a service worker in the context of performance optimization in React-based SPA. It begins with the initial state where no service worker is registered, followed by the installation of the first version. If errors occur, the process enters an error state. Once successfully installed, the service worker proceeds to activation, where it completes setup tasks and cleans up resources from previous versions. In its active state, the service worker intercepts network requests (Fetch/Message) and utilizes cached resources to improve load times and reduce server requests, enhancing the performance and responsiveness of the React SPA. The diagram also highlights termination points where the service worker may become inactive, ensuring efficient resource management.

When application data updates, React performs component rendering to reflect the changes in the user interface. However, excessive rendering can significantly reduce performance. React.memo and PureComponent help optimize rendering for functional and class components, respectively. These methods prevent unnecessary re-renders when the component's props have not changed, which is particularly beneficial when working with large and complex interfaces [5].

In the case of large data lists, such as in tables or element lists, rendering all items at once can severely slow down the application. List virtualization displays only the elements visible on the screen, significantly reducing the load on the DOM and improving performance. Libraries like react-window or react-virtualized are often used for this purpose.

Inefficient state management can lead to unnecessary re-renders, especially in complex applications with large amounts of data. Optimizing state updates involves using the useMemo hook for memoizing calculations and the useCallback hook for memoizing functions passed to child components. This helps avoid unnecessary resource consumption during component rendering.

To ensure fast responsiveness to user actions, it is important to optimize asynchronous operations, such as server requests or handling large volumes of data [6]. Instead of loading all the data upfront, the application can load only the data needed at the moment. This can be achieved using asynchronous requests with async/await or the Axios library for sending HTTP requests. This approach reduces load times and minimizes server load.

In some cases, approaches like debouncing or task timing can be used to limit the number of concurrent asynchronous requests. For example, when the user enters data into a text field, requests to the server can be delayed until the user stops typing, preventing unnecessary requests and reducing server load.

React SPA can consume significant memory, especially when working with large datasets or long-living components (table 1).

Table 1

*Results for js web frameworks benchmark [7].*

<i>Name</i>	<i>Vanillajs (mb/s)</i>	<i>Vue (mb/s)</i>	<i>React (mb/s)</i>
<i>Ready memory. Memory usage after page load.</i>	<i>0,6</i>	<i>0,8</i>	<i>1,1</i>
<i>Run memory. Memory usage after adding 1,000 rows.</i>	<i>1,8</i>	<i>3,7</i>	<i>4,9</i>
<i>Update every 10<sup>th</sup> row for 1k rows (5cycles). Memory usage after clicking update every10th row 5 times</i>	<i>2</i>	<i>3,7</i>	<i>5,5</i>
<i>Creating/clearing 1k rows (5 cycles). Memory usage after creating and clearing 1000 rows 5 times</i>	<i>0,7</i>	<i>1,2</i>	<i>1,9</i>
<i>Run memory 10k. Memory usage after adding 10,000 rows.</i>	<i>11,9</i>	<i>27,7</i>	<i>36,1</i>
<i>Geometric mean of. All factors in the table</i>	<i>1,02</i>	<i>1,89</i>	<i>2,61</i>

Using libraries like Redux or React Context allows centralization of state management and efficient usage across different parts of the application. It is also important to release resources when components are removed to avoid memory leaks.

Comparative analysis of optimization methods

When developing SPA in React, there are numerous optimization techniques, each with its pros and cons depending on the application's context and needs. The first thing a user notices when launching any web application is its loading speed.

Lazy Loading and Code Splitting are two methods aimed at reducing load times, but their applicability depends on the specifics of the app. Lazy loading ensures that only the necessary components are loaded at the moment, significantly speeding up the initial load. However, if components are not properly split or are loaded too late, this could cause delays in content display. Code splitting, implemented with tools like Webpack, allows the app to be divided into modules that load as needed.

When combined, these approaches can significantly boost performance. For example, splitting the code into smaller chunks allows key modules to load at startup, while the rest load dynamically as necessary, improving user experience.

React.memo and PureComponent prevent redundant renders when props or state change. These methods are particularly useful with components that update frequently but don't change. However, overusing them may lead to performance issues, as React needs to perform extra prop comparisons. These methods are effective for components dependent on simple data but may not be optimal for more complex or dynamic interfaces.

List Virtualization, on the other hand, significantly improves performance when dealing with large data sets. Instead of rendering all list items, virtualization renders only those visible to the user, reducing the load on the DOM and speeding up rendering [8]. This is particularly beneficial for long lists or tables, where each item may contain complex components. However, virtualization requires additional configuration and may be unnecessary for small or static lists.

Asynchronous Data Loading reduces the time spent loading initial resources. Using techniques like debouncing or asynchronous requests with async/await helps load data only when needed, improving the user experience. However, frequent requests or complex asynchronous operations can harm performance if not managed with proper tools, like caching or cancelling unnecessary requests.

State Management Optimization also plays a crucial role in performance. In large applications, global state management libraries like Redux can be redundant, leading to additional overhead in managing state and synchronizing data. Using React Context and local states, when possible, helps reduce re-renders and improve performance.

Service workers and caching can significantly reduce load times during subsequent visits, as static resources are cached in the browser. This provides a substantial advantage for high-traffic apps, where frequent server calls can be costly. However, not all resources can or should be cached, and it is essential to configure a proper caching strategy to avoid using outdated data.

Profiling tools, such as React DevTools and Lighthouse, help identify performance bottlenecks by measuring metrics like render time, memory usage, and response time. However,

unlike caching methods, profiling is a reactive technique that doesn't reduce load time but instead helps identify issues for future optimization.

### Conclusion

Optimizing SPA performance in React is crucial for improving user experience and app efficiency. Modern techniques, such as lazy loading, code splitting, service workers, and asynchronous data loading, help significantly reduce load times and increase app responsiveness. Each method has its characteristics and areas of application, requiring a careful approach based on project specifics. For instance, React.memo and PureComponent reduce unnecessary renders, while list virtualization efficiently handles large data sets, minimizing DOM load.

However, optimization requires careful tuning and combining various techniques to achieve the best results. Improving performance isn't just about the initial load but also involves all aspects of app operation, including state management and asynchronous tasks. With the development of profiling and performance analysis tools like React DevTools and Lighthouse, optimization will become more precise and effective.

Thus, the strategic application of optimization techniques not only accelerates app performance but also creates a better user experience, which is a key factor in the success of modern React-based web applications.

\*\*\*

1. Ricky J. Development of Front-End Web Applications Utilizing Single Page Application Framework and React.js Library // International Journal Software Engineering and Computer Science. 2023. Vol. 3 No. 3. P. 529 – 536.
2. Fabio F. Detecting code smells in React-based Web apps // Information and Software Technology. 2023. Vol. 155.
3. Răzvan-Mihail B. Analysing the performance impacts of lazy loading in web applications // Journal of Information Systems & Operations Management, 2024. Vol. 18.1. P. 1-15.
4. Using Service Workers / MDN WEB DOCS // URL: [https://developer.mozilla.org/en-US/docs/Web/API/Service\\_Worker\\_API/Using\\_Service\\_Workers](https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers) (date of application: 05.10.2024).
5. Shailesh S. Mobile Web Performance Optimization / Modern Web Performance Optimization. 2020. P. 79-103.
6. Sidorov D. Leveraging web components for scalable and maintainable development // Sciences of Europe. 2024. № 150. P. 87-89.
7. Results for js web frameworks benchmark. / Krausest GitHub // URL: [https://krausest.github.io/js-framework-benchmark/2023/table\\_chrome\\_113.0.5672.63.html](https://krausest.github.io/js-framework-benchmark/2023/table_chrome_113.0.5672.63.html) (date of application: 29.10.2024).
8. Zeyu C. A multi-threaded virtual DOM difference method based on web worker. // Sixth International Conference on Computer Information Science and Application Technology. 2023. Vol. 12800.











Рецензируемый научный журнал

**ТЕНДЕНЦИИ РАЗВИТИЯ НАУКИ И ОБРАЗОВАНИЯ  
№116, Декабрь 2024**

Часть 19