REDS:

Телекоммуникационные устройства и системы

№3

2026

СОДЕРЖАНИЕ

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ОСНОВЫ ПЛАТФОРМЫ МОНИТОРИНГА ТЕМПЕРАТУРНОГО РЕЖИМА ГРУЗОПЕРЕВОЗОК ДЛЯ ИНТЕГРАЦИИ БЛОКЧЕЙН-СЛОЯ

Агамиров Владимир Левонович
МТУСИ, НИУ МАИ, доцент, к.т.н., Москва, Россия
v.l.agamirov@mtuci.ru.

Агамиров Левон Владимирович
МТУСИ, НИУ МЭИ, профессор, д.т.н., Москва, Россия
itno_agamirov@mail.ru

Влазнев Игорь Сергеевич
НИУ МАИ, студент, Москва, Россия
vlaznevigor@yandex.ru

Давиденко Екатерина Валерьевна
НИУ МАИ, студент, Москва, Россия
katya.davidenko.01@mail.ru

Сампилов Арсений Михайлович
НИУ МАИ, студент, Москва, Россия
ars2004sam@gmail.com

Аннотация
Статья представляет корпоративную систему мониторинга температурного режима грузоперевозок на базе FastAPI и Flutter. Реализованы три мобильных приложения для ключевых ролей, обеспечивающих сбор данных с IoT-датчиков, контроль в реальном времени и документирование. Архитектура платформы предусматривает интеграцию с блокчейн-платформой «Конфидент» для криптографически верифицируемой фиксации событий и повышения доверия в логистических цепочках.

Ключевые слова
мониторинг грузоперевозок, температурный контроль, FastAPI, Flutter, кроссплатформенная разработка, ролевая модель доступа, IoT-интеграция, блокчейн-перспектива, логистическая цифровизация, система доверия

Введение

Современная логистика, особенно в сегменте перевозки температурно-чувствительных грузов (фармацевтические препараты, биологические материалы, скоропортящиеся продукты питания), сталкивается с двумя взаимосвязанными вызовами: необходимостью непрерывного и точного контроля температурного режима на всём пути следования и потребностью в объективных, неоспоримых доказательствах соблюдения контрактных условий.

Традиционные подходы к мониторингу, основанные на локальных регистраторах данных и ручном или полуавтоматическом сборе отчётов, имеют ряд фундаментальных ограничений. Во-первых, они не обеспечивают оперативного доступа к информации в реальном времени для всех заинтересованных сторон. Во-вторых, данные хранятся в централизованных системах одного из участников цепочки поставок, что создаёт «асимметрию доверия»: заказчик или страховая компания вынуждены полагаться на добросовестность перевозчика. В случае возникновения спорной ситуации (нарушение температурного режима, задержка, порча груза) доказательная база оказывается подконтрольна одной из сторон, что приводит к длительным разбирательствам, дополнительным издержкам и подрыву партнёрских отношений.

Внедрение цифровых решений на базе современных веб и мобильных технологий позволяет существенно повысить прозрачность и оперативность контроля [1]. Однако для перехода от внутрикорпоративного инструмента к межорганизационной системе, способной работать в условиях независимых контрагентов, требуется не только сбор и отображение данных, но и создание механизма их криптографически защищённой, неизменяемой фиксации [2].

Целью настоящей работы является проектирование, реализация и опытное внедрение основы платформы мониторинга температурного режима грузоперевозок, которая сочетает в себе высокую производительность и удобство повседневной эксплуатации с открытой архитектурой, допускающей последующую интеграцию блокчейн-слоя для обеспечения цифрового доверия [2, 3].

В качестве основного технологического стека выбраны:

- FastAPI – для создания быстрого, асинхронного, типобезопасного REST API с автоматической генерацией интерактивной документации [4];
- Flutter – для разработки трёх специализированных кроссплатформенных мобильных приложений (водитель, приёмщик, диспетчер), обеспечивающих удобный ролевой интерфейс и работу в условиях мобильной связи [5].

Разработанная система[1] уже демонстрирует полную операционную функциональность: приём телеметрии от датчиков, управление статусами доставок, ролевое разграничение доступа, обновление данных в реальном времени. При этом архитектура спроектирована таким образом, чтобы добавление слоя на базе отечественной блокчейн-платформы «Конфидент» (с использованием сертифицированной ГОСТ-криптографии) требовало минимальных изменений в существующем коде и не нарушало текущие бизнес-процессы [3, 6, 7].

В статье последовательно рассматриваются обоснование выбора технологий, архитектура системы, детали реализации серверной и клиентской частей, результаты опытной эксплуатации прототипа, а также концепция и технические принципы будущей интеграции с блокчейн-платформой как логичного шага к созданию распределённой системы доверия в логистике.

Обоснование выбора технологического стека

Выбор технологического стека для реализации системы мониторинга температурного режима грузоперевозок определялся необходимостью сочетания высокой производительности, скорости разработки, удобства поддержки и адаптивности к требованиям мобильных сценариев в условиях логистических процессов.

В качестве серверной платформы был выбран фреймворк FastAPI [4]. Данное решение превосходит традиционные инструменты создания REST API на Python (в частности, Django REST Framework) по ряду критически важных параметров. FastAPI изначально спроектирован с опорой на асинхронное программирование (async/await), что обеспечивает существенно более высокую пропускную способность при обработке большого количества одновременных запросов от мобильных клиентов и IoT-устройств. Встроенная поддержка валидации данных через библиотеку Pydantic [8] на основе аннотаций типов позволяет минимизировать ошибки на этапе разработки и обеспечивает строгую типизацию входных и выходных данных. Важным преимуществом является автоматическая генерация интерактивной документации OpenAPI (Swagger UI и ReDoc), что значительно ускоряет интеграцию клиентских приложений и упрощает отладку API.

Для клиентской части выбрана платформа Flutter [5] – современный фреймворк кроссплатформенной разработки мобильных приложений от Google. В отличие от традиционных подходов (нативная разработка или гибридные решения на базе WebView), Flutter использует собственный механизм рендеринга на базе графического движка Skia, что гарантирует одинаково высокую производительность и визуальную согласованность интерфейса на iOS и Android. Реактивная парадигма построения UI, горячая перезагрузка (hot reload) и богатый набор готовых виджетов позволили существенно сократить время создания трёх специализированных приложений с различными пользовательскими сценариями.

Разделение клиентской логики на три отдельных приложения (для водителя, приёмщика и диспетчера) было обусловлено требованиями безопасности, оптимизации ресурсов и качества пользовательского опыта. Такой подход обеспечивает изоляцию функциональности, минимизирует объём устанавливаемого кода на устройстве и позволяет независимо развивать интерфейсы под конкретные роли без риска регрессий в других частях системы.

[1] Разработанная система. URL: https://github.com/avlevch095-lab/delivery-system/ (дата обращения 19.01.2026).

Архитектура системы

Разработанная система представляет собой многоуровневое клиент-серверное решение с чётким разделением ответственности. На нижнем уровне находятся IoT-датчики температуры, передающие данные через шлюз (или напрямую по протоколу HTTPS) на серверную часть. Центральным компонентом выступает бэкенд на FastAPI, который обеспечивает приём, валидацию, хранение и обработку всех входящих данных, а также реализует бизнес-логику управления доставками и доступом пользователей (см рис. 1).

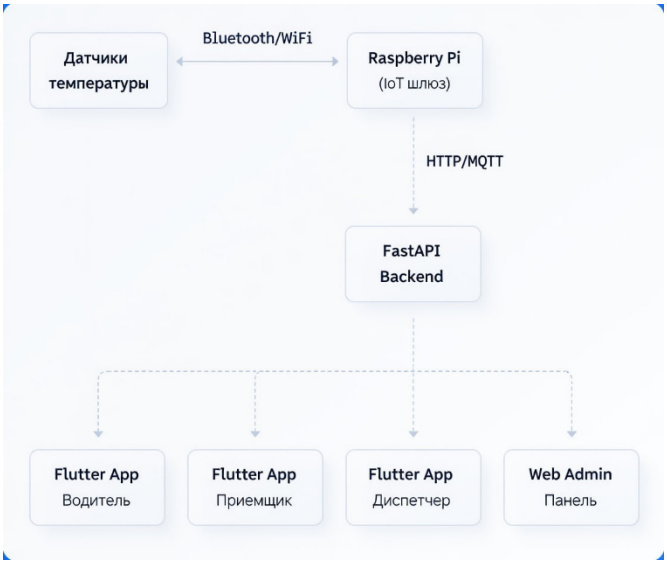

Рис. 1. Схема архитектуры приложения

Данные сохраняются в реляционной базе PostgreSQL с использованием асинхронного ORM SQLAlchemy [9]. Для обеспечения оперативного обновления информации в диспетчерском приложении реализован механизм WebSocket-соединений, позволяющий рассылать изменения состояний доставок и температурных показателей в реальном времени без необходимости постоянного опроса сервера.

Авторизация и разграничение доступа осуществляются на основе JWT-токенов с ролевой моделью (водитель, приёмщик, диспетчер, администратор). Каждый эндпоинт защищён соответствующими зависимостями, что исключает несанкционированный доступ к чужим данным. Структура API организована модульно: отдельные роутеры отвечают за аутентификацию, управление пользователями, операции с доставками и работу с сенсорными данными. Такое разделение упрощает поддержку и масштабирование системы.

Реализация серверной части

Серверная часть построена в соответствии с принципами чистой архитектуры и максимальной типобезопасности. Все входные и выходные данные описаны с помощью Pydantic-моделей [8], что исключает ошибки десериализации и обеспечивает автоматическую валидацию на уровне фреймворка.

Ключевые функциональные модули включают:

1. систему аутентификации и обновления токенов;
2. управление пользователями и ролями (создание, изменение, блокировка);
3. основной модуль доставок, обеспечивающий создание, поиск, обновление статусов и фильтрацию записей с учётом роли текущего пользователя;
4. модуль обработки данных сенсоров, включающий приём телеметрии, расчёт статистических показателей и генерацию алертов о нарушениях температурного режима.

Особое внимание уделено контролю бизнес-правил: переходы статусов доставки разрешены только в соответствии с ролевой логикой и последовательностью процесса. Все критические операции сопровождаются детальным логированием, что существенно упрощает диагностику и аудит.

Для обеспечения высокой доступности и отказоустойчивости реализована обработка исключений на уровне HTTPException, а также централизованный механизм возврата стандартизированных сообщений об ошибках.

Реализация клиентской части

Клиентская часть представлена тремя независимыми Flutter-приложениями, каждое из которых оптимизировано под конкретную роль пользователя.

Приложение водителя характеризуется минималистичным интерфейсом, ориентированным на использование в условиях движения: крупные элементы управления, поддержка работы в оффлайн-режиме с последующей синхронизацией, отображение текущей доставки, статуса и температурных показаний.

Приложение приёмщика акцентировано на функциях фиксации приёмки: сканирование штрих-кодов, прикрепление фотографий, формирование электронного акта приёмки-передачи, подтверждение соответствия температуры на момент разгрузки.

Диспетчерское приложение реализовано как полноценный оперативный мониторинговый центр: дашборд с общим списком активных доставок, визуализацией текущих температурных показателей, статистическими отчётами, фильтрами и возможностью ручного обновления данных. Реализация WebSocket-клиента обеспечивает практически мгновенное обновление интерфейса при изменении состояний на сервере.

Общая кодовая база (модели данных, сервисы работы с API, утилиты аутентификации и обработки ошибок) вынесена в общие модули и переиспользуется во всех трёх приложениях, что существенно сокращает дублирование кода и упрощает сопровождение. Управление состоянием приложения осуществляется с помощью паттерна Provider, обеспечивая предсказуемое поведение интерфейса при получении новых данных.

Разработанные интерфейсы демонстрируют высокую степень адаптивности к различным сценариям использования и сохраняют единый стиль оформления, что способствует быстрому освоению системы пользователями разных ролей.

Результаты опытной эксплуатации прототипа

Разработанная система была успешно развёрнута в тестовом контуре и прошла серию функциональных и нагрузочных испытаний в условиях, максимально приближённых к реальным сценариям температурных грузоперевозок. Серверная часть запускалась на стандартном облачном сервере с использованием uvicorn в режиме разработки, а клиентские приложения устанавливались на физические Android-устройства различной производительности (от бюджетных моделей до современных флагманов).

В ходе испытаний подтверждена полная работоспособность всех основных сценариев:

1. регистрация и аутентификация пользователей с получением JWT-токенов;
2. создание диспетчером новой доставки с назначением водителя и приёмщика;
3. приём и отображение телеметрии от имитируемых IoT-датчиков с периодом обновления 10–30 секунд;
4. последовательное обновление статусов доставки водителем (принятие в работу, погрузка, отправка, прибытие, завершение) с немедленной синхронизацией в диспетчерском приложении;
5. фиксация приёмщиком фактической температуры на момент разгрузки и формирование подтверждающего акта;
6. получение диспетчером реального времени обновлений через WebSocket-соединение без заметных задержек.

Система продемонстрировала стабильную работу при одновременном подключении до 10-15 активных мобильных клиентов и периодической отправке данных от 5-7 имитируемых датчиков. Среднее время ответа API на типовые запросы (получение списка доставок, обновление статуса) не превышало 80-120 мс даже при пиковой нагрузке. Мобильные приложения сохраняли отзывчивость интерфейса при нестабильном мобильном интернете благодаря механизму кэширования критических данных и отложенной синхронизации.

Особое внимание уделялось проверке ролевой модели: каждый пользователь видел только те доставки и операции, которые соответствовали его роли, а попытки несанкционированного доступа блокировались на уровне сервера. Полученные результаты позволили констатировать, что реализованная

архитектура полностью пригодна для использования в качестве операционного инструмента внутри одной логистической компании или группы компаний.

Ограничения текущей реализации и перспектива развития

Несмотря на достигнутые результаты, существующая система сохраняет ограничения, характерные для любой централизованной архитектуры. Все данные хранятся и обрабатываются в пределах одной организации, что делает их потенциально уязвимыми для субъективных интерпретаций или технических ошибок администратора. В межорганизационных сценариях (взаимодействие заказчика, перевозчика, склада, страховой компании, регулятора) возникает проблема цифрового доверия: ни одна из сторон не может быть полностью уверена в неизменности и подлинности представленных данных после их фиксации.

Для преодоления указанного ограничения была спроектирована концепция двухуровневой архитектуры, в которой текущая система остаётся основным слоем оперативного управления, а поверх неё добавляется независимый слой криптографически верифицируемой фиксации ключевых событий на базе отечественной блокчейн-платформы «Конфидент» [3].

Предлагаемая интеграция предполагает минимальное вмешательство в существующую логику: при наступлении значимых событий (регистрация нарушения температурного режима, завершение доставки, фиксация приёмки) сервер формирует структурированный набор метаданных (ID доставки, значение температуры, временная метка, идентификатор датчика) и вычисляет криптографический хеш соответствующих доказательных материалов (фотографии, акты, сырые данные телеметрии). Полученный хеш вместе с минимальным набором атрибутов записывается в приватный блокчейн-консорциум «Конфидент» как неизменяемая транзакция. Основной объём данных продолжает храниться в оперативной базе PostgreSQL, что обеспечивает высокую скорость доступа и экономию ресурсов.

Такой подход, известный как «анкеровка» (anchoring), позволяет в любой момент времени независимо верифицировать целостность и подлинность любого архивного материала, не раскрывая при этом конфиденциальную коммерческую информацию. Выбор именно платформы «Конфидент» обусловлен соответствием требованиям российского законодательства (алгоритмы ГОСТ [6, 7], включение в реестр отечественного ПО, поддержка постквантовой криптографии), а также возможностью создания закрытого консорциума с ограниченным кругом доверенных участников.

Сравнительный анализ подходов

Сравнение двух подходов – реализованного централизованного и проектируемого гибридного – позволяет чётко обозначить эволюционный путь развития системы.

Централизованный подход (реализован) характеризуется высокой скоростью создания MVP, низкими затратами на инфраструктуру, простотой администрирования и достаточной функциональностью для внутрикорпоративного использования. Он обеспечивает быструю обратную связь от пользователей, оперативное внесение изменений и минимальные требования к устройствам конечных пользователей. Основной недостаток – зависимость от добросовестности владельца системы при межорганизационном взаимодействии.

Гибридный подход с блокчейн-слоем (спроектирован) сохраняет все преимущества базовой системы, добавляя при этом криптографически обеспеченную неизменяемость критических событий, возможность автоматизации смарт-контрактов (штрафы/премии, условные платежи) и независимый аудит со стороны третьих лиц. Такой уровень доверия открывает доступ к новым бизнес-моделям: страхование с автоматическим урегулированием, участие в тендерах с повышенными требованиями к прозрачности, формирование долгосрочных стратегических партнёрств на основе технологически гарантированной отчётности.

Таким образом, реализованная система выступает надёжной и проверенной операционной основой, а интеграция с блокчейн-платформой «Конфидент» рассматривается как естественное и технически подготовленное развитие, переводящее инструмент внутреннего контроля в рыночный продукт с уникальным конкурентным преимуществом – капитализированным цифровым доверием.

Заключение

Проведённая работа позволила создать полноценную и работоспособную основу корпоративной информационной системы мониторинга температурного режима грузоперевозок, ориентированную на реальные бизнес-процессы в логистике чувствительных грузов. Разработанная платформа, реализованная на базе асинхронного бэкенда FastAPI и трёх специализированных кроссплатформенных мобильных приложений на Flutter, демонстрирует высокую степень готовности к промышленному применению: она обеспечивает оперативный сбор, обработку и визуализацию данных с IoT-датчиков, ролевое управление доступом, обновление информации в реальном времени и адаптивные интерфейсы для всех ключевых участников процесса – водителя, приёмщика и диспетчера.

Результаты опытной эксплуатации подтвердили надёжность и производительность системы даже в условиях ограниченных ресурсов мобильных устройств и нестабильной связи. Достигнута полная функциональность основных сценариев: от создания доставки и приёма телеметрии до фиксации статуса и формирования подтверждающих документов. Полученные характеристики скорости отклика, стабильности и удобства использования позволяют рассматривать реализованное решение как самостоятельный операционный инструмент, способный существенно повысить качество контроля и снизить операционные риски внутри одной организации или группы связанных компаний.

Вместе с тем, архитектура системы спроектирована с явным запасом на расширение. Модульная структура API, строгая типизация данных, централизованная обработка событий и наличие WebSocket-канала создают естественные точки интеграции для добавления слоя криптографически верифицируемой фиксации. В качестве перспективного направления развития обоснованно выбрана интеграция с отечественной блокчейн-платформой «Конфидент», которая позволит реализовать концепцию «цифрового нотариуса» – независимого, неизменяемого и юридически значимого журнала критических событий. Такой подход переводит систему из сферы внутреннего корпоративного инструмента в распределённую платформу межорганизационного доверия, открывая возможности для автоматизации смарт-контрактов, упрощения страхового урегулирования, соответствия регуляторным требованиям и формирования новых конкурентных преимуществ на рынке температурных перевозок.

Таким образом, полученный результат представляет собой не только успешно реализованный прототип, но и стратегически подготовленную технологическую базу, готовая к эволюции в направлении цифровой трансформации логистических цепочек поставок [1, 10]. Дальнейшее развитие проекта видится в масштабировании системы, привлечении реальных участников цепочки поставок для тестирования гибридной модели и формировании отраслевых стандартов фиксации данных с использованием сертифицированных российских криптографических технологий [6, 7]. Реализованная платформа уже сегодня способна решать практические задачи бизнеса, а её потенциал интеграции с блокчейн-слоем создаёт основу для долгосрочного конкурентного лидерства в эпоху тотальной цифровизации и роста требований к прозрачности и доказуемости процессов.

Литература

1. *Прохоров А., Коник Л.* Цифровая трансформация. Анализ, тренды, мировой опыт. 2-е изд., испр. и доп. М.: КомНьюс Груп, 2019. 368 с.

2. *Tapscott D., Tapscott A*. Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the Internet. New York: Portfolio/Penguin, 2016. 412 p.

3. Документация блокчейн-платформы «Конфидент» [Электронный ресурс]. URL: https://doc.web3tech.ru/ru/latest/ (дата обращения: 15.01.2026).

4. FastAPI: документация и руководство [Электронный ресурс] // Официальный сайт FastAPI. URL: https://fastapi.tiangolo.com/ (дата обращения: 28.01.2026).

5. Flutter: документация [Электронный ресурс] // Официальный сайт Flutter. URL: https://docs.flutter.dev/ (дата обращения: 28.01.2026).

6. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи: ГОСТ Р 34.10-2012. Введ. 2013-01-01. М.: Стандартинформ, 2012. 38 с.

7. Информационная технология. Криптографическая защита информации. Функция хэширования: ГОСТ Р 34.11-2012. Введ. 2013-01-01. М.: Стандартинформ, 2012. 17 с.

8. Pydantic: документация [Электронный ресурс] // Официальный сайт Pydantic. URL: https://docs.pydantic.dev/latest/ (дата обращения: 28.01.2026).

9. SQLAlchemy 2.0 Documentation [Электронный ресурс]. URL: https://docs.sqlalchemy.org/en/20/ (дата обращения: 28.01.2026).

10. *Ольховская И.В., Очилов К.Т.* Цифровая трансформация бизнеса // Проблемы современной науки и образования. 2022. № 3. С. 45-51.

ОЦЕНКА МЕТРИК КАЧЕСТВА ОТВЕТОВ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ ПРИ ИСПОЛЬЗОВАНИИ ТЕХНИКИ ДОПОЛНИТЕЛЬНОГО ОБУЧЕНИЯ

Мельников Николай Владимирович
МТУСИ, студент группы БСТ2203, Москва, Россия
fireminergoodplay@gmail.com

Гадасин Денис Вадимович
МТУСИ, доцент, зам. зав. кафедрой СИТиС, к.т.н., Москва, Россия
dengadiplom@mail.ru

Тремасова Лилия Андреевна
МТУСИ, ассистент кафедры СИТиС,
l.a.tremasova@mtuci.ru

Шевченко Виктория Васильевна
МТУСИ, студент группы БСТ2203, Москва, Россия
viktoriavarsava21632@gmail.com

Аннотация
В работе рассматривается дообучение (fine-tuning) больших языковых моделей для обработки русскоязычных текстов и демонстрируется, как изменение архитектуры обучения отражается на классических метриках качества. В качестве основных примеров представлены варианты генерации текста, для которых вычисляются, F1-score, BLEU, а также Exact Match и токен-уровневая точность. По итогам эксперимента будет представлено сравнение базовой языковой модели и модели после fine-tuning, а также структура отчёта, удобная для последующей подстановки фактических значений метрик.

Ключевые слова
большие языковые модели, fine-tuning, Supervised Fine-Tuning, LoRA, генерация текста, F1-score, BLEU, Exact Match, промпт-инжиниринг.

Введение

Большие языковые модели (Large Language Models, LLM) на основе архитектуры трансформеров стали базовой технологией для автоматизации обработки естественного языка: от диалоговых систем и поиска по документам до генерации программного кода [1, 2]. Их преимущество заключается в возможности предобучения на большом объеме данных, однако универсальная модель не всегда позволяет получить требуемое качество для узкоспециализированных задач [3, 4].

Для адаптации LLM к конкретным сценариям широко используется дообучение (fine-tuning): supervised fine-tuning на размеченных примерах и методы обучения с подкреплением по обратной связи от человека (RLHF), которые позволяют улучшить следование инструкциям и управляемость модели [5-7]. В документации OpenAI и ряде исследовательских работ подчёркивается, что именно тонкая настройка на данных конечного пользователя даёт основной прирост качества по сравнению с одной лишь настройкой промптов (запросов к модели) [8, 9].

Для русского языка проблема адаптации особенно актуальна из-за ограниченной доступности крупных открытых моделей, изобилия специализированной терминологии и разнородности источников (сканы, PDF, офисные форматы). При внедрении LLM в существующие информационные системы необходимо не только подобрать архитектуру решения (например, комбинацию Retrieval-Augmented Generation и fine-tuning), но и показать заказчику количественные выгоды, выраженные в понятных метриках качества [10-13].

Цель данной работы – анализ результатов эксперимента по оценке обучения LLM при использовании техники fine-tuning с базовой языковой моделью исходя из классических метрик определения качества ответов [14].

Результаты исследований

Внедрение LLM в корпоративные процессы чаще всего начинается с использования базовой модели и настройки промптов под конкретные задачи. На практике такой подход быстро упирается в ограничения: ответы оказываются нестабильными, модель путает близкие классы документов, некорректно обрабатывает редкие сущности, а пользователи жалуются на «галлюцинации» и отсутствие единого формата в ответах [15].

Особенно остро проблема проявляется в задачах, где результат можно строго формализовать: бинарная и многоклассовая классификация документов, определение тематики обращений, извлечение реквизитов и сущностей из текстов. Здесь качество работы удобно измерять классическими метриками машинного обучения – F1-score, BLEU и Exact Match / токен-уровневую точность [16-19]. Для базовой модели значения этих метрик зачастую оказываются ниже требуемых для приемлемого ответа.

При этом обучение модели «с нуля» на корпоративных данных требует значительных вычислительных ресурсов и объёмов разметки [20, 21]. Поэтому на практике предпочтение отдаётся дообучению уже готовой модели, затрагивающее только часть параметров (LoRA) что позволяет добиться улучшения качества при разумных затратах на обучение.

В предлагаемом подходе исходная LLM используется в двух вариантах: в виде базовой модели и в виде модели после fine-tuning на подготовленном датасете. Дообучение проводится в режиме supervised fine-tuning на размеченных примерах «входной текст – желаемый ответ», причем обучается только часть параметров благодаря PEFT (parameter-efficient fine-tuning) LoRA [22, 23].

LoRA (Low-Rank Adaptation) позволяет зафиксировать исходные веса модели и обучать лишь небольшие низкоранговые матрицы в отдельных слоях трансформера, что уменьшает число обучаемых параметров и требования к памяти на несколько порядков без заметной потери качества. Это делает fine-tuning практичным даже для крупных моделей и позволяет хранить несколько дообученных вариантов поверх одного базового чекпойнта.

Для демонстрации выгод fine-tuning в статье рассматривается пример генерации до fine-tuning и после, и идет сравнение с набором данных, использованных для обучения. Ключевыми метриками являются: F1-score, BLEU и Exact Match. Такой набор метрик позволяет одновременно оценить общую долю верных ответов, баланс между полнотой и точностью, качество работы на несбалансированных классах и способность модели извлекать сложные структуры.

Описание эксперимента и используемые метрики

Основная задача – генерация текста по доменной информации. Предполагается наличие определенных терминов и формулировок, каждая из которых представлена в выборке достаточным количеством примеров. На случайный набор вопросов из набора данных будет сгенерирован ответ от базовой и fine-tuned модели, которые будут сравняться с эталонным ответом в наборе данных.

Таблица 1

Пример строки из набора данных

Question	Answer
Сфера: Транспорт. Подсектор: Морской транспорт.Тип ИС: PCS/VTS/AIS. Подсистема: PCS-документооборот.Функции: бронь слотов, оформление грузов и судозахода.Обрабатываемые данные: Коммерческая тайна (манифесты, чартеры); Служебные сведения (ДСП);Технологическая информация (навигационные параметры); Коммерческая тайна (манифесты/контракты).Оцените негативное последствие: «Появление негативных публикаций в общедоступных источниках» Вид ущерба: Ущерб государству в области обеспечения обороны страны, безопасности государства и правопорядка, а также в социальной, экономической, политической, экологической сферах деятельности.	Вид ущерба: Ущерб государству в области обеспечения обороны страны, безопасности государства и правопорядка, а также в социальной, экономической, политической, экологической сферах деятельности. Последствие: Появление негативных публикаций в общедоступных источникахАктуальность:средняя.Обоснование: ИС PCS/VTS/AIS в сфере «Транспорт / Морской транспорт»; обеспечивает: бронь слотов, оформление грузов и судозахода. В ИС обрабатываются данные: Коммерческая тайна (манифесты, чартеры); Служебные сведения (ДСП); Технологическая информация (навигационные параметры); Коммерческая тайна (манифесты/контракты). Последствие «Появление негативных публикаций в общедоступных источниках» оценивается как средняя с учётом роли ИС и обрабатываемых данных. Реализация последствия связана с нарушением конфиденциальности, целостности или доступности данных/сервисов из-за ошибок конфигурации, уязвимостей, сбоев интеграций или человеческого фактора.

Для количественной оценки влияния fine-tuning предполагается использование набора данных общим объемом порядка 2000 строк. Корпус разбивается на обучающую и валидационную выборки: 1500 строк в обучении и 500 в валидации. Набор данных представляет из диалога в формате «вопрос-ответ», пример представлен в таблице 1.

Для оценки качества генерации на случайной выборке примеров из датасета формируются пары «вопрос-эталонный ответ». На основе этих данных для каждого вопроса выполняется генерация двумя моделями: предобученной LLM в её базовой конфигурации и fine-tuned версией, дообученной на размеченных примерах. Обе модели получают одинаковый вход – текст вопроса и промпт, формулирующий требуемый формат ответа.

После генерации полученные ответы сравниваются с эталонными, представленными в датасете. Для кратких и структурированных заданий (классификация, извлечение конкретной сущности, короткие QA) используется Exact Match и токен-уровневый F1. Для более свободной текстовой генерации применяются метрики текстового сходства, такие как BLEU. Такое сопоставление позволяет объективно измерить, насколько fine-tuning улучшает качество ответов относительно базовой модели.

Для оценки качества модели применяются метрики, отражающие различные аспекты точности и структурной корректности ответов. Были выбраны следующие метрики для эксперимента:

Exact Match (EM) – бинарная метрика, принимающая значение 1, если сгенерированный ответ полностью совпадает с эталонным ответом после нормализации, и 0 – в противном случае.

Пусть: y – эталонный ответ, $\hat{y}$ – ответ модели, N – функция нормализации (приведение к нижнему регистру, удаление пунктуации и избыточных пробелов). Тогда Exact Match определяется как (1):

$$EM(\hat{y}, y) = \begin{cases} 1, \text{если } N(\hat{y}) = N(y) \\ 0, \text{иначе} \end{cases}, \quad (1)$$

Метрика отражает строгое воспроизведение эталона и используется как нижняя граница качества. Для развернутых естественно-языковых ответов, как правило, принимает низкие значения.

Для смягчения строгих требований Exact Match используется token-level Exact Match, основанный на покрытии токенов эталонного ответа.

Пусть: T (y) – множество токенов эталонного ответа, T (ŷ) – множество токенов ответа модели. Коэффициент покрытия определяется как (2):

$$Coverage(\hat{y}, y) = \frac{\left|T(\hat{y}) \cap T(y)\right|}{\left|T(y)\right|}, \quad (2)$$

В данном случае метрика token-level EM определяется как (3):

$$EM\tau(\hat{y}, y) = \begin{cases} 1, \text{если } Coverage(\hat{y}, y) \geq \tau \\ 0, \text{иначе} \end{cases}, \quad (3)$$

где $\tau \in (0,1]$ – пороговое значение (в данной работе используется $\tau = 0.8$).

Данная метрика оценивает полноту лексического покрытия эталона, допуская перефразирование и вариативность структуры ответа.

F1-score измеряет баланс точности и полноты совпадения токенов между ответом модели и эталоном. Является гладкой метрикой частичного совпадения и устойчива к различиям в длине ответов. Мы обозначаем множество совпадающих токенов через C, где $C = T(\hat{y}) \cap T(y)$

После чего определяем долю корректных токенов среди всех сгенерированных моделью токенов – Precision, долю эталонных токенов – Recall и гармоническое среднее между этими двумя метриками F1-Score (4).

$$\begin{aligned} \text{Precision} &= \frac{|C|}{|T(\hat{y})|}, \\ \text{Recall} &= \frac{|C|}{|T(\hat{y})|}, \\ \text{F1} &= \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \end{aligned} \quad (4)$$

BLEU – корпусная метрика, измеряющая степень совпадения n-грамм между сгенерированными и эталонными ответами с учетом штрафа за избыточную длину.

Формально BLEU определяется как (5):

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \quad (5)$$

где: p_n – точность n-грамм, w_n – веса n-грамм, BP – коэффициент штрафа за краткость.

В работе используется реализация SacreBLEU, обеспечивающая воспроизводимость вычислений.

С учетом структурированного характера ответов вводится Field-level Exact Match, при котором ответ представляется как набор ключевых полей: $y = \{f_1, f_2, \ldots, f_k\}$

В данной работе используются поля: вид ущерба, наименование последствия, актуальность. Метрика FieldEM определяется как доля совпавших полей (6):

$$\text{FieldEM}(\hat{y}, y) = \frac{1}{k}\sum_{i=1}^{k} l(N(\hat{f}_i) = N(f_i)), \quad (6)$$

Метрика устойчива к вариативности свободного текстового обоснования и отражает структурную корректность ответа.

Для оценки качества ключевого классификационного решения вводится метрика Actuality Exact Match, вычисляемая только по полю «Актуальность», которая определяется как (7).

$$\text{ActualityEM}\left(\hat{y}, y\right) = 1\left(\hat{a} = a\right), \quad (7)$$

Метрика напрямую отражает способность модели корректно классифицировать уровень актуальности негативного последствия.

Результаты эксперимента целесообразно представить в виде сравнения метрик до и после fine-tuning.

Анализ результатов проведенного эксперимента

Все полученные в ходе эксперимента значения метрик были сведены в единую сводную таблицу (табл. 2), что позволило обеспечить сопоставимость результатов и провести систематическое сравнение базовой и fine-tuned моделей по различным аспектам качества генерации. В таблице представлены значения лексико-семантических метрик (BLEU, F1), строгих метрик совпадения (EM_strict, EM_token), а также структурных и категориальных показателей (FieldEM, ActualityEM), что даёт возможность комплексной оценки влияния fine-tuning на поведение модели.

В таблице 2 отражены как абсолютные значения показателей, так и их относительные изменения после дообучения.

На рисунке 1 представлено сравнение моделей по значениям Exact Match для различных полей ответа. Данная группа метрик отражает способность модели строго воспроизводить эталонные формулировки без отклонений.

Таблица 2

Сравнение результатов до и после применения техники Fine-tuning

Модель	BLEU	EM-strict	EM-token	F1	Field-EM	Actuality-EM
Fine-Tuned модель	52.413	0	0.1	0.697	0.9	0.8
Базовая модель	45.331	0	0	0.646	0.9	0.7

Из графика видно, что значения Strict Exact Match остаются низкими для обеих моделей, что является ожидаемым результатом для генеративных задач, допускающих вариативность формулировок. При этом token-level Exact Match демонстрирует увеличение покрытия эталонных токенов после fine-tuning, что свидетельствует о более точном воспроизведении ключевых элементов ответа. Это указывает на повышение лексической согласованности без потери гибкости формулировок.

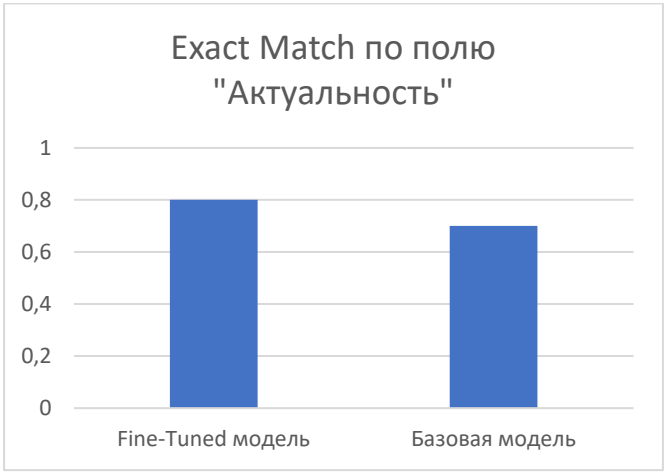

Рис. 1. Сравнение моделей по Exact Match по полям

Сравнение моделей по метрике BLEU представлено на рисунке 2. Данная метрика отражает степень совпадения n-грамм между сгенерированными и эталонными ответами и используется как индикатор лексико-фразовой согласованности.

После fine-tuning наблюдается рост значения BLEU на 7.08 пункта, что указывает на более близкое соответствие формулировок модели эталонным ответам. Это свидетельствует о том, что дообучение способствует лучшему усвоению доменной терминологии и устойчивых синтаксических конструкций, характерных для рассматриваемой предметной области.

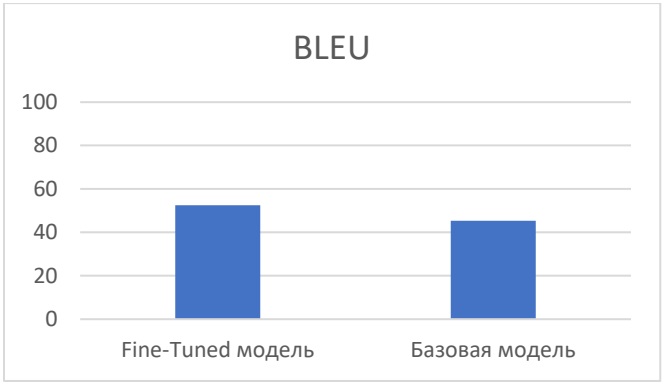

Рис. 2. Сравнение моделей по BLEU

На рисунке 3 приведено сравнение моделей по значению F1-score. Эта метрика отражает баланс между точностью и полнотой воспроизведения значимых токенов эталонного ответа.

Рост F1-score с 0.646 до 0.697 показывает, что fine-tuned модель не только чаще воспроизводит релевантные элементы, но и делает это с меньшим количеством избыточных или нерелевантных токенов. Данный результат особенно важен для задач извлечения структурированной информации, где требуется точное соответствие ключевым компонентам ответа.

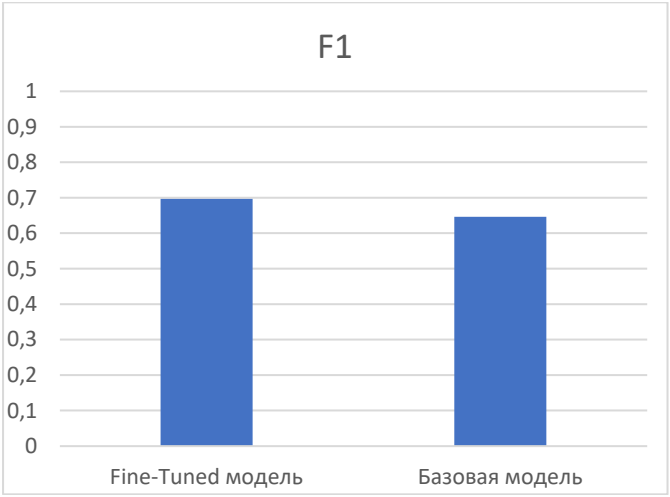

Рис. 3. Сравнение моделей по f1-score

Дополнительным индикатором эффективности fine-tuning является увеличение значения ActualityEM с 0.7 до 0.8, что указывает на более корректную классификацию уровня актуальности негативных последствий. Это демонстрирует улучшение способности модели принимать дискретные категориальные решения в условиях доменной специфики.

При этом сохранение значения FieldEM на уровне 0.9 для обеих моделей свидетельствует о том, что дообучение не привело к деградации структурной организации ответов. Модель сохраняет способность корректно заполнять ключевые поля, что является важным требованием для прикладных сценариев.

Таким образом, совокупный анализ табличных и графических представлений результатов показывает, что fine-tuning приводит к систематическому улучшению качества генерации по большинству рассматриваемых метрик. Улучшения затрагивают как лексико-семантические характеристики ответов, так и корректность классификационных решений, при сохранении структурной целостности формата вывода. Это подтверждает, что параметр-эффективное дообучение является практически оправданным инструментом адаптации LLM для специализированных прикладных задач.

Заключение

В контексте прикладных задач анализа последствий информационной безопасности эффективность LLM определяется как её способность корректно, полно и воспроизводимо формировать ответы, соответствующие заданному эталону, при сохранении семантической адекватности и структурной согласованности вывода.

Формально эффективность LLM рассматривается как совокупность следующих аспектов: лексическое соответствие эталонному ответу; семантическая близость с допустимыми перефразированиями; структурная корректность ключевых атрибутов ответа; точность принятия критических решений (например, категориальная оценка актуальности негативного последствия).

Ввиду генеративной природы LLM для оценки эффективности используется набор комплементарных метрик, каждая из которых отражает отдельный аспект качества ответа.

Многочисленные исследования показывают, что даже относительно небольшое количество качественно размеченных примеров способно существенно улучшить поведение LLM на целевой задаче при использовании supervised fine-tuning и RLHF.

Использование параметр-эффективных методов, таких как LoRA и адаптеры, позволяет достичь сопоставимого качества при заметно меньших вычислительных ресурсах, что важно для развёртывания моделей в ограниченных средах (on-premise, частные облака). В таких условиях fine-tuning становится экономически оправданной альтернативой постоянному использованию больших универсальных моделей.

При интерпретации результатов необходимо учитывать, что избыточный рост метрик на тестовой выборке может быть признаком переобучения. Поэтому важно следить за стабильностью показателей на валидационной выборке и при необходимости использовать дополнительные техники регуляризации или расширения набора данных, а также методики оценки результатов [24].

Проведённый эксперимент подтвердил, что дообучение больших языковых моделей на

специализированном корпусе существенно повышает качество генерации доменно-специфичных ответов. Изменение значений ключевых метрик после fine-tuning демонстрирует прирост как по лексико-семантическим показателям (BLEU, F1), так и по метрикам, отражающим корректность классификационных решений (ActualityEM). При этом структурные характеристики ответов, измеряемые через FieldEM, сохраняются на прежнем уровне, что свидетельствует об отсутствии деградации формата вывода при адаптации модели. Такой профиль изменений указывает, что fine-tuning обеспечивает прирост качества за счёт лучшего использования контекстуальных признаков и улучшенного воспроизведения терминологических конструкций, не нарушая при этом структуру целевых ответов [25-31].

Полученные результаты подтверждают практическую целесообразность параметр-эффективного fine-tuning в задачах анализа последствий информационной безопасности, где требуется одновременно формализованность, воспроизводимость и корректность принятия классификационных решений. Использование набора комплементарных метрик, включающих как строгие (Exact Match), так и частичные меры совпадения (F1, BLEU), обеспечивает многомерную оценку качества и позволяет достоверно выявлять улучшения модели по различным аспектам поведения. Данный подход может служить основой для последующей оценки эффективности моделей при внедрении в корпоративные процессы, а также для сравнения различных стратегий адаптации.

Дальнейшее развитие работы видится в расширении масштаба эксперимента и включении дополнительных сценариев тестирования, ориентированных на устойчивость моделей к вариативности лингвистических формулировок, а также в применении более сложных оценочных методик, учитывающих влияние длины вывода, консистентность аргументации и устойчивость к некорректным входным данным. В совокупности это позволит формировать не только количественную, но и качественную картину применимости fine-tuned моделей для задач промышленного уровня.

Литература

1. *Гадасин Д. В., Пак Е. В., Коровушкина В. М., Мелькова Е. К.* Предобработка текстовой информации на основе термов естественного языка // REDS: Телекоммуникационные устройства и системы. 2022. Т. 12, № 1. С. 4-11. EDN PDGAVP.

2. *Золотарева П. Ю., Гадасин Д. В., Маклачков К. А.* Методы обработки информации в распределенных информационных системах // Тенденции развития Интернет и цифровой экономики : Труды VI Международной научно-практической конференции, Симферополь-Алушта, 01-03 июня 2023 года. Симферополь: ИП Зуева, 2023. С. 187-189. EDN LGONZK.

3. *Hu E., Shen Y., Wallis P.* et al. LoRA: Low-Rank Adaptation of Large Language Models // Proceedings of the International Conference on Learning Representations (ICLR). 2022. URL: https://arxiv.org/abs/2106.09685 (дата обращения: 13.01.2026).

4. *Гордеев Д. С., Шевелев С. В.* Интернет вещей для "умной" городской среды // Вестник связи. 2019. № 6. С. 3-7. EDN DDZDES.

5. *Gadasin D. V., Shvedov A. V., Vakurin I. S.* Determination of Semantic Proximity of Natural Language Terms for Subsequent Neural Network Training // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 – Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744290. EDN LASMDY.

6. *Гадасин Д. В., Шведов А. В., Мелькова Е. К.* Структурирование данных исходя из центра масс // Актуальные проблемы и перспективы развития экономики : Труды XXI Международной научно-практической конференции, Симферополь-Гурзуф, 20-22 октября 2022 года. Симферополь: Крымский федеральный университет им. В.И. Вернадского, 2022. С. 266-268. EDN RFCCST.

7. *Гадасин Д. В.* Построение бинарного дерева минимальной цены // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 11. С. 38-44. DOI 10.36724/2072-8735-2024-18-11-38-44. EDN GMCEWG.

8. *Dettmers T., Pagnoni A., Holtzman A., Zettlemoyer L.* QLoRA: Efficient Finetuning of Quantized LLMs // Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL). 2023, pp. 100-114. URL: https://arxiv.org/abs/2305.14314 (дата обращения: 13.01.2026).

9. *Lialin V., Deshpande A., Rumshisky A.* Scaling Down to Scale Up: A Guide to Parameter-Efficient Fine-Tuning // Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2023, pp. 548-566. URL: https://arxiv.org/abs/2303.15647 (дата обращения: 13.01.2026).

10. *Gadasin D. V., Koltsova A. V., Gadasin D. D.* Algorithm for Building a Cluster for Implementing the 'Memory as a Service' Service in the IoT Concept // 2021 Systems of Signals Generating and Processing in the Field of on Board Communications, Conference Proceedings, Moscow, 16-18 марта 2021 года. Moscow, 2021. P. 9416112. DOI 10.1109/IEEECONF51389.2021.9416112. EDN VRPCFG.

11. *Гадасин Д. В., Смальков Н. А., Кузин И. А.* Использование метода роя частиц для балансировки нагрузки в сетях Интернета вещей // Системы синхронизации, формирования и обработки сигналов. 2022. Т. 13, № 2. С. 17-23. EDN LIUWNT.

12. *Gadasin D. V., Shvedov A. V., Yudin A. A.* Clustering methods in large-scale systems // Synchroinfo Journal. 2020. Vol. 6, No. 5, pp. 21-24. DOI 10.36724/2664-066x-2020-6-5-21-24. EDN XHNSYV.

13. *Zhang S., Sun T., Shao Y. et al.* Instruction Tuning for Large Language Models: A Survey // Transactions of the Association for Computational Linguistics. 2023. Vol. 11, pp. 1–21. URL: https://arxiv.org/abs/2308.10792 (дата обращения: 13.01.2026).

14. *Papineni K., Roukos S., Ward T., Zhu W.-J.* BLEU: A Method for Automatic Evaluation of Machine Translation // Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL). Philadelphia, 2002, pp. 311-318.

15. *Lin C.-Y.* ROUGE: A Package for Automatic Evaluation of Summaries // Proceedings of the ACL Workshop on Text Summarization Branches Out. Barcelona, 2004, pp. 74-81.

16. *Zolotukhin P. A., Melkova E. K., Gadasin D. V., Korovushkina V. M.* Using Intelligent Testing as a Tool to Improve the Quality of Transport Information Systems // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 – Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744348. EDN NOMJLX.

17. *Гадасин Д. В., Шведов А. В.* Применение транспортной задачи для балансировки нагрузки в условиях нечеткости исходных данных // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 1. С. 13-20. DOI 10.36724/2072-8735-2024-18-1-13-20. EDN WKNPIX.

18. *Shvedov A. V., Gadasin D. V., Pak E. V.* Application of the Backman Model for the Distribution of Traffic Flows in Networks with Segment Routing // 2022 Systems of Signals Generating and Processing in the Field of on Board Communications, SOSG 2022 – Conference Proceedings, Moscow, 15-17 марта 2022 года. Moscow, 2022. DOI 10.1109/IEEECONF53456.2022.9744344. EDN RBMTBQ.

19. *Banerjee S., Lavie A.* METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments // Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization. – 2005, pp. 65-72.

20. *Gadasin D. V., Shvedov A. V., Kuzin I. A.* Reconstruction of a Three-Dimensional Scene from its Projections in Computer Vision Systems // 2021 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex, TIRVED 2021 – Conference Proceedings, Moscow, 11-12 ноября 2021 года. Moscow, 2021. DOI 10.1109/TIRVED53476.2021.9639161. EDN CKSNPA.

21. *Gadasin D. V., Shvedov A. V., Kuzin I. A.* A model for representing the color and depth metric characteristics of objects in an image // 2021 Systems of Signal Synchronization, Generating and Processing in Telecommunications, SYNCHROINFO 2021 – Conference Proceedings, Svetlogorsk, Kaliningrad Region, 30 июня – 02 июля 2021 года. Svetlogorsk, Kaliningrad Region, 2021. P. 9488349. DOI 10.1109/SYNCHROINFO51390.2021.9488349. EDN YAYZVP.

22. *Liang P., Bommasani R., Lee T. et al.* Holistic Evaluation of Language Models (HELM) // Proceedings of NeurIPS. 2022. URL: https://arxiv.org/abs/2211.09110 (дата обращения: 13.01.2026).

23. *Chiang T. H., Lee C.-H., Chen Y.-N. et al.* ChatGPT is a General-Purpose Natural Language Processing Model: A Survey // IEEE Access. 2023.

24. *Милицин, Ю. А., Шевелев С. В.* Методика оценки ценности результатов НИР // Вестник связи. 2014. № 3. С. 45-48. EDN UHLXRB.

25. *Gadasin D. V., Shvedov A. V., Klygina O. G.* Organization of Interaction Between the Concept of Fog Computing and Segment Routing for the Provision of IoT Services in Smart Grid Networks // Wave Electronics and Its Application in Information and Telecommunication Systems. 2022. Vol. 5, No. 1, pp. 141-146. EDN UQSHRH.

26. *Гадасин Д. В., Шведов А. В.* Проблемы интеграции концепции "Интернет вещей" и облачных вычислений // Технологии информационного общества : Материалы XIII Международной отраслевой научно-технической конференции, Москва, 20-21 марта 2019 года. Т. 2. М.: Издательский дом Медиа Паблишер, 2019. С. 22-23. EDN MEQRFA.

27. *Гадасин Д. В., Шведов А. В., Алексеева Е. А.* Информационная энтропия в стохастических сетях связи // Телекоммуникационные и вычислительные системы 2020 : Труды международной научно-технической конференции, Москва, 14-17 декабря 2020 года / Московский технический университет связи и информатики. М.: Горячая линия – Телеком, 2020. С. 108-116. EDN IOGLQH.

28. *Гадасин Д. В., Шведов А. В., Кузин И. А.* Трехмерная реконструкции объекта по одному изображению с использованием глубоких свёрточных нейронных сетей // T-Comm: Телекоммуникации и транспорт. 2022. Т. 16, № 7. С. 29-35. DOI 10.36724/2072-8735-2022-16-7-29-35. EDN YTLCNW.

29. *Shvedov A. V., Gadasin D. V., Alyoshintsev A. V.* Segment routing in data transmission networks // T-Comm: Телекоммуникации и транспорт. 2022. Vol. 16, No. 5, pp. 56-62. DOI 10.36724/2072-8735-2022-16-5-56-62. EDN VAYLJQ.

30. *Alyoshintsev A. V., Gadasin D. V., Vakurin D. S., Chelyshkov P. D.* Methods for evaluating the noise immunity of modems // T-Comm: Телекоммуникации и транспорт. 2025. Vol. 19, No. 9, pp. 50-58. DOI 10.36724/2072-8735-2025-19-9-50-58. EDN TGKCQD.

31. *Гадасин Д. В.* Способ определения основных узлов сети для анализа ее состояния // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19, № 12. С. 16-24. DOI 10.36724/2072-8735-2025-19-12-16-24. EDN FGAATI.

ОСОБЕННОСТИ ПРИМЕНЕНИЯ ФАЗЗИНГ-ТЕСТИРОВАНИЯ ДЛЯ РАЗЛИЧНЫХ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Гадасин Даниил Денисович
Московский технический университет связи и информатики, студент группы М092501(70), Москва, Россия
d.d.gadasin@mtuci.ru

Маклачкова Виктория Валентиновна
Московский технический университет связи и информатики, ст. преп. кафедры СИТиС, Москва, Россия
v.v.maklachkova@mtuci.ru

Аннотация
В статье рассматриваются современные подходы к фаззинг-тестированию как методу динамического анализа программного обеспечения, направленному на выявление дефектов, уязвимостей и ошибок выполнения. Особое внимание уделяется сравнению фаззинг-инструментов и методик, применяемых в различных языковых экосистемах, включая языки с ручным управлением памятью и управляемые языки программирования. Анализируются особенности генерации входных данных, механизмы сбора информации о покрытии кода, а также роль динамических анализаторов и средств инструментирования. В работе приводятся статистические данные, подтверждающие рост масштабов и эффективности фаззинг-тестирования в современных open-source проектах. Рассматриваются преимущества и ограничения фаззинга в зависимости от используемого языка программирования и среды выполнения, а также области его практического применения для повышения надёжности и устойчивости программных систем.

Ключевые слова
фаззинг-тестирование, динамический анализ, тестирование программного обеспечения, генерация входных данных, покрытие кода, уязвимости программного обеспечения, языки программирования, OSS-Fuzz.

Введение

Фаззинг-тестирование (англ. fuzz testing, fuzzing) является одним из наиболее эффективных и широко применяемых методов динамического анализа программного обеспечения, направленным на выявление ошибок, уязвимостей и нестандартных состояний выполнения программы. Суть фаззинга заключается в автоматической генерации большого количества входных данных, часто случайных или частично модифицированных, которые подаются на вход тестируемой программы с целью спровоцировать сбои, аварийные завершения, утечки памяти (англ. memory leak) или некорректное поведение.

Актуальность фаззинг-тестирования подтверждается статистическими данными сообществ и открытых инфраструктур. Так, согласно данным сервиса Open Source Fuzzing Introspection [1] — который агрегирует информацию о состоянии fuzz-тестирования проектов, интегрированных в платформу OSS-Fuzz – на сегодняшний день в OSS-Fuzz включено более 1300 open-source проектов, для которых существует множество программных обёрток для фаззинг-тестирования (англ. fuzzing harness) и инструментальных сборок. Среди этих проектов фиксируется тысячи отдельных фаззинг-инструментов (фазеров) и миллионы проанализированных функций, распределённых по языкам программирования: например, Python, C, C++, Java, Go и Rust, что отражает широкое проникновение и разнообразие применения фаззинга в различных языковых экосистемах, на рисунке 1 представлено распределение проектов OSS-Fuzz по языкам программирования [1].

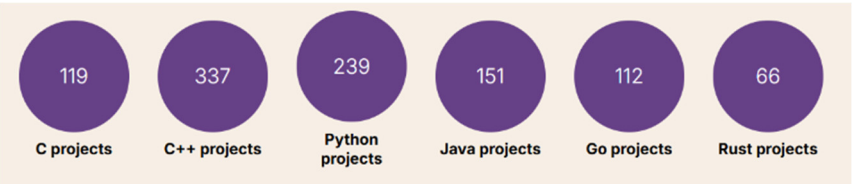

Рис. 1. Распределение проектов OSS-Fuzz по языкам программирования

Первоначально фаззинг применялся преимущественно для тестирования системного программного обеспечения, написанного на языках низкого уровня, таких как C и C++. Однако с развитием языков программирования, ростом популярности управляемых сред выполнения и появлением сложных высокоуровневых фреймворков, подходы к фаззинг-тестированию начали активно адаптироваться под различные языковые экосистемы. Сегодня существуют специализированные фаззеры для C/C++, Java, Python, JavaScript, Go, Rust и других языков, каждый из которых учитывает особенности модели памяти, исполнения кода и обработки исключений.

Целью данной статьи является сравнение подходов фаззинг-тестирования под различные языки программирования, анализ их сильных и слабых сторон, а также выявление факторов, влияющих на эффективность фаззинга в зависимости от используемого языка.

Общие принципы фаззинг-тестирования

Независимо от используемого языка программирования и конкретной реализации инструмента, фаззинг-тестирование базируется на ряде фундаментальных принципов, определяющих его эффективность и область применимости. Прежде всего, фаззинг требует наличия чётко определённой точки входа — функции, метода, интерфейса прикладного программирования (API), обработчика протокола или иного компонента, принимающего внешние данные. Именно такие точки входа наиболее подвержены ошибкам, поскольку обрабатывают потенциально некорректный или злонамеренно сформированный ввод.

Вторым ключевым принципом является автоматизированная генерация входных данных. В зависимости от подхода фаззер может либо создавать данные «с нуля», либо модифицировать заранее подготовленные корректные примеры. Генерация должна быть масштабируемой и способной покрывать широкий диапазон входных состояний, включая граничные значения, нестандартные комбинации параметров и некорректные форматы. При этом качество входных данных напрямую влияет на глубину анализа и вероятность достижения редких, но критичных состояний выполнения программы.

Современные фаззинг-инструменты могут классифицироваться по стратегии генерации входных данных и использованию обратной связи от выполнения программы. В простейшем случае применяется случайная генерация входных данных, не учитывающая структуру формата и поведение тестируемого кода. Более практичными являются подходы, основанные на модификации заранее подготовленного набора корректных входных данных. Наиболее развитые решения дополняют процесс генерации механизмами анализа покрытия кода, что позволяет адаптивно направлять процесс тестирования и повышать его результативность. Общая последовательность функционирования фаззинг-инструмента представлена на рисунке 2.

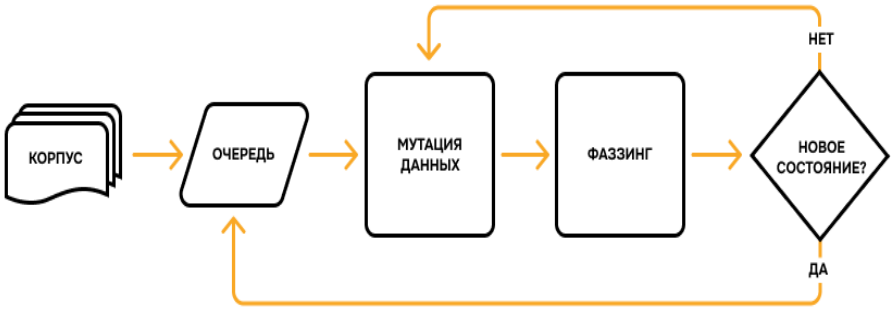

Рис. 2. Общая последовательность работы фаззинг-инструмента

Неотъемлемым элементом фаззинг-тестирования является мониторинг выполнения программы. Фаззер должен уметь обнаруживать и фиксировать аномальные ситуации, такие как аварийные завершения, необработанные исключения, нарушения доступа к памяти, зависания или логические ошибки. В системах, ориентированных на языки с ручным управлением памятью (например, C и C++), особое внимание уделяется детектированию переполнений буфера (англ. buffer overflow), использованию освобожденной области памяти (англ. use-after-free) и двойного освобождения памяти (англ. double free). В управляемых средах (Java, Python, C#) акцент смещается в сторону выявления логических ошибок, некорректных состояний объектов и необработанных исключений.

Современные фаззеры условно делятся на несколько типов в зависимости от стратегии генерации входных данных. Наиболее простым является случайный фаззинг (англ. random fuzzing), при котором

входные данные формируются полностью случайным образом. Несмотря на простоту реализации, данный подход обладает низкой эффективностью при тестировании сложных форматов и редко достигает глубоко вложенных участков кода.

Более развитым является мутационный фаззинг, основанный на использовании начального набора корректных входных данных (seed-корпуса). Эти данные подвергаются систематическим модификациям, что позволяет сохранять частичную валидность ввода и существенно повышает вероятность выявления ошибок.

Наиболее результативным с практической точки зрения считается фаззинг с управлением по покрытию кода (англ. coverage-guided fuzzing), лежащий в основе таких инструментов, как AFL++, libFuzzer и Honggfuzz. Использование информации о покрытии позволяет эффективно исследовать большие и сложные кодовые базы при ограниченных вычислительных ресурсах.

Отдельным важным принципом является воспроизводимость результатов фаззинг-тестирования. Обнаруженная ошибка должна быть зафиксирована таким образом, чтобы разработчик мог воспроизвести сбой с минимальными усилиями. Для этого фаззеры сохраняют входные данные, вызвавшие аномалию, а также дополнительную информацию о состоянии выполнения программы, что особенно важно при интеграции фаззинга в процессы безопасной разработки программного обеспечения (Secure SDLC).

Выбор конкретного подхода, стратегии генерации данных и инструмента фаззинг-тестирования во многом определяется языком программирования и средой выполнения. Языковые особенности определяют доступность информации о покрытии, стоимость одного тестового запуска, типы характерных ошибок и сложность инструментирования. В связи с этим универсальные принципы фаззинга реализуются по-разному в различных языковых экосистемах, что и обуславливает необходимость их сравнительного анализа.

Нормативные требования к проведению фаззинг-тестирования

Несмотря на то, что фаззинг-тестирование как метод возник и активно развивался в инженерной и исследовательской среде, его применение в промышленной разработке программного обеспечения всё чаще соотносится с требованиями действующих стандартов в области качества и безопасности. В российской и международной практике фаззинг напрямую не всегда выделяется как отдельный метод, однако он полностью укладывается в рамки динамического тестирования и анализа программного обеспечения.

В контексте российских стандартов фаззинг-тестирование соотносится прежде всего с требованиями ГОСТ 28195-2014 [2], где указывается необходимость проведения испытаний, направленных на выявление отказов и некорректного поведения программ при ошибочных или экстремальных входных данных. Фаззинг в данном случае выступает как автоматизированный способ реализации таких испытаний.

Также релевантным является ГОСТ Р ИСО/МЭК 25010-2015 [3], определяющий модель качества программного обеспечения. Использование фаззинг-тестирования способствует повышению показателей надёжности, отказоустойчивости и защищённости, так как позволяет выявлять дефекты, приводящие к аварийным завершениям, утечкам памяти и уязвимостям программного обеспечения и информационной системы в целом [4,5].

В рамках процессов жизненного цикла программного обеспечения фаззинг может рассматриваться как часть верификации и валидации, описанных в ГОСТ Р ИСО/МЭК 12207-2010 [6], а также как элемент непрерывного тестирования в современных процессах разработки программного обеспечения [7]. В данном стандарте подчёркивается необходимость применения динамических методов тестирования для подтверждения корректности реализации и устойчивости программных компонентов.

Фаззинг для языков низкого уровня: C и C++

Языки C и C++ традиционно считаются основной и наиболее приоритетной целью фаззинг-тестирования. Это обусловлено отсутствием встроенных механизмов защиты памяти, ручным управлением ресурсами, слабой изоляцией данных и высокой плотностью низкоуровневых операций. Подобные особенности приводят к высокой вероятности возникновения критических дефектов, включая переполнение буфера, выход за границы массива, использованию освобожденной области памяти, двойное освобождение памяти, разыменование нулевых или висячих указателей, а также неопределённое поведение (англ. undefined behavior).

Для программ, написанных на C и C++, фаззинг часто является наиболее эффективным способом выявления уязвимостей безопасности, так как многие ошибки управления памятью могут не проявляться при обычном функциональном тестировании. Особенно уязвимыми являются компоненты, обрабатывающие внешние данные: парсеры файловых форматов, сетевые протоколы, библиотеки сериализации, драйверы и системные утилиты.

Наиболее распространённым и результативным подходом для C/C++ является фаззинг с управлением по покрытию кода (англ. coverage-guided) с использованием инструментализации на этапе компиляции. Исходный код компилируется с добавлением специальных инструкций, позволяющих фаззеру получать информацию о выполненных базовых блоках и ветвях. Данный подход даёт возможность направить процесс генерации входных в недоступные ранее участки программы во время выполнения программы. Основными инструментами данного типа являются AFL, AFL++, libFuzzer и Honggfuzz.

В языках C и C++ широко применяется внутрипроцессный (in-process) фаззинг, при котором тестируемая функция вызывается фаззером напрямую, не создавая для этого отдельный процесс. Данный подход обеспечивает высокую скорость выполнения, что очень важно при анализе большого количества кода, однако данный метод требует аккуратного восстановления состояния программы между запусками и повышенного внимания над управлением памятью.

Важную роль в фаззинге языков C и C++ играют динамические анализаторы и санитайзеры. Анализатор адресов памяти (AddressSanitizer, ASan) используется для обнаружения ошибок работы с памятью, таких как buffer overflow и use-after-free. Анализатор неопределённого поведения (UndefinedBehaviorSanitizer, UBSan) позволяет выявлять неопределённое поведение, а анализатор использования неинициализированной памяти (MemorySanitizer, MSan) — использование неинициализированной памяти. Хотя применение санитайзеров увеличивает накладные расходы и снижает производительность, их использование существенно повышает полноту и точность обнаружения дефектов.

Преимуществом фаззинга для C и C++ является высокая вероятность выявления критических и потенциально эксплуатируемых уязвимостей, что делает данный подход незаменимым в задачах обеспечения безопасности программного обеспечения. Вместе с тем к недостаткам относятся высокая стоимость одного тестового запуска, сложность настройки инструментализации и необходимость значительных вычислительных ресурсов. Эффективное применение фаззинга в данной области требует тщательной подготовки входных данных, корректной конфигурации фаззера и глубокого понимания архитектуры тестируемого программного обеспечения.

Фаззинг для управляемых языков: Java и C#

Управляемые языки программирования, такие как Java и C#, используют виртуальную машину (JVM и CLR соответственно) и автоматическое управление памятью. Наличие механизма автоматической очистки неиспользуемых объектов (сборщика мусора, англ. Garbage Collector), строгой типизации и встроенных механизмов проверки границ массивов существенно снижает вероятность классических ошибок управления памятью, характерных для C/C++. Однако это не означает отсутствия уязвимостей как таковых. На практике в приложениях на Java и C# часто встречаются логические ошибки, некорректная обработка пользовательского ввода, ошибки сериализации и десериализации, а также уязвимости, приводящие к отказу в обслуживании (Denial of Service).

Фаззинг-тестирование для Java и C# преимущественно ориентировано на тестирование высокоуровневых компонентов: публичных API, библиотек, сервисных интерфейсов и сетевых протоколов. Важной особенностью является необходимость генерации структурированных входных данных, соответствующих сигнатурам методов, типам параметров и контрактам интерфейсов. В отличие от низкоуровневых языков, фаззинг, основанный на случайной генерации байтовых последовательностей, как правило, оказывается малоэффективным, так как большая часть таких входов отбрасывается на этапе проверки типов или парсинга.

Существенную роль в фаззинге управляемых языков играет уровень абстракции, предоставляемый виртуальной машиной. Инструменты фаззинга могут использовать возможности JVM и CLR для сбора информации о покрытии кода, выполнении методов, ветвлений и исключений. Это упрощает процесс встраивания дополнительных инструментов по сравнению с языками C и C++, в которых требуется модификация машинного кода или компиляция с особыми флагами, однако наличие JIT-компиляции и динамической типизации может влиять на стабильность сбора покрытия кода и воспроизводимость результатов фаззинга.

Для Java и C# в основном применяются специализированные фаззинг-фреймворки, ориентированные на уровень языка и экосистемы. В Java таковыми являются, например, Jazzer, JQF и интеграции с OSS-Fuzz, которые используют подход с управлением по покрытию кода и напрямую взаимодействуют с байткодом JVM. В экосистеме .NET применяются инструменты, ориентированные на CLR и управляемый код, это включает фаззинг на уровне библиотек и сетевых компонентов.

Преимуществом фаззинга управляемых языков является большая стабильность среды выполнения и меньший риск аварийного завершения тестирования по сравнению с языками C и C++. Однако основными ограничениями является сложность генерации валидных входных данных и меньшая вероятность обнаружения критических уязвимостей.

Фаззинг для языков Java и C# чаще используется как инструмент повышения надёжности и устойчивости информационных систем, а также для выявления ошибок, способных привести к отказу в обслуживании или нарушению бизнес-логики программного обеспечения.

Фаззинг для динамически типизированных языков: Python и JavaScript

Динамически типизированные языки программирования, такие как Python и JavaScript, характеризуются высокой гибкостью и выразительностью, что упрощает разработку, но одновременно повышает риск возникновения ошибок времени выполнения. Отсутствие строгой статической типизации приводит к тому, что значительная часть ошибок выявляется только во время исполнения программы. Типичными проблемами являются исключения, связанные с несовпадением типов, некорректными структурами данных, ошибками преобразования форматов, а также нарушениями логики обработки пользовательского ввода.

Фаззинг-тестирование для Python и JavaScript чаще всего реализуется на уровне интерпретатора или среды выполнения и ориентировано на тестирование функций, модулей и библиотек, принимающих внешние данные. Одним из самых эффективных подходов является генерация корректных, с точки зрения синтаксиса языка, данных включая объекты, JSON-структуры или HTTP запросы. Данный подход, по сравнению с генерацией случайных байтов, позволяет достичь большего покрытия кода.

Сбор информации о покрытии кода в языках такого типа осуществляется за счет средств интерпретатора или библиотек, не входящих в стандартный набор языка. Это позволяет достаточно точно отслеживать выполнение функций, ветвлений и исключений, однако наличие динамических конструкций, таких как динамическое выполнение кода (англ. eval), рефлексия и динамическая подгрузка модулей, может усложнять анализ результатов фаззинга и снижать воспроизводимость найденных ошибок.

Преимуществом фаззинга для Python и JavaScript является низкая стоимость одного тестового запуска и высокая скорость итераций, что обеспечивает быструю обратную связь. Это делает фаззинг удобным инструментом для регулярного тестирования в процессе разработки. В то же время значительная часть обнаруженных дефектов не относится к критическим уязвимостям безопасности, а связана с устойчивостью приложения, обработкой ошибок и корректностью бизнес-логики. Тем не менее для библиотек, веб-приложений и сервисов, активно работающих с внешними данными, фаззинг в таких языках остаётся важным средством повышения надёжности и качества программного обеспечения.

Фаззинг для современных системных языков: Go и Rust

Современные системные языки программирования Go и Rust занимают промежуточное положение между традиционными низкоуровневыми языками и управляемыми средами выполнения. Go использует строгую статическую типизацию и автоматическое управление памятью с помощью механизма автоматической очистки неиспользуемых объектов, в то время как Rust применяет модель владения и заимствований, обеспечивающую безопасность памяти на этапе компиляции без использования механизма автоматической очистки неиспользуемых объектов.

Фаззинг для Go активно поддерживается на уровне стандартного инструментария языка. Начиная с последних версий, фаззинг интегрирован непосредственно в систему тестирования, что упрощает его использование и делает фаззинг частью повседневного процесса разработки. Основное внимание при фаззинг-тестировании Go-программ уделяется выявлению паник, некорректных состояний, ошибок обработки входных данных и проблем синхронизации в конкурентных программах. При этом сбор информации о покрытии осуществляется средствами компилятора и среды исполнения программы, что обеспечивает высокую эффективность подходов с управлением по покрытию кода.

Rust, благодаря строгой системе типов, проверке заимствований и контролю владения памятью, существенно снижает количество уязвимостей, связанных с некорректным доступом к памяти. Тем не менее фаззинг остаётся актуальным инструментом для выявления логических ошибок, некорректных переходов между состояниями и проблем во взаимодействии с небезопасным (англ. unsafe) кодом, который широко используется для интеграции с системными библиотеками и кодом на C/C++.

Подходы к фаззинг-тестированию в Rust во многом заимствуют идеи, применяемые в экосистеме C/C++, включая фаззинг с управлением по покрытию кода и инструментализацию на этапе компиляции. При этом они адаптированы под особенности компилятора и экосистемы Rust, что позволяет эффективно сочетать высокую производительность фаззинга с дополнительными гарантиями безопасности, предоставляемыми языком. В результате фаззинг для Go и Rust рассматривается как важный элемент обеспечения надёжности и устойчивости системного программного обеспечения, особенно в компонентах, обрабатывающих недоверенные входные данные.

Заключение

Фаззинг-тестирование зарекомендовало себя как универсальный и высокоэффективный метод динамического анализа программного обеспечения, применимый на различных этапах жизненного цикла разработки. Несмотря на общую идею автоматической генерации некорректных входных данных, практическая реализация фаззинга существенно различается в зависимости от языка программирования, среды выполнения и целевого класса дефектов [8].

Проведённое исследование показало, что эффективность фаззинг-тестирования напрямую зависит от адаптации метода к особенностям языка программирования и среды выполнения.

Таким образом, можно сделать вывод, что универсальных фаззинг-решений не существует. Выбор инструмента, стратегии генерации входных данных и уровня инструментализации должен осуществляться с учётом конкретных целей тестирования. В этом контексте фаззинг выступает не как замена традиционных методов контроля качества [9], а как их важное дополнение, позволяющее существенно повысить надёжность и безопасность программного обеспечения [10-21].

Литература

1. OSS-Fuzz Introspector [Электронный ресурс]. Режим доступа: https://introspector.oss-fuzz.com/ (дата обращения: 28.01.2026).

2. ГОСТ 28195-2014. Оценка качества программных средств. Введ. 2015-07-01. М.: Стандартинформ, 2015. 20 с.

3. ГОСТ Р ИСО/МЭК 25010-2015. Системная и программная инженерия. Требования и оценка качества систем и программных средств (SQuaRE). Модель качества систем и программных средств. Введ. 2016-07-01. М.: Стандартинформ, 2016. 28 с.

4. *Dokuchaev V. A., Maklachkova V. V., Statev V. Yu.* Classification of personal data security threats in information systems // T-Comm: Телекоммуникации и транспорт. 2020. Vol. 14, No. 1, pp. 56-60. DOI 10.36724/2072-8735-2020-14-1-56-60. EDN QOGYHH.

5. *Маклачкова В. В.* Основные риски персональных данных в мультиоблачных информационных средах // Экономика и качество систем связи. 2025. № 3(37). С. 169-181.

6. ГОСТ Р ИСО/МЭК 12207-2010. Системная и программная инженерия. Процессы жизненного цикла программных средств. Введ. 2012-03-01. М.: Стандартинформ, 2012. 104 с.

7. *Chen C., Kim S. Y., et al.* A systematic review of fuzzing techniques // Computers & Security. Vol. 75, pp. 118-137. 2018.

8. *Keller B. N., Meyers B. S., Meneely A.* What Happens When We Fuzz? Investigating OSS-Fuzz Bug History // 2023 IEEE/ACM 20th International Conference on Mining Software Repositories (MSR), Melbourne, Australia, 2023, pp. 207-217, doi: 10.1109/MSR59073.2023.00038.

9. *Shirai T., Nourry O., Kashiwa Y., et al*. Large-Scale Empirical Analysis of Continuous Fuzzing: Insights from 1 Million Fuzzing Sessions // 2025 IEEE Transactions on software engineering, Vol. 14, No. 8, AUGUST 2025.

10. *Статьев В. Ю., Докучаев В. А., Маклачкова В. В*. Информационная безопасность на пространстве "Больших данных" // T-Comm: Телекоммуникации и транспорт. 2022. Т. 16, № 4. С. 21-28. DOI 10.36724/2072-8735-2022-16-4-21-28. EDN IXUYWS.

11. *Dokuchaev V. A., Maklachkova V. V., Statyev V. Yu.* Data subject as augmented reality // Synchroinfo Journal. 2020. Vol. 6, No. 1, pp. 11-15. DOI 10.36724/2664-066X-2020-6-1-11-15. EDN ULPVZC.

12. *Шведов А. В., Гадасин Д. В., Коровушкина В. М., Мелькова Е. К.* Интеллектуальное тестирование как способ повышения качества информационной системы // REDS: Телекоммуникационные устройства и системы. 2022. Т. 12, № 2. С. 43-52. EDN GOLZGE.

13. *Гадасин Д. В., Шведов А. В., Кузин И. А.* Трехмерная реконструкции объекта по одному изображению с использованием глубоких свёрточных нейронных сетей // T-Comm: Телекоммуникации и транспорт. 2022. Т. 16, № 7. С. 29-35. DOI 10.36724/2072-8735-2022-16-7-29-35. EDN YTLCNW.

14. *Shvedov A. V., Gadasin D. V., Alyoshintsev A. V*. Segment routing in data transmission networks // T-Comm: Телекоммуникации и транспорт. 2022. Vol. 16, No. 5, pp. 56-62. DOI 10.36724/2072-8735-2022-16-5-56-62. EDN VAYLJQ.

15. *Alyoshintsev A. V., Gadasin D. V., Vakurin D. S., Chelyshkov P. D*. Methods for evaluating the noise immunity of modems // T-Comm: Телекоммуникации и транспорт. 2025. Vol. 19, No. 9, pp. 50-58. DOI 10.36724/2072-8735-2025-19-9-50-58. EDN TGKCQD.

16. *Гадасин Д. В.* Способ определения основных узлов сети для анализа ее состояния // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19, № 12. С. 16-24. DOI 10.36724/2072-8735-2025-19-12-16-24. EDN FGAATI.

17. *Мелькова Е. К., Шведов А. В., Тремасова Л. А., Гадасин Д. В*. Организация кластера исходя из функции принадлежности // Системы синхронизации, формирования и обработки сигналов. 2023. Т. 14, № 1. С. 30-39. EDN CNVIJU.

18. *Зимнин А. С., Гадасин Д. Д., Галицкий М. В.* и др. Сложности цифровой трансформации предприятий малого и среднего бизнеса // Методические вопросы преподавания инфокоммуникаций в высшей школе. 2025. Т. 14, № 3. С. 27-33. EDN QGFLBX.

19. *Маклачкова В. В., Гадасин Д. В., Литвинов Д. С., Кувшинов М. И*. Проектирование системы поддержки принятия решений по подбору подарка // DSPA: Вопросы применения цифровой обработки сигналов. 2024. Т. 14, № 1. С. 51-61. EDN CVTSLG.

20. *Павлов С. В., Докучаев В. А., Маклачкова В. В., Гадасин Д. Д.* Автоматизация поддержки принятия решений при подборе специалистов по управлению сложными инфокоммуникационными системами // T-Comm: Телекоммуникации и транспорт. 2023. Т. 17, № 12. С. 36-43. DOI 10.36724/2072-8735-2023-17-12-36-43. EDN DWHCMB.

21. *Плотников П. С., Неронов Ф. А., Маклачкова В. В.* Анализ уязвимостей технологии блокчейн // DSPA: Вопросы применения цифровой обработки сигналов. 2022. Т. 12, № 3. С. 33-38. EDN PQOXCT.

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ПРОЦЕССЕ РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПРОМЫШЛЕННОЙ ПЛОЩАДКОЙ

Ковтун Игорь Иванович
Московский технический университет связи и информатики,
к.т.н., доцент, доцент кафедры «Системное программирование», Москва, Россия,
i.i.kovtun@mtuci.ru

Лебедева Елизавета Игоревна
Московский технический университет связи и информатики,
студент группы БВТ2403, Москва, Россия,
tjc.work-source@yandex.ru

Аннотация
Рассмотрен подход к автоматизации и интеллектуализации процессов управления активами промышленной недвижимости. В целях повышения эффективности процессов комплексной автоматизации промышленных площадок и мощностей обоснована целесообразность интеграции функционального и реляционного моделирования. При этом, предпринимается попытка применения методов искусственного интеллекта, в т.ч. заполнения слотов заранее предопределенных SQL-фреймов, полученных по результатам анализа функциональной модели. Описаны подходы к разработке АРМ с использованием SQL-технологий.

Ключевые слова
промышленная площадка, комплексная автоматизация, автоматизированное рабочее место, методология IDEF0, реляционная модель данных, Structured Query Language, СУБД PostgreSQL, SQL-фрейм

Введение

В современных условиях эффективное использование производственных мощностей в крупных промышленных регионах России, таких как Центральный федеральный округ, является стратегической задачей государственного масштаба. В соответствии с промышленной политикой Российской Федерации рост эффективности эксплуатации индустриальных промышленных площадок напрямую влияет на темпы импортозамещения и технологического суверенитета [1]. Реализация данных процессов опирается на обширную нормативно-правовую базу, в т.ч. Гражданский кодекс Российской Федерации, см. гл. 34 «Аренда», Федеральный закон от 31.12.2014 № 488-ФЗ «О промышленной политике в Российской Федерации».

На сегодняшний день без средств автоматизации и интеллектуализации эффективное управление активами, пожалуй, невозможно. А, как известно, комплексную автоматизацию можно осуществлять различными способами. Таким образом, оказались востребованы методы выбора наилучшего способа автоматизации индустриальной промышленной площадки. При этом, рассмотрения отдельного проекта комплексной автоматизации недостаточно. Скорее, возникает задача формирования портфеля ИТ-проектов, их всесторонней, комплексной оценки с последующим выбором оптимального варианта [2].

К сожалению, традиционные подходы к проектированию автоматизированных систем не всегда позволяют достоверно описать объект обследования на ранних стадиях проектирования автоматизированной системы [3]. Так, например, традиционно используемая реляционная модель данных, при всей ее стройности, зачастую не дает понимания функциональной структуры управления предприятием [4]. В свою очередь, методология функционального моделирования IDEF0, разработанная для автоматизации промышленных предприятий [5], имея наглядность, не раскрывает структуру самой информации и способы ее хранения. Применяемые CASE-средства и модели типа RWIM и BPIN также обладают недостатками – они либо излишне перегружены специфической нотацией, затрудняющей верификацию бизнес-логики, либо имеют ограниченные возможности по бесшовной интеграции реляционных таблиц в динамические функциональные блоки. При этом, отсутствие интеллектуальных алгоритмов верификации на стыке моделей приводит к возникновению логических коллизий.

Учитывая вышесказанное, целью научной работы, итоги которой подводятся в настоящей статье, является разработка подхода к формализации процессов управления активами индустриальных площадок в интересах российских промышленных предприятий и государственных управляющих

компаний, позволяющего устранить недостатки существующих моделей, методологий и технологий проектирования автоматизированных систем посредством создания единого интегрированного решения с элементами интеллектуального контроля данных.

Задачи исследования:

– функциональное моделирование общих закономерностей процессов аренды промышленных площадей;

– разработка типовой модели информационных потоков и реляционных процессов управления промышленными площадями;

– интеграция реляционной и функциональной модели управленческих процессов в единое комплексное решение на базе SQL с внедрением алгоритмов интеллектуальной проверки данных;

– подведение итогов научного исследования и определение направлений его дальнейшего развития.

1. Функциональное моделирование основных закономерностей процессов аренды промышленных площадок

В ходе проведенного комплексного анализа бизнес-процессов управления фондом индустриальной недвижимости в промышленном регионе была построена и верифицирована функциональная модель по принципу AS-IS. Применение методологии функционального моделирования IDEF0 позволило декомпозировать сложную систему управления имущественным комплексом на элементарные функции, установив выявленные связи между информационными потоками. Установлено, что процесс обеспечения арендной деятельности в интересах отечественных промышленных предприятий целесообразно свести к виду, фрагмент которого представлен на рис. 1.

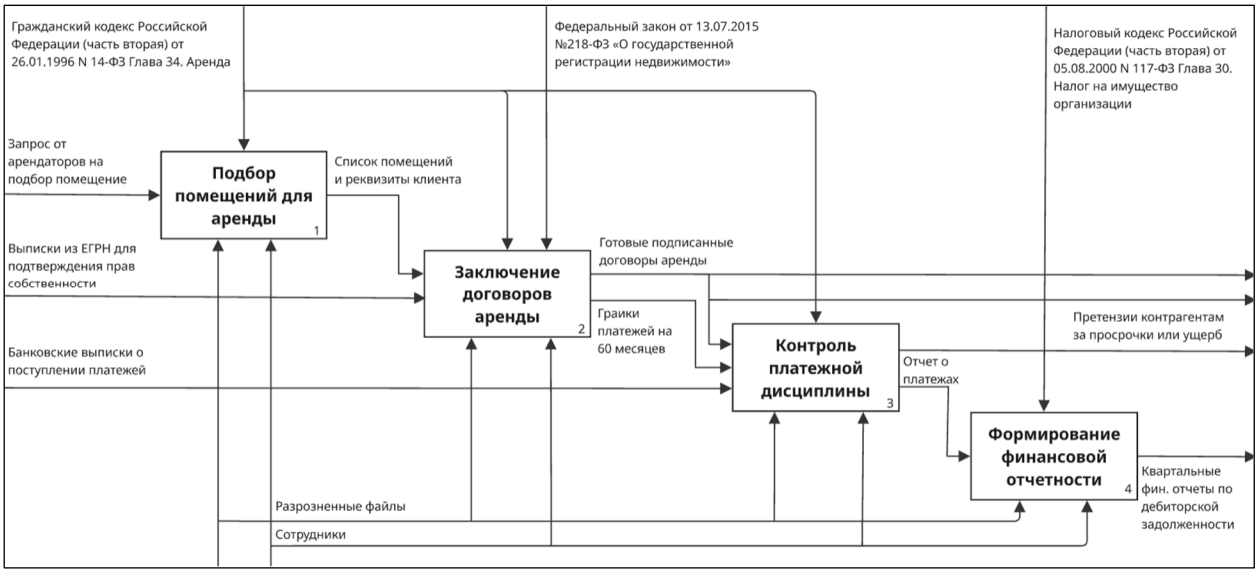

Рис. 1. Фрагмент процесса учета ресурсов индустриальной промышленной площадки в нотации IDEF0

При этом входами процесса являются:

– заявка на аренду;

– учредительные документы контрагента;

– техническое задание на параметры требуемого помещения.

Процесс регламентируется следующими нормативными правовыми актами и нормативными документами, определяющими порядок управления имуществом в интересах государства:

– Федеральный закон от 31.12.2014 № 488-ФЗ «О промышленной политике в Российской Федерации»;

– Федеральный закон от 26.01.1996 № 14-ФЗ «Гражданский кодекс Российской Федерации»;

– постановление Правительства Российской Федерации от 01.11.2012 № 1119 «О требованиях к защите персональных данных»;

– Международный стандарт финансовой отчётности по аренде IFRS 16 «Аренда».

Механизмами, участвующими в обеспечении деятельности по управлению площадями, являются:

– менеджер по аренде;

– сотрудник юридического отдела;
– инженер по техническому аудиту;
– администратор базы данных.

Результатом процесса управления являются:

– заключенный договор аренды;
– акт приема-передачи площадей;
– график платежей;
– итоговая аналитическая отчетность по эффективности использования промышленного потенциала региона.

Основными выявленными в процессе анализа деятельности проблемами являются:

– возможность регистрации недостоверной и противоречивой информации в процессе заполнения технических характеристик объектов и расчете арендных ставок;
– большое количество монотонной ручной работы по подготовке документов и ведению реестров, что ограничивает время для принятия эффективных управленческих решений;
– отсутствие централизованного цифрового архива и единой базы данных.

Для решения выявленных проблем предложено внести изменения в организацию процесса управления промышленными активами, которые нашли свое отражение в модели TO-BE. При этом, важным элементом развития выступает интеллектуализация системы. Так, например, с целью исключения ошибок, связанных с человеческим фактором, данные предлагается автоматически сопоставлять с уже имеющимися в базе. Интеллектуальная составляющая АРМ обеспечивается за счет внедрения алгоритмов автоматической верификации параметров объектов. В случае возникновения коллизий, например, при попытке сдачи объекта, который уже арендован на предстоящий период, операторы будут получать соответствующие уведомления, генерируемые системой поддержки принятия решений.

С учетом предложенного решения порядок управления деятельностью будет следующим:

– потенциальный арендатор подает заявку на производственную площадь. При этом, информация об арендаторе и промышленной площадке заносится в электронную базу данных;
– вся информация об оказанных услугах, начислениях и платежах также вносится в электронный реестр. При проверке прав доступа и технического состояния персонал использует данные из АРМ;
– в конце отчетного периода информация консолидируется в итоговый баланс использования мощностей.

В процессе подготовки формального вида представленных нововведений также использована функциональная модель IDEF0, фрагмент которой отображен на рис. 2.

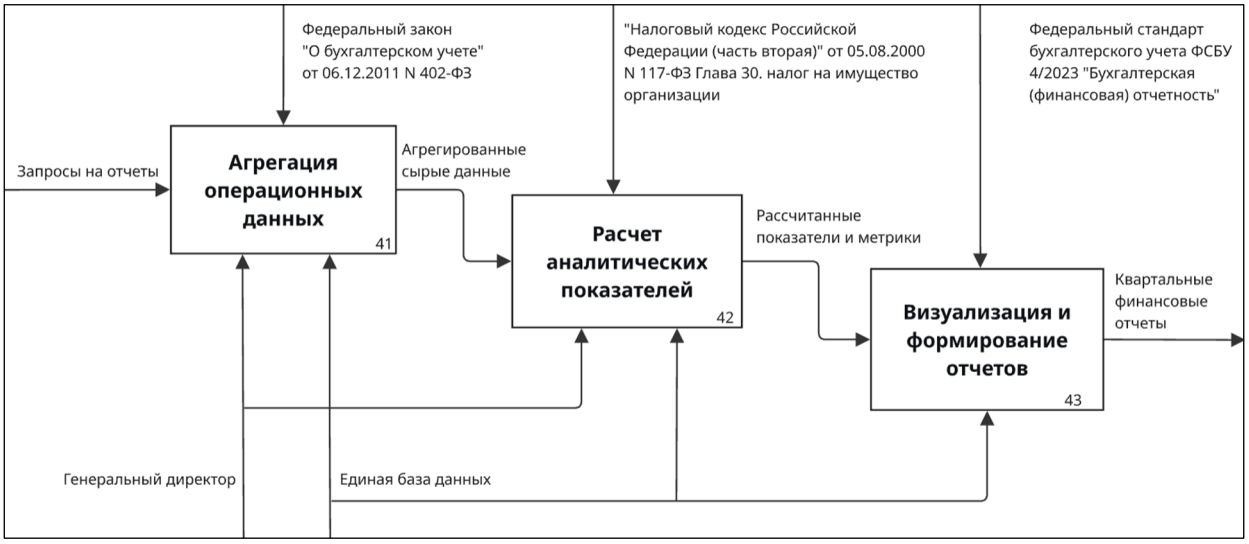

Рис. 2. Фрагмент диаграммы предложенного процесса учетной деятельности

Согласно диаграмме, сначала происходит автоматизированный прием и верификация заявки. На вход поступают технические требования, сведения о предприятии и онлайн-анкета. Интеллектуализация на данном этапе позволяет системе проводить предварительный скоринг заявки на соответствие доступным мощностям фонда. Ресурсами, необходимыми для выполнения указанных функций, являются специалист по управлению активами, а также специалист по информационной безопасности Результатом процесса являются платежные поручения и договор с электронным паспортом сделки.

Проведенное обследование позволяет выявить следующие типовые запросы к системе:

– вывести общую сумму задолженности по всем арендаторам заданного индустриального парка;

– вывести список помещений, освобождающихся в ближайший месяц, с сортировкой по дате окончания договора;

– вывести технические характеристики и контактные данные ответственных лиц для заданного оператором производственного корпуса.

2. Разработка типовой модели информационных потоков и реляционных процессов

Логическая структура, отражающая взаимосвязи между объектами индустриальной площадки, разработана на основе анализа различного рода документации и проведенного интервью с персоналом. Проведенный анализ показывает, что в системе выделяются следующие ключевые объекты: Договор, Объект недвижимости, Арендатор, Тип объекта, Счёт, Статус оплаты. Связи между объектами, определяющие логику взаимодействия, представлены на рис. 3.

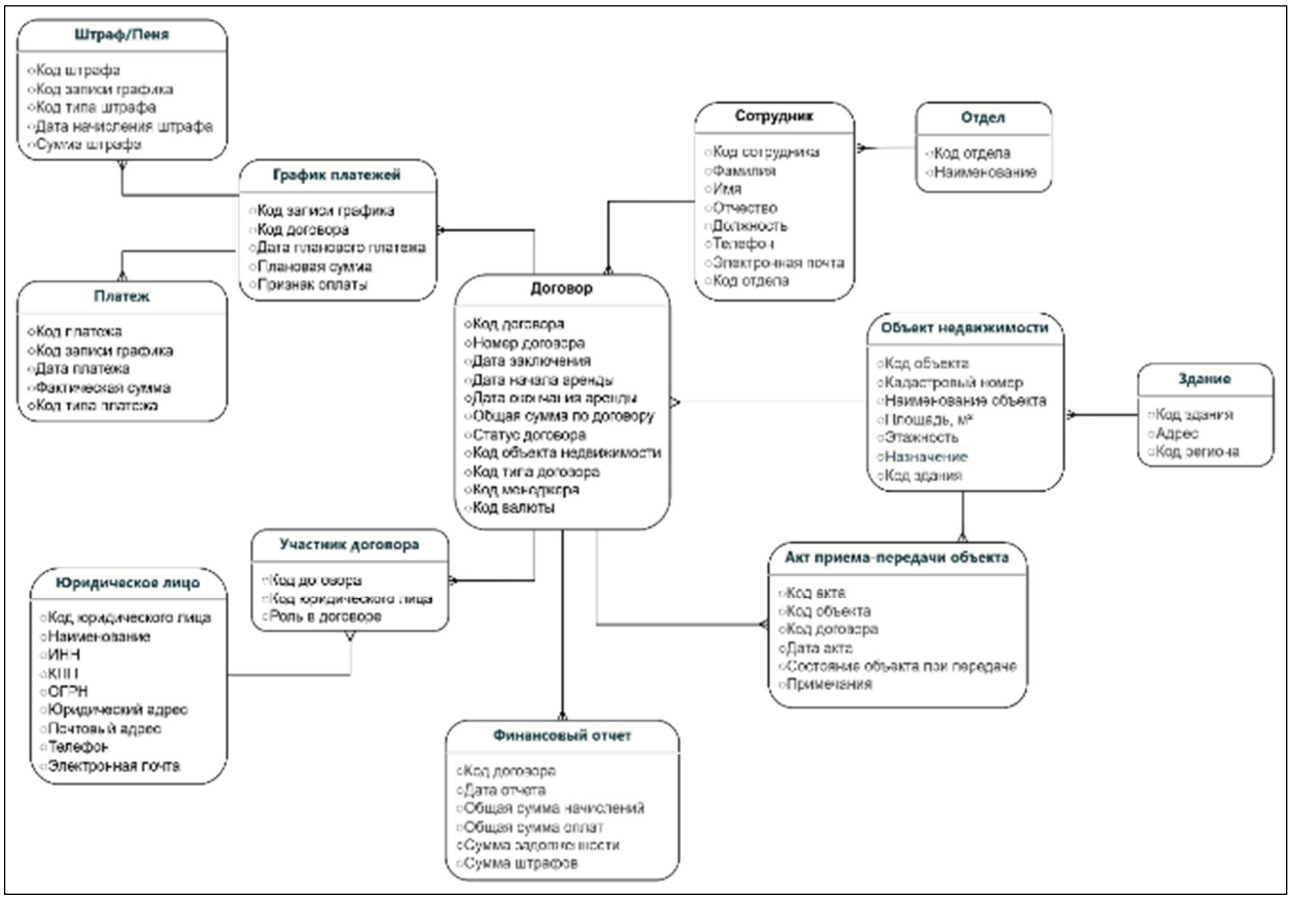

Рис. 3. Логическая модель данных, отражающая объекты промышленной площадки и взаимосвязи между ними

Логическая модель объектов промышленный площадки и взаимосвязей между ними позволила спроектировать физическую модель данных. При этом, учтена специфика организации данных, характерная для сервера баз данных PostgreSQL. Фрагмент такой модели, отражающий структуру фонда технических средств и контрактные обязательства, представлен на рис. 4.

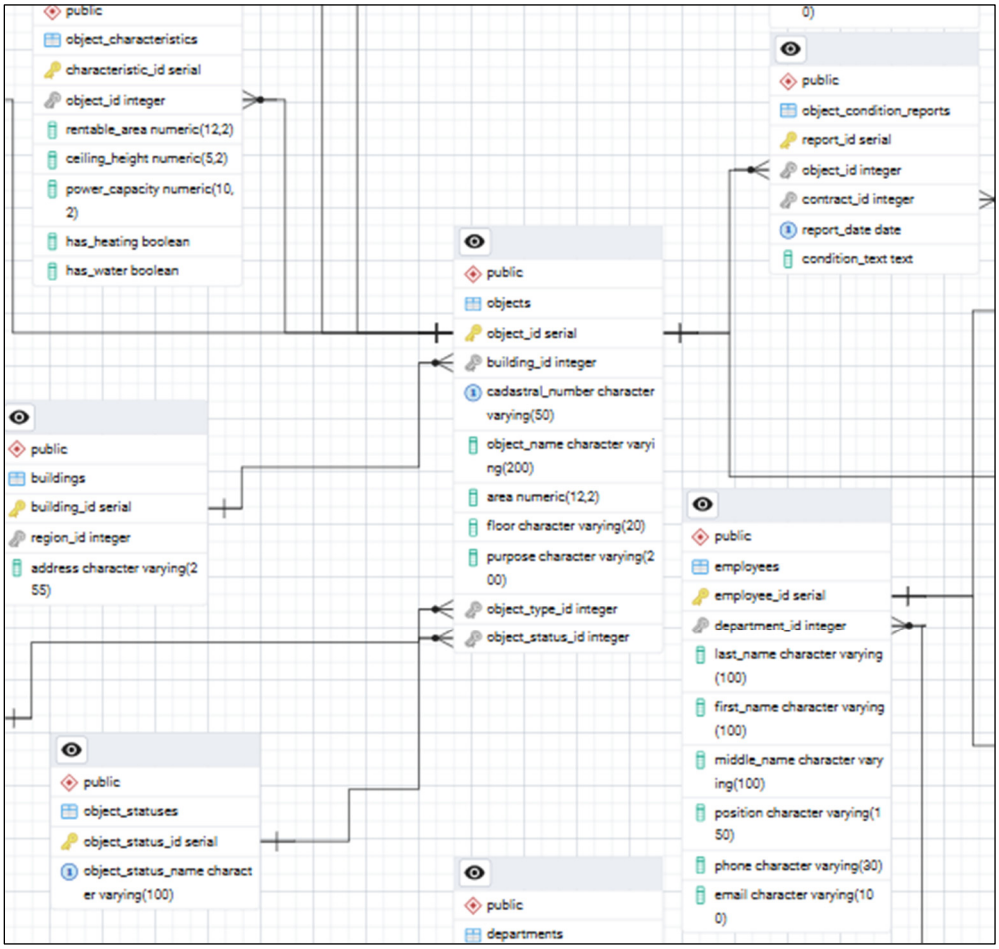

Рис. 4. Фрагмент физической модели данных, отражающий структуру фонда технических средств и контрактные обязательства

Структура таблиц предполагает использование специфических типов данных для точного учета характеристик площадей и финансовых транзакций. В качестве примера, опишем структуру двух наиболее значимых таблиц, обеспечивающих учет фондов и контроль поступлений – Objects и Contracts, см. таблицы 1 и 2 соответственно.

Для обеспечения интеллектуализации АРМ на уровне архитектуры БД в таблицы внедрена система альтернативных ключей. Связи атрибутов в рамках альтернативных ключей позволяют реализовать автоматический контроль бизнес-логики, предотвращая ввод некорректных данных еще на этапе записи в репозиторий.

Таблица 1

Структура таблицы Objects

Название поля	Тип данных	Пояснение	Ключи
Object_id	Int	Первичный ключ	Primary Key
Building_id	Int	Здание	АК2.1
Cadastral_number	Varchar	Кадастровый номер	АК1.1
Object_name	Varchar	Объект (площадь)	АК2.2
Area	Numeric	Площадь, м^2	
Floor	Varchar	Этаж	АК2.3
Purpose	Varchar	Назначение	
Object_type_id	Int	Тип объекта	Foreign Key
Object_status_id	Int	Статус объекта	Foreign Key

Использование композитного альтернативного ключа АК2 исключает риск дублирования помещений в рамках одного строения. Это обеспечивает уникальность каждой записи и позволяет системе проверять данные по площадям автоматически, еще до их попадания в основную программу.

Таблица 2

Структура таблицы Contracts

Название поля	Тип данных	Пояснение	Ключи
Contract_id	Int	Первичный ключ	Primary Key
Contract_number	Varchar	Номер договора	AK1.1
Contract_date	Date	Дата заключения	AK1.2
Date_from	Date	Дата начала действия	AK2.1
Date_to	Date	Дата окончания действия	
Status	Varchar	Статус договора	
Total_amount	Numeric	Общая сумма	
Object_id	Int	Объект (площадь)	Foreign Key, AK2.2
Contract_type_id	Int	Вид договора	Foreign Key
Manager_id	Int	Менеджер	Foreign Key
Currency_id	Int	Валюта	Foreign Key

Таблица связывает финансовые параметры с конкретными объектами через систему внешних ключей. Использование альтернативного ключа предотвращает наложение сроков аренды, не позволяя закрепить одну площадь за разными договорами в один период. Это автоматизирует контроль графиков и исключает ошибки при расчете платежей.

3. Интеграция функциональной и реляционной модели управленческих процессов

Объединим достоинства функциональной и реляционной модели, обеспечивая более точное и всестороннее представление информационных потоков и производственных процессов индустриальных промышленных площадок. При этом, особенности интеграции целесообразно формализовать посредством языка SQL [6], который позволяет эффективно описать как модель, представленную на рис. 4, так и запросы, выявленные в ходе анализа диаграммы, представленной на рис. 2.

Хочется отметить, что в процессе интеграции использованы методы искусственного интеллекта. Так, например, заранее предопределенные слоты SQL-фреймов, полученных по результатам анализа функциональной диаграммы, заполняются данными, полученными в ходе анализа реляционной модели [7]. Код формирования таблиц представлен в листинге 1, а запросы к ним – в листинге 2.

Листинг 1

Фрагмент программного кода формирования таблиц

```
CREATE TABLE objects (
object_id SERIAL PRIMARY KEY,
building_id INT NOT NULL,
object_name VARCHAR(255) NOT NULL,
area DECIMAL(10,2) NOT NULL,
cadastral_number VARCHAR(50) UNIQUE,
id_type INT REFERENCES object_types(id_type),
id_status INT REFERENCES object_statuses(id_status),
CONSTRAINT area_positive CHECK (area > 0)
CONSTRAINT ak_objects UNIQUE (building_id,
  object_name, floor)
);

CREATE TABLE contracts (
contract_id SERIAL PRIMARY KEY,
contract_number VARCHAR(100) NOT NULL
  UNIQUE,
contract_date DATE NOT NULL,
date_from DATE NOT NULL,
date_to DATE NOT NULL,
total_amount DECIMAL(15,2),
object_id INT REFERENCES objects(object_id),
manager_id INT REFERENCES employees(employee_id)
CONSTRAINT ak_contract_num UNIQUE
```

```
  (contract_number, contract_date),
CONSTRAINT ak_contract_date UNIQUE
  (object_id, date_from)
);
```

Созданный, при этом, триггер совместно с функцией автоматически блокирует попытки двойного бронирования одной и той же площади на пересекающиеся периоды. Это позволяет перенести проверку целостности данных с уровня прикладного ПО непосредственно на уровень базы данных, минимизируя риск ошибки оператора при вводе.

Листинг 2

Фрагмент программного кода запросов, представленных в параграфе 1

```
1. Формирование перечня объектов с техническими характеристиками (мощность, отопление, тип)
SELECT o.object_name, ot.object_type_name,
  oc.power_capacity, oc.has_heating,
  b.address
FROM objects o
JOIN buildings b ON b.building_id =
  o.building_id
JOIN object_types ot ON ot.object_type_id =
  o.object_type_id
LEFT JOIN object_characteristics oc ON
  oc.object_id = o.object_id
ORDER BY b.address, o.object_name;

2. Сравнение плановых и фактических поступлений платежей по объектам недвижимости
SELECT o.object_name, SUM(ps.planned_amount)
  AS sum_planned, COALESCE(SUM(p.amount), 0)
  AS sum_paid
FROM objects o
JOIN contracts c ON c.object_id = o.object_id
JOIN payment_schedules ps ON ps.contract_id =
  c.contract_id
LEFT JOIN payments p ON p.schedule_id =
  ps.schedule_id
GROUP BY o.object_name
ORDER BY o.object_name;

3. Сводный отчет по портфелю объектов, площадям и платежам в разрезе регионов
SELECT r.region_name,
  COUNT(DISTINCT o.object_id) AS
  objects_count, SUM(o.area) AS total_area,
  SUM(fr.total_payment) AS total_payments
FROM regions r
JOIN buildings b ON b.region_id = r.region_id
JOIN objects o ON o.building_id =
  b.building_id
JOIN contracts c ON c.object_id = o.object_id
JOIN financial_reports fr ON fr.contract_id =
  c.contract_id
GROUP BY r.region_name
ORDER BY total_payments DESC;
```

Приведенные запросы обеспечивают прозрачный контроль за финансовыми потоками и эффективностью использования имущественного фонда в различных разрезах. Таким образом, программная реализация модели предоставляет инструментарий как для оперативного учета и долгосрочного стратегического планирования деятельности предприятия.

Заключение

Результаты проведенного научного исследования могут быть сформулированы следующим образом:

– проанализирован порядок деятельности по управлению промышленными площадями в регионе, а также нормативно-правовая база, регламентирующая учет арендных операций. Выявлены критические точки в процессах мониторинга фондов и предложен вариант комплексного АРМ для государственных управляющих структур;

– обоснован выбор СУБД и инструментария. Спроектирована логическая и физическая структура данных, отражающая специфику промышленной недвижимости региона. Внедрена система альтернативных ключей, обеспечивающая первичный уровень автоматического контроля целостности данных;

– реализована концепция интеллектуализации управления активами посредством интеграции функциональной и реляционной моделей. В программный код на языке SQL внедрены алгоритмы активного мониторинга (триггеры), обеспечивающие автоматическое выявление логических коллизий и исключение дублирования арендных периодов;

– разработано программное обеспечение, обеспечивающее интеграцию информационной базы и автоматизированных запросов для формирования отчетности по ключевым финансовым и техническим метрикам.

За счет комплексного подхода к автоматизации [8] удалось сократить сроки простоя индустриальной промышленной площадки, оптимизировать распределение производственных площадей под нужды импортозамещения. Централизованный подход к хранению информации позволил обеспечить выполнение требований законодательства Российской Федерации к защите конфиденциальной информации и персональных данных.

Литература

1. *Мамателашвили О.В., Байбурин Р.Р.* Роль цифрового суверенитета в обеспечении экономической безопасности России в условиях геополитической нестабильности // Экономика. Право. Инновации. 2025. №4. С. 30-40.

2. *Ковтун И.И.* Теория и практика проектирования государственных информационных систем. Учебное пособие. СПб.: НИЦ АРТ, 2023. 194 с.

3. *Ковтун И.И.* Проблемы моделирования проектных решений в процессе проектирования автоматизированных информационных систем // Информатизация и связь. 2012. № 8. С. 145-151.

4. *Полищук Ю.В.* Базы данных и их безопасность. Учебное пособие. М.: ИНФРА-М, 2025. 210 с.

5. *Ковтун И.И.* Функционально-реляционный анализ вариантов автоматизации сложных организационно-технических систем. Монография. СПб.: НИЦ АРТ, 2024. 214 с.

6. *Ковтун И.И., Романенко Г.С.* Компьютерная реализация методов формального анализа сложных организационно-технических систем в процессе проведения комплексной автоматизации // Информатизация и связь. 2022. № 7. С. 38-43.

7. *Воронова Л.И., Воронов В.И.* Big data. Методы и средства анализа. Учебное пособие. М.: МТУСИ, 2016. 33 с.

8. *Ковтун И.И., Романенко Г.С.* Математическое моделирование, численные методы и комплексы программ в задачах повышения эффективности многономенклатурных производств // Информатизация и связь. 2020. № 1. С. 48-54.

РАЗРАБОТКА ПРОТОТИПА ЗАЩИЩЕННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ЗНАНИЯМИ НА ОСНОВЕ ТЕХНОЛОГИИ RAG

Парамонова Анна Александровна
Московский технический университет связи и информатики, студент, Москва, Россия

Раковский Дмитрий Игоревич
Московский Технический Университет Связи и Информатики,
к.т.н., доцент кафедры «Информационная безопасность», Москва, Россия
a.a.paramonova@edu.mtuci.ru

Аннотация
Данное исследование посвящено разработке защищённой системы управления знаниями на основе технологии Retrieval-Augmented Generation (RAG). Рассматриваются архитектурные принципы построения RAG-систем, особенности применения больших языковых моделей и связанные с ними риски информационной безопасности. Предлагается архитектура системы, обеспечивающая семантический поиск, генерацию ответов на естественном языке и локальное развертывание с соблюдением требований защиты информации.

Ключевые слова
системы управления знаниями, Retrieval-Augmented Generation, RAG-системы, большие языковые модели, информационная безопасность, семантический поиск, контроль доступа, локальное развертывание, защита информации.

Введение

В условиях цифровой трансформации объем корпоративной информации непрерывно растет, что делает задачу эффективного управления знаниями критически важной для организаций. Традиционные системы управления знаниями, основанные на полнотекстовом поиске и ключевых словах, демонстрируют ограниченную эффективность при работе с неструктурированными данными и не обеспечивают контекстно-зависимый поиск информации.

Развитие больших языковых моделей и появление подхода Retrieval-Augmented Generation (RAG) позволили объединить семантический поиск и генерацию ответов на естественном языке, что качественно изменило способы взаимодействия пользователей с корпоративными базами знаний. Вместе с тем использование RAG в корпоративной среде сопровождается существенными рисками информационной безопасности, связанными с утечкой конфиденциальных данных, некорректным разграничением доступа и уязвимостями, специфичными для LLM.

Особую актуальность данная проблема приобретает в условиях требований российского законодательства в области защиты информации и локализации данных. В связи с этим возникает необходимость разработки защищённой системы управления знаниями на основе RAG, обеспечивающей семантический поиск и генерацию ответов при соблюдении требований информационной безопасности и возможности локального развертывания.

Целью работы является разработка прототипа защищенной системы управления знаниями на основе технологии RAG, позволяющей взаимодействовать с информацией, циркулирующей в закрытом контуре корпорации.

Актуальность применения систем управления знаниями в корпоративных средах

Системы управления знаниями (Knowledge Management Systems, KMS) представляют собой совокупность организационных, методических и программно-технических средств, предназначенных для накопления, хранения, структурирования, поиска и распространения знаний внутри организации. В отличие от традиционных информационных систем, ориентированных преимущественно на работу со структурированными данными, системы управления знаниями направлены на обработку как формализованных, так и слабо структурированных информационных ресурсов.

Концепция управления знаниями сформировалась в конце XX века и получила развитие в работах И. Нонаки и Х. Такеути, которые предложили разделять знания на явные и неявные [1]. Явные знания поддаются формализации и могут быть представлены в виде документов, инструкций и баз данных,

тогда как неявные знания связаны с практическим опытом, интуицией и профессиональными навыками сотрудников.

Современные системы управления знаниями классифицируются по функциональному назначению (корпоративные порталы знаний, системы электронного документооборота, экспертные системы), по технологической основе (реляционные базы данных, полнотекстовый поиск, интеллектуальные системы с машинным обучением) и по масштабу внедрения (персональные, групповые, корпоративные и межорганизационные решения).

Большие языковые модели (Large Language Models, LLM) представляют собой нейронные сети с архитектурой трансформер, обученные на огромных объемах текстовых данных. Эти модели могут понимать и генерировать текст, выполнять рассуждения, отвечать на вопросы и решать задачи обработки естественного языка [2].

Архитектура трансформера обеспечивает масштабируемость моделей и возможность обучения на больших объемах текстовых данных. В зависимости от архитектурной схемы модели подразделяются на модели с энкодером, модели с декодером и комбинированные решения типа encoder-decoder. В задачах генерации текста и диалоговых систем наибольшее распространение получили модели типа decoder-only, поскольку они позволяют последовательно формировать ответ, опираясь на предыдущий контекст.

Ключевыми характеристиками LLM являются количество параметров, размер контекстного окна, поддерживаемые языки и способность к обобщению. Увеличение числа параметров, измеряемое миллиардами, как правило, повышает качество генерации, однако требует значительных вычислительных ресурсов [3].

Одной из фундаментальных проблем LLM является склонность к генерации недостоверной информации, так называемых галлюцинаций. Данная особенность связана с вероятностной природой языковых моделей, которые последовательно предсказывают следующий элемент текста, основываясь на статистических закономерностях, а не на проверке фактической достоверности информации. В корпоративных системах управления знаниями это ограничение требует дополнительных механизмов контроля и верификации [4].

Технология Retrieval-Augmented Generation (RAG) представляет собой подход к построению интеллектуальных систем обработки естественного языка, сочетающий методы информационного поиска и генеративные возможности больших языковых моделей. Основная идея RAG заключается в том, что языковая модель при формировании ответа опирается не только на знания, заложенные в процессе обучения, но и на актуальную информацию, извлекаемую из внешнего источника знаний.

Классическая архитектура RAG включает несколько логически связанных компонентов. На первом этапе осуществляется подготовка базы знаний, которая предполагает загрузку документов из различных источников, извлечение текстового содержимого, его разбиение на фрагменты и построение векторных представлений. Векторизация выполняется с использованием моделей эмбеддингов, которые преобразуют текстовые фрагменты в числовые векторы, отражающие их семантическое содержание.

Следующим этапом является поиск релевантной информации. Пользовательский запрос преобразуется в векторное представление, после чего выполняется поиск наиболее близких векторов в базе знаний. Для этого используются методы приближенного поиска ближайших соседей, позволяющие эффективно работать с большими объемами данных. Результатом данного этапа является набор текстовых фрагментов, наиболее соответствующих смыслу запроса.

На этапе генерации ответа отобранные фрагменты включаются в контекст языковой модели. Модель формирует связный текстовый ответ, опираясь на предоставленный контекст и заданные правила генерации. Такой подход позволяет снизить вероятность генерации недостоверной информации и повысить прозрачность результатов.

Важным аспектом RAG является стратегия разбиения документов. Слишком крупные фрагменты могут привести к потере релевантности, тогда как чрезмерно мелкое разбиение увеличивает объем базы знаний и усложняет поиск. На практике применяются различные стратегии, включая разбиение по абзацам, предложениям или смысловым блокам.

Современные коммерческие RAG-решения, такие как Microsoft Copilot for Microsoft 365, Google Vertex AI Search и Amazon Kendra, обеспечивают высокую масштабируемость и развитую инфраструктуру, однако предполагают обработку данных на серверах провайдера. Это создает риски утечки конфиденциальной информации и формирует зависимость от внешнего поставщика услуг. Решения с открытым исходным кодом, такие как LangChain, LlamaIndex и Haystack, предоставляют гибкую технологическую основу, но, как правило, не включают встроенных механизмов комплексной защиты информации, что требует дополнительной разработки систем аутентификации, авторизации и аудита.

В RAG-системах ключевой компонент извлечения информации реализуется через семантический поиск: текстовые фрагменты документов преобразуются в векторные представления (эмбеддинги), после чего выполняется поиск ближайших векторов к вектору запроса. Для хранения и поиска по эмбеддингам применяются специализированные средства векторного поиска и векторные базы данных, оптимизированные под операции поиска ближайших соседей в многомерном пространстве.

Выбор векторного хранилища в составе RAG-системы определяется рядом факторов: требуемой производительностью, поддержкой фильтрации по метаданным (что важно для реализации разграничения доступа), удобством локального развертывания и возможностью интеграции с используемым стеком разработки. В корпоративных системах управления знаниями особенно важны поддержка хранения метаданных и возможность применения политик доступа на этапе извлечения контекста до передачи данных языковой модели.

Актуальные проблемы обеспечения информационной безопасности в системах управления знаниями

Системы управления знаниями, построенные на основе технологии RAG и больших языковых моделей, представляют собой сложные информационные системы, сталкивающиеся с широким спектром угроз безопасности. Согласно классической модели конфиденциальности, целостности и доступности, угрозы классифицируются по трем основным направлениям [5].

Угрозы конфиденциальности включают несанкционированный доступ к защищаемым документам, утечку информации через ответы системы, перехват данных при передаче, получение доступа к векторным представлениям документов и анализ логов для извлечения конфиденциальной информации. Угрозы целостности связаны с модификацией документов в базе знаний, «отравлением» векторной базы данных вредоносными данными и подменой ответов языковой модели [6].

Специфической для LLM-систем угрозой является внедрение (инъекция) запросов, когда злоумышленник манипулирует входными данными для изменения поведения модели. Прямая инъекция выглядит как попытка переопределить инструкции системы через запрос пользователя. Косвенная инъекция через документы более опасна, когда в документ внедряются скрытые инструкции, которые выполняются при включении фрагмента в контекст [7].

Разработка защищенной системы управления знаниями должна осуществляться в строгом соответствии с требованиями российского законодательства. Федеральный закон № 152-ФЗ «О персональных данных» устанавливает принципы обработки персональных данных и требует обеспечения точности, достаточности и актуальности данных [8]. ГОСТ Р ИСО/МЭК 27001-2021 определяет требования к системе менеджмента информационной безопасности [9]. Приказы ФСТЭК России устанавливают базовый и дополнительный наборы мер для систем разных уровней защищенности [10-13].

Локальное развертывание языковых моделей и RAG-систем обеспечивает полный контроль над обрабатываемыми данными и исключает риски, связанные с передачей конфиденциальной информации внешним провайдерам. В отличие от облачных решений, локальные системы позволяют организациям самостоятельно определять политики безопасности, контролировать доступ к данным и обеспечивать соответствие требованиям локализации информации согласно российскому законодательству. Кроме того, использование локальных моделей снижает зависимость от доступности внешних сервисов и устраняет риски изменения условий обслуживания или прекращения поддержки API со стороны поставщика.

Структура разрабатываемого прототипа

Проектируемая система представляет собой защищённую корпоративную платформу управления знаниями, обеспечивающую семантический поиск и генерацию ответов на естественном языке с учетом требований информационной безопасности. Проектирование выполняется на принципах Security by Design и Zero Trust. Система реализуется в виде многоуровневой архитектуры (табл. 1).

Таблица 1

Уровень	Компоненты и функции
Пользовательский	Веб-интерфейс для ввода запросов и просмотра ответов
Доступа и маршрутизации	API-шлюз: аутентификация, валидация, маршрутизация
Управления доступом	Ролевая и атрибутная модели, контроль прав
Обработки документов	Загрузка, извлечение текста, разбиение, векторизация
Семантического поиска	Векторный поиск с фильтрацией по метаданным доступа
Выполнения LLM	Локальная языковая модель для генерации ответов
Аудита и мониторинга	Регистрация событий безопасности и действий пользователей

Ключевыми архитектурными решениями являются: фильтрация результатов семантического поиска по правам доступа до формирования контекста LLM; локальное выполнение языковой модели; разделение контуров индексации документов и пользовательских запросов; централизованное журналирование событий безопасности.

В рамках работы реализован прототип системы. Подсистема формирования базы знаний обеспечивает автоматическую загрузку документов, их разбиение на фрагменты и преобразование в векторные представления с использованием модели sentence-transformers. Полученные эмбеддинги сохраняются в векторной базе данных ChromaDB, что обеспечивает эффективный семантический поиск (рис. 1).

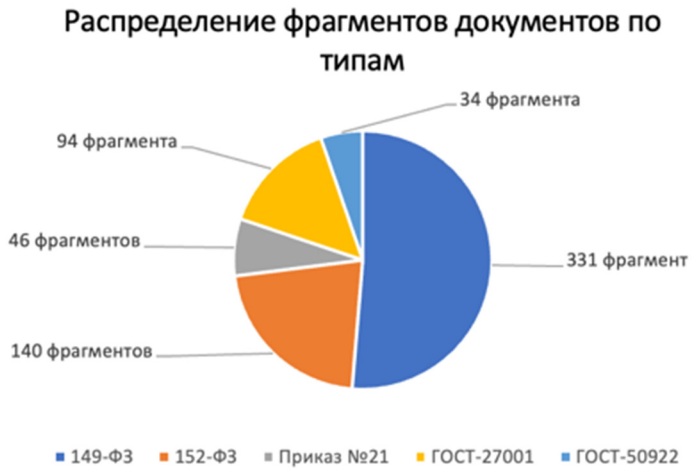

Рис. 1. Результаты индексации нормативных документов

Модуль RAG объединяет семантический поиск и генерацию текста языковой моделью. При поступлении запроса система выполняет семантический поиск релевантных фрагментов, формирует контекст и генерирует ответ с указанием источников информации (рис. 2).

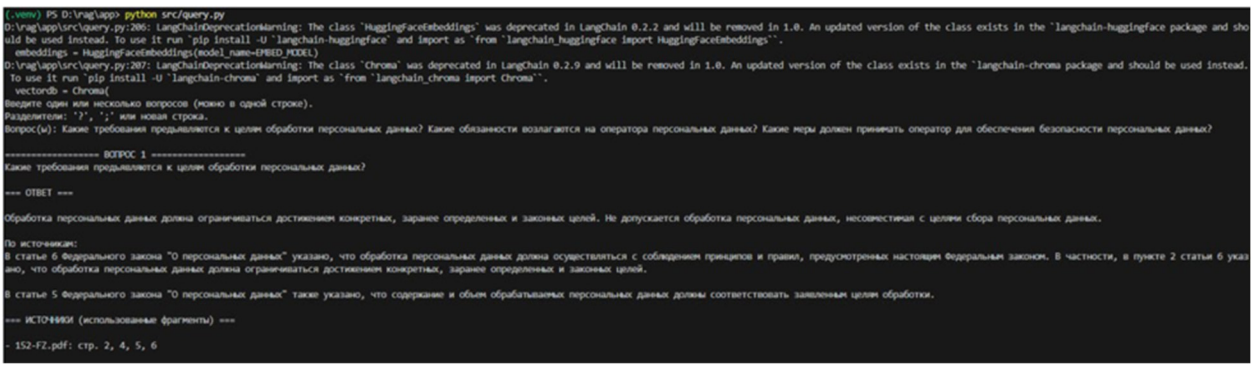

Рис. 2. Пример обработки пользовательского запроса

Подсистема аутентификации и авторизации обеспечивает контролируемый доступ к системе. Добавление учетных записей осуществляется администратором через интерфейс командной строки (рис. 3а). Пароли хранятся в виде криптографических хэшей с использованием алгоритма bcrypt (рис. 3б).

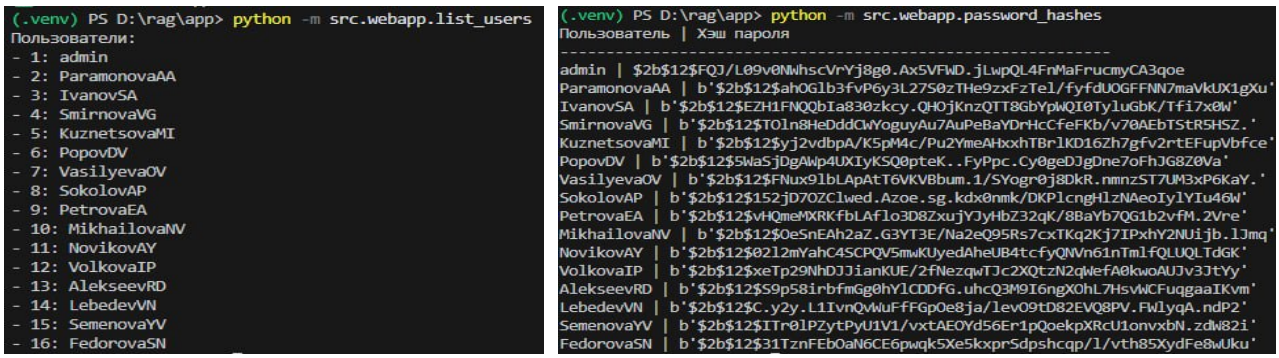

Рис. 3. Реализованный функционал прототипа: а) Просмотр списка пользователей администратором; б) Содержимое базы данных с хэшированными паролями

Аутентификация осуществляется через веб-интерфейс (рис. 4). При успешной аутентификации пользователь получает доступ к функциональности системы (рис. 5). При неуспешной попытке доступ не предоставляется, событие фиксируется в журнале аудита.

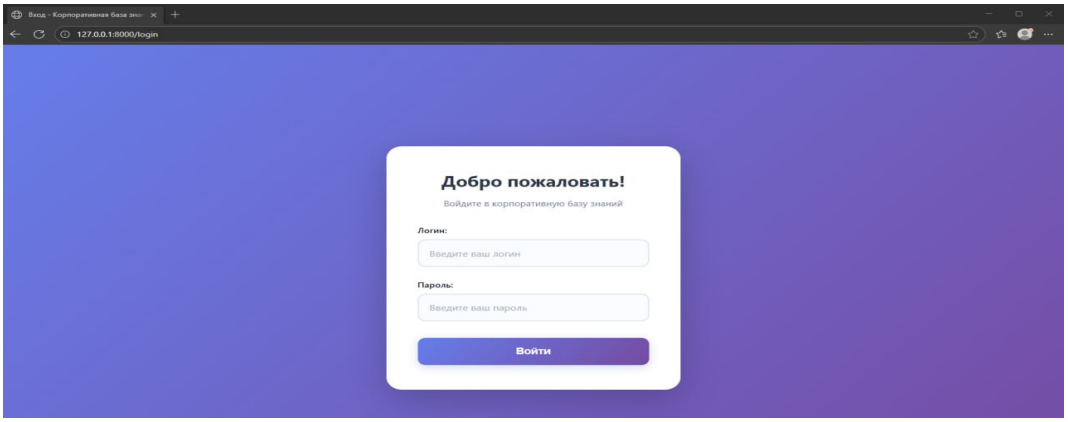

Рис. 4. Форма аутентификации пользователя и ввод учетных данных пользователем

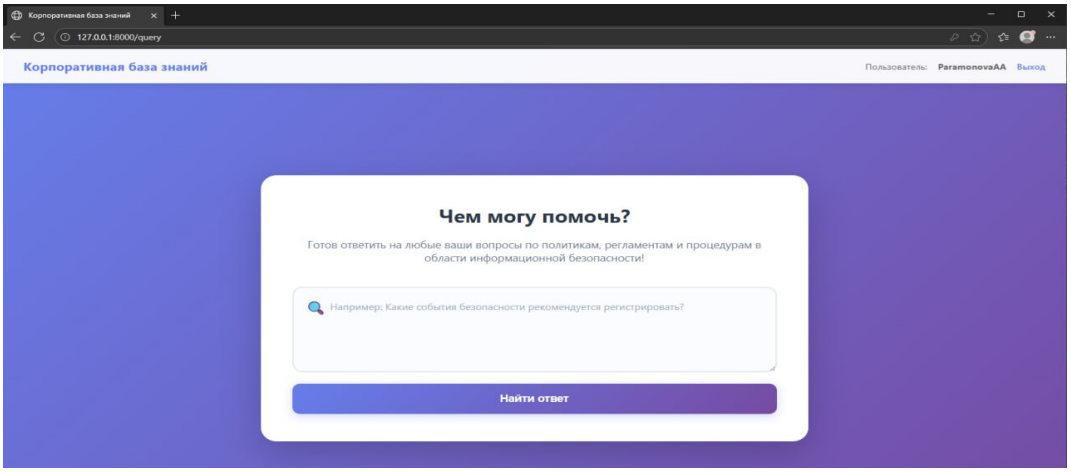

Рис. 5. Интерфейс системы после успешной аутентификации

Веб-интерфейс обеспечивает работу с базой знаний. Система поддерживает обработку одиночных запросов (рис. 6) и множественных запросов в рамках одного обращения (рис. 7). Для каждого ответа указываются документы-источники и номера страниц, что обеспечивает прозрачность и возможность верификации результатов.

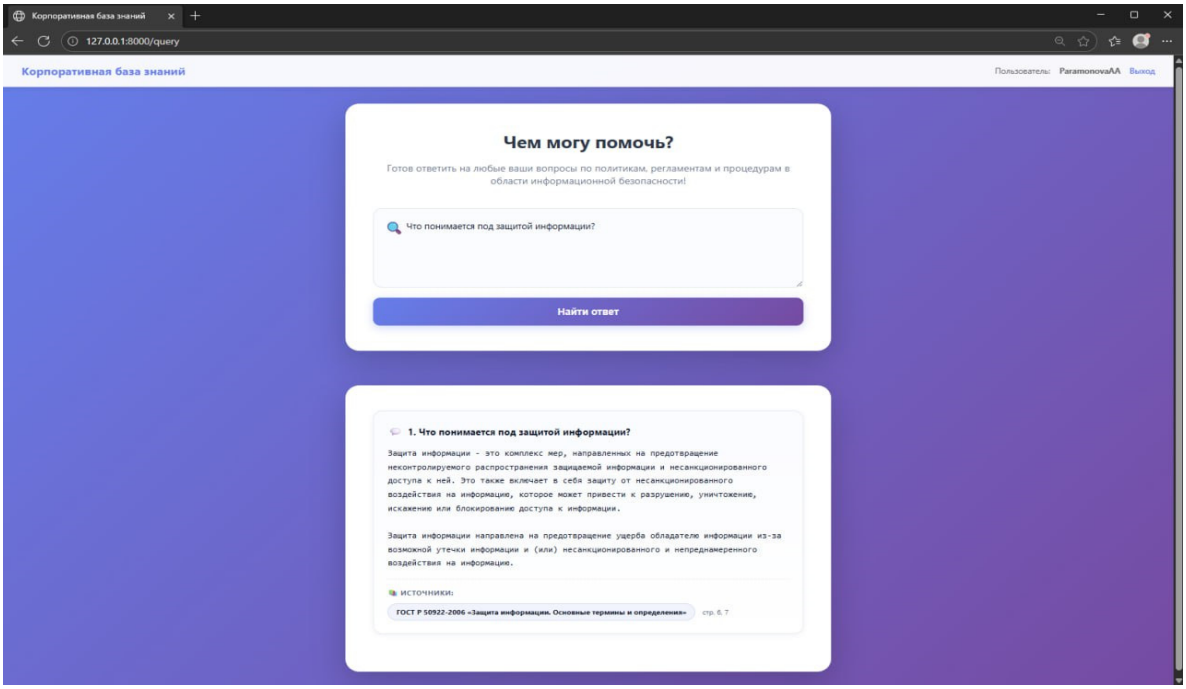

Рис. 6. Отображение ответа с указанием документа-источника

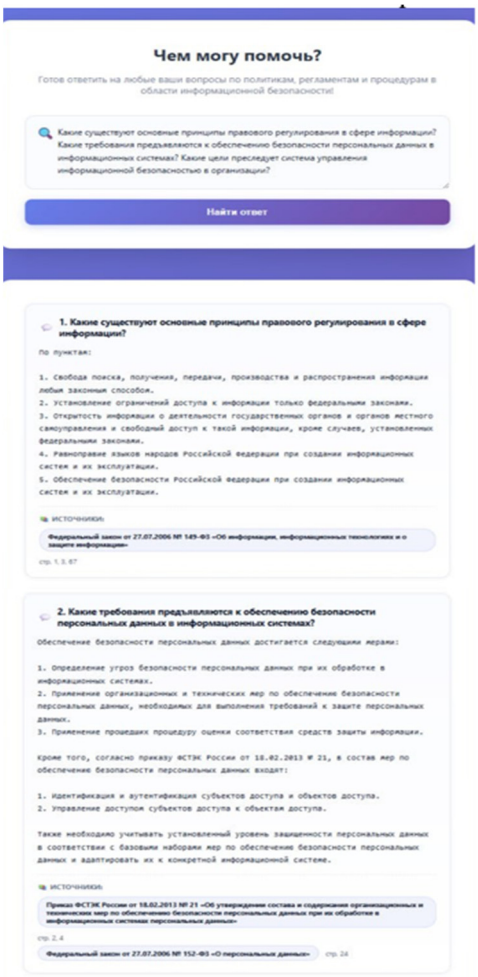

Рис. 7. Отображение ответа на три вопроса с указанием документа-источника

Подсистема регистрации и аудита событий безопасности обеспечивает прослеживаемость действий пользователей и администраторов. В журнале фиксируются события аутентификации, загрузки документов, выполнения запросов и попытки нарушения политики безопасности. Журналы защищены от несанкционированного изменения и используются для анализа инцидентов.

Заключение

В рамках исследования была разработана и реализована защищённая система управления знаниями на основе технологии Retrieval-Augmented Generation, ориентированная на локальное развертывание и работу с корпоративными документами, содержащими чувствительную информацию.

Проведён комплексный анализ современных систем управления знаниями, архитектурных особенностей RAG-подхода и рисков информационной безопасности, связанных с использованием больших языковых моделей. На основе выявленных ограничений существующих решений была спроектирована архитектура системы, обеспечивающая семантический поиск, генерацию ответов на естественном языке и многоуровневый контроль доступа.

Реализованный прототип демонстрирует практическую применимость предложенных архитектурных решений, включая фильтрацию данных до передачи контекста языковой модели, локальное выполнение LLM и централизованный аудит событий безопасности. Система обеспечивает работу с нормативно-правовыми документами в области информационной безопасности, автоматическую индексацию, семантический поиск и генерацию ответов с указанием источников.

Результаты тестирования подтверждают корректность функционирования основных компонентов системы и соответствие реализованных механизмов требованиям информационной безопасности. Разработанный прототип может быть использован в качестве основы для дальнейшего развития корпоративных систем управления знаниями и расширения функциональности с учётом специфики предметной области и требований организации.

Литература

1. *Нонака И., Такеути Х.* Компания – создатель знаний: зарождение и развитие инноваций в японских фирмах : пер. с англ. М. : Олимп-Бизнес, 2011. 384 с.

2. *Рассел С., Норвиг П.* Искусственный интеллект: современный подход : пер. с англ. 3-е изд. М.: Вильямс, 2020. 1136 с.

3. *Васильев В. Н., Киселев А. А.* Информационная безопасность информационных систем : учеб. пособие. М.: Академия, 2019. 336 с.

4. *Парамонова А. А., Раковский Д. И.* Анализ методов обнаружения спам-сообщений с применением больших языковых моделей // Гибкое производство, цифровая трансформация, информационная безопасность и экология : Сборник трудов по материалам I Всероссийской научно-практической конференции с международным участием, Москва, 28-29 октября 2025 года. Курск: Московский технический университет связи и информатики, ЗАО "Университетская книга", 2025. С. 227-235. EDN GFEBWA.

5. *Золотарев В. В., Трофимычев И. И.* Модель оценки компрометации системы с использованием элементов управления знаниями // Инфокоммуникационные технологии. 2025. Т. 23, № 1(89). С. 121-127. DOI 10.18469/ikt.2025.23.1.12. EDN OEYRBW.

6. *Аршинский Л. В., Жукова М. С., Шурховецкий Г. Н.* Проблемы информационной безопасности в системах искусственного интеллекта, использующих модель предметной области // Информационные технологии и математическое моделирование в управлении сложными системами. 2024. № 1(21). С. 36-44. EDN EPCTPL.

7. *Иваничкин В. В., Янченко Е. Н.* Актуальность построения модели угроз безопасности информационных систем // Современные тенденции развития аграрной науки : Сборник научных трудов III международной научно-практической конференции, Брянск, 11-12 декабря 2024 года. Брянск: Брянский государственный аграрный университет, 2024. С. 505-507. EDN FMATCN.

8. ГОСТ Р ИСО/МЭК 27001-2021. Информационные технологии. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. Введ. 2022-01-01. М.: Стандартинформ, 2021.

9. *Соловьев И. В.* Защита информации в информационных системах. М.: БХВ-Петербург, 2021. 416 с.

10. *Гостев А. А.* Информационная безопасность корпоративных систем. М.: Горячая линия – Телеком, 2020. 384 с.

11. *Шелухин О. И., Раковский Д. И.* Редкие аномальные события. проблемы обнаружения и обработки // Наукоемкие технологии в космических исследованиях Земли. 2025. Т. 17, № 4. С. 54-68. DOI 10.36724/2409-5419-2025-17-4-54-68. EDN GDQHTU.

12. *Раковский Д. И.* Влияние проблемы многозначности меток классов системных журналов на защищенность компьютерных сетей // Наукоемкие технологии в космических исследованиях Земли. 2023. Т. 15, № 1. С. 48-56. DOI 10.36724/2409-5419-2023-15-1-48-56. EDN QVUZJT.

13. *Шелухин О. И., Раковский Д. И.* Выбор метрических атрибутов редких аномальных событий компьютерной системы методами интеллектуального анализа данных // T-Comm: Телекоммуникации и транспорт. 2021. Т. 15, № 6. С. 40-47. DOI 10.36724/2072-8735-2021-15-6-40-47. EDN YJDUYV.

ИССЛЕДОВАНИЕ ВЛИЯНИЯ ФИКСИРОВАННЫХ ТОЧЕК НА УСТОЙЧИВОСТЬ АЛГОРИТМА RSA

Трушин Алексей Андреевич
РТУ МИРЭА, студент, Москва, Россия
trushin.a.a@edu.mirea.ru

Аннотация
В данной статье исследуется проблема фиксированных точек в асимметричном криптографическом алгоритме RSA. Анализируется количество таких точек, факторы, влияющие на их число, и методы их нахождения. Особое внимание уделяется влиянию фиксированных точек на криптостойкость алгоритма, а также рассматриваются возможности использования стационарных фиксированных точек для компрометации криптосистемы RSA. Представленные результаты позволяют определить фиксированные точки при различных параметрах криптосистемы, что подчеркивает важность тщательного выбора параметров RSA для обеспечения безопасности.

Ключевые слова:
криптография, ассиметричное шифрование, RSA, фиксированные точки, криптостойкость.

Введение

Алгоритм шифрования RSA, разработанный Роном Ривестом, Ади Шамиром и Леонардом Адлеманом, является одним из наиболее широко используемых криптографических алгоритмов с открытым ключом. Его криптографическая стойкость базируется на вычислительной сложности задачи факторизации больших полупростых чисел. RSA находит широкое применение в создании и проверке цифровых подписей, для обеспечения безопасного обмена ключами и аутентификации. Стандартный размер ключа составляет 2048 бит.

Актуальность данной работы обусловлена повсеместным применением алгоритма RSA и необходимостью глубокого понимания его потенциальных уязвимостей.

Несмотря на то, что в ряде исследований было показано существование фиксированных точек, при которых зашифрованное сообщение остается неизменным, вопросы о методах их нахождения и влиянии на криптостойкость остаются недостаточно изученными.

Данная статья направлена на восполнение этого пробела, предоставляя точные данные для нахождения фиксированных точек при заданных параметрах криптосистемы. Полученные результаты могут быть полезны как для теоретического понимания принципов работы RSA, так и для практического применения при выборе безопасных параметров криптосистемы.

1. RSA-ключи

RSA-ключи применяются для создания электронной цифровой подписи, для аутентификации при установлении TSLсоединения [1].

Общая структура электронной подписи выглядит следующим образом (рис. 1):

1. Вычисляется хэш-значение сообщения (это может быть электронный документ, программное обеспечение или что-то другое).
2. Создается электронная подпись с использованием ассиметричного алгоритма и закрытого ключа.
3. Результат – это исходное сообщение и значение зашифрованного хэш-значения.

Важно, что для создания подписи используется закрытый ключ. Это значит, что подписать может только тот, кто имеет секретный ключ, а проверить любой, так как открытый ключ храниться публично.

На рисунке 3 приведен возможный список наборов алгоритмов. Из 21 набора, 15 содержат в своем списке RSA. И как видно из рисунка 4, сервер выбрал 4-ый набор, в котором есть наш алгоритм.

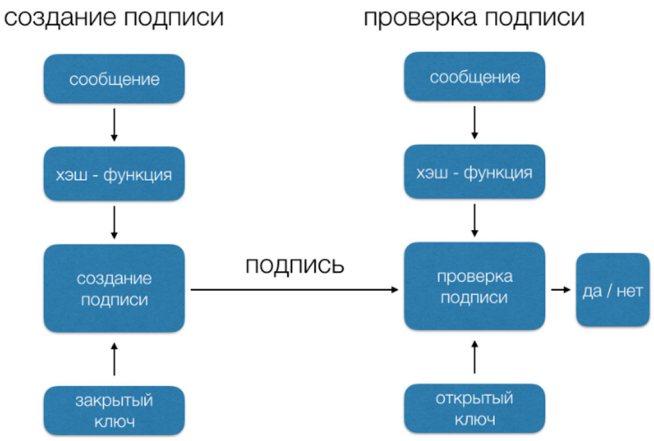

Рис. 1. Создание электронной подписи

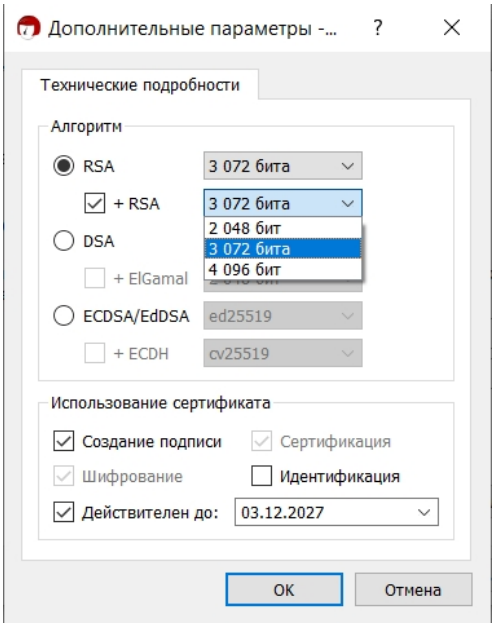

Рис.2. Создание самоподписанного сертификата в Kleopatra

В протоколе TLS RSA используется для взаимной аутентификации сервера и клиента, а также для безопасного установления сеансовых ключей.

```
Cipher Suites (21 suites)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (0x009f)
    Cipher Suite: TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 (0x009e)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 (0xc024)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 (0xc023)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 (0xc028)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 (0xc027)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a)
    Cipher Suite: TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014)
    Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013)
    Cipher Suite: TLS_RSA_WITH_AES_256_GCM_SHA384 (0x009d)
    Cipher Suite: TLS_RSA_WITH_AES_128_GCM_SHA256 (0x009c)
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA256 (0x003d)
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA256 (0x003c)
    Cipher Suite: TLS_RSA_WITH_AES_256_CBC_SHA (0x0035)
    Cipher Suite: TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)
    Cipher Suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA (0x000a)
```

Рис. 3. Поддерживаемые криптографические алгоритмы в TLS 1.2

```
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)
```

Рис. 4. Выбранный сервером набор алгоритмов

2. Установка параметров криптосистемы

Как известно, в основе алгоритма RSA лежит возведение в степень по модулю [1, 2].

$$c = m^e \bmod n$$
$$m = c^d \bmod n$$

где e – открытая экспонента для шифрования;
m – открытый текст;
c – шифрованный текст;
d – секретная экспонента для расшифровывания;

$$ed = 1(\mathrm{mod}(p-1)(q-1))$$

$n = p \cdot q$ – модуль системы (открытый).

При этом p, q – секретные простые числа.

Когда создается криптосистема, то для нескольких пользователей могут быть установлены общие параметры. В данном случае для разных пользователей можно установить один модуль n. Но при этом должны быть разные экспоненты шифрования e, так как e^{-1} имеет однозначное значение, и следовательно, при одинаковых e закрытая экспонента d будет одна. Тогда электронная подпись теряет свое свойство – подтверждения авторства, так будет невозможно определить, кто именно подписал документ.

3. Оценка количества фиксированных точек

G. R. Blakley и I. Borosh в своей работе [3] описали, что данный алгоритм шифрования не всегда скрывает сообщение. При использовании алгоритма RSA есть точки, которые в шифрованном и открытом виде равны:

$$m^e = m \bmod n \quad (1)$$

Количество таких точек зависит от использованных параметров e, p, q. Общее количество таких точек s находится по следующей формуле [1, 2]:

$$s = (1 + d_1) \cdot (1 + d_2) \quad (2)$$

где $d_1 = НОД(e-1, p-1), d_2 = НОД(e-1, q-1)$

Раскрыв скобки, получаем следующий результат:

$$s = 1 + d_1 + d_2 + d_1 \cdot d_2 \quad (3)$$

3.1 Определение минимального количества точек

Рассмотрим случай, когда количество фиксированных точек минимально. Так как p, q – простые большие числа, следовательно $p-1$ и $q-1$ четные, e имеет следующие ограничения при выборе:

$$НОД(e, \varphi(n)) = НОД(e, (p-1)(q-1)) = 1$$

где $\varphi(n)$ – функция Эйлера.[6]

Это значит, что e тоже нечетное, а $e-1$, соответственно, – четное. Предположим, что $e-1$ и $p-1$, а также $e-1$ и $p-1$ не имеют общих делителей кроме 2. Следовательно,

$$d_1 = d_2 = 2$$

Из этого получаем, что количество фиксированных точек будет равно:

$$s = 1 + 2 + 2 + 2 \cdot 2 = 9$$

3.2 Определение максимального количества точек

Рассмотрим, при каких параметрах количество фиксированных точек максимально. Так как алгоритм основан на модульной арифметике, то для однозначного дешифрования значения открытого текста не могут быть больше или равны n.

Будем считать, что $e > p$ и $e > q$ Если $e - 1$ можно представить в виде:

$$\begin{cases} e - 1 = (p - 1) \cdot v_1 \\ e - 1 = (q - 1) \cdot v_2 \end{cases}, v_1, v_2 \in \mathbb{N}$$

Следовательно,

$$d_1 = p - 1, d_2 = q - 1$$

Количество точек в этом случае будет равно

$$s = (1 + p - 1)(1 + q - 1) = p \cdot q = n$$

Это значит, что все точки будут открытыми, и шифрование с использованием данных параметров не несет в себе смысла.

Пример

Выберем следующие параметры:

$$p = 1009,$$
$$q = 2003,$$
$$n = 2021027,$$
$$\varphi(n) = 2018016,$$
$$e = 144145$$

Получаем:

$$s = (1 + 1008)(1 + 2002) = 2021027$$

4. Нахождение фиксированных точек

Имеем:

$$m^e = m (\mathrm{mod}\ pq)$$
$$m^e - m = 0 (\mathrm{mod}\ pq)$$
$$m(m^{e-1} - 1) = 0 (\mathrm{mod}\ pq)$$

Последнее уравнение можно заменить системой [4]:

$$\begin{cases} m(m^{e-1} - 1) = 0 (\mathrm{mod}\ p) \\ m(m^{e-1} - 1) = 0 (\mathrm{mod}\ q) \end{cases} \quad (4)$$

Рассмотрим решение уравнения:

$$m(m^{e-1} - 1) = 0 (\mathrm{mod}\ p) \quad (5)$$

Следовательно, из уравнения (5) получаем следующее:

$$\begin{cases} m = 0 (\mathrm{mod}\ p) \\ m^{e-1} = 1 (\mathrm{mod}\ p) \end{cases} \quad (6)$$

$m = 0 (\mathrm{mod}\ p)$ значит, что m – это все числа, кратные p. Так как рассматриваются числа, не превышающие p, это значит m имеет всего одно решение $m = 0$.

Перейдем ко второму уравнению системы (6). По малой теореме Ферма [5] для любого m, не делящегося на p, выполняется $m^{p-1} \equiv 1(\mathrm{mod}\ p)$.

Также имеем циклическую мультипликативную группу по модулю p.

Пусть g – первообразный корень по модулю p. Тогда любое число m, не кратное p, можно представить в виде:

$$m \equiv g^x (\mathrm{mod}\ p)$$

где x – некоторое целое число от 0 до $p-2$.

Обозначим $e-1=k$.

Имеем

$$m^k = (g^x)^k = g^{xk} (\mathrm{mod}\ p)$$

Порядок первообразного корня циклической группы равен $p-1$.

$$g^N = 1(\mathrm{mod}\ p) \tag{7}$$

Равенство (7) выполняется тогда и только тогда, когда $p-1$ делит N. Следовательно,

$$x \cdot k = 0(\mathrm{mod}\ p-1) \tag{8}$$

Обозначим $d = НОД(p-1,k)$. Разделим уравнение (8) на d

$$x\left(\frac{k}{d}\right) = 0\left(\mathrm{mod}\frac{p-1}{d}\right)$$

Теперь, так как мы разделили на НОД левую и правую часть, а также модуль, $\frac{k}{d}$ и $\frac{p-1}{d}$ стали взаимно простыми, следовательно, можно сократить на $\frac{k}{d}$. Получаем:

$$x = 0\left(\mathrm{mod}\frac{p-1}{d}\right)$$

Из данного уравнения получаем, что x имеет следующие решения:

$$x = 0, \frac{p-1}{d}, \frac{2(p-1)}{d}, ..., \frac{(d-1)(p-1)}{d}$$

Всего значений x будет d. Теперь для найденных x получаем значения m:

$$m = g^x (\mathrm{mod}\ p)$$

Итак, для уравнения (5) имеем d решений. Для уравнения по модулю q, решение аналогично.

Когда найдены решения для модулей p и q по отдельности, теперь надо перейти к модулю n. Для этого нужно решить следующие системы:

$$\begin{cases} m = 0 \bmod p \\ m = 0 \bmod q \end{cases} \tag{9}$$

$$\begin{cases} m = 0 \bmod p \\ m^{e-1} = 1 \bmod q \end{cases} \tag{10}$$

$$\begin{cases} m^{e-1} = 1 \bmod p \\ m = 0 \bmod q \end{cases} \tag{11}$$

$$\begin{cases} m^{e-1} = 1 \bmod p \\ m^{e-1} = 1 \bmod q \end{cases} \quad (12)$$

Так как решения уравнений из системы (9) дают по одному решению, то и решение системы будет одно. В системах (10) и (11) одно уравнение дает одно решение, а другое d_1 и d_2 соответственно, следовательно перебирая различные корни уравнений получим d_1 и d_2 решений соответственно. В системе (12) аналогично, перебирая различные решения уравнений, которых d_1 и d_2. Общее количество решений последней системы будет равно d_1 и d_2.

Для каждого уравнения в системах есть некоторое количество решений. В общем случае можно это записать так:

$$\begin{cases} m = a(\bmod p) \\ m = b(\bmod q) \end{cases}$$

Решение данных систем основано на Китайской теореме об остатках (КТО) [1, 5]:

$$M = \left[a\left(q^{-1}\left(\bmod p\right)\right)q + b\left(p^{-1}\left(\bmod q\right)\right)p \right] \bmod n$$

где $q^{-1}(\bmod p), p^{-1}(\bmod q)$ – обратные элементы в кольце по модулям p и q соответственно.

Суммируя все полученные ответы, имеем:

$$s = 1 + d_1 + d_2 + d_1 \cdot d_2$$

Что в точности равно уравнению (3).

5. Стационарные фиксированные точки

Рассмотрим случай, когда количество точек минимально и равно 9. Следовательно $d_1 = d_2 = 2$.

Обозначим m_p – остаток от деления на p, m_q – остаток от деления на q.

Решение уравнения (5) имеет три решения.

$$m_p = 0, 1, p-1$$

Для q аналогично

$$m_q = 0, 1, q-1$$

Перебирая все возможный комбинации m_p и m_q, получим следующий результат:

$$\begin{cases} m_p = 0 \\ m_q = 0 \end{cases} \quad (13)$$

$$\begin{cases} m_p = 0 \\ m_q = 1 \end{cases} \quad (14)$$

$$\begin{cases} m_p = 0 \\ m_q = q-1 \end{cases} \quad (15)$$

$$\begin{cases} m_p = 1 \\ m_q = 0 \end{cases} \quad (16)$$

$$\begin{cases} m_p = 1 \\ m_q = 1 \end{cases} \quad (17)$$

$$\begin{cases} m_p = 1 \\ m_q = q - 1 \end{cases} \tag{18}$$

$$\begin{cases} m_p = p - 1 \\ m_q = 0 \end{cases} \tag{19}$$

$$\begin{cases} m_p = p - 1 \\ m_q = 1 \end{cases} \tag{20}$$

$$\begin{cases} m_p = p - 1 \\ m_q = q - 1 \end{cases} \tag{21}$$

Решая данные системы, получаем следующие результаты:

- система (13) даст решение $m = 0$;
- система (16) даст решение $m = 1$;
- система (21) дает решение $m = -1 = n - 1$;
- системы (16), (19) дают решение вида $m = t_1 q$;
- система (14), (15) дают решение вида $m = t_2 p$;
- система (18) дает решение вида $m = t_3 p + 1$;
- система (20) дает решение вида $m = t_4 q + 1$;

где $t_1, t_2, t_3, t_4 \in \mathbb{N}$.

Среди полученных результатов, есть три тривиальные точки: $0, 1, n-1$. Остальные точки интересны тем, что при разложении на простые множители, максимальный простой множитель будет p или q.

7. Оценка распределения фиксированных точек

Как было показано ранее, значение фиксированных точек в алгоритме RSA не зависит от e, оно зависит только от p и q. От e зависит только количество фиксированных точек.

Рассмотрим распределение количества фиксированных точек при разных параметрах. Зафиксируем p, q и будем изменять e в пределах от 1 до $(p-1)(q-1)$. Выберем $p = 1009, q = 2003$. Результаты приведены на рисунке 5.

Видно, что график носит периодический характер. Максимальное количество точек равно 2021027, что равняется произведению p и q. Значения e, при которых получено максимальное количество точек приведено в таблице 1.

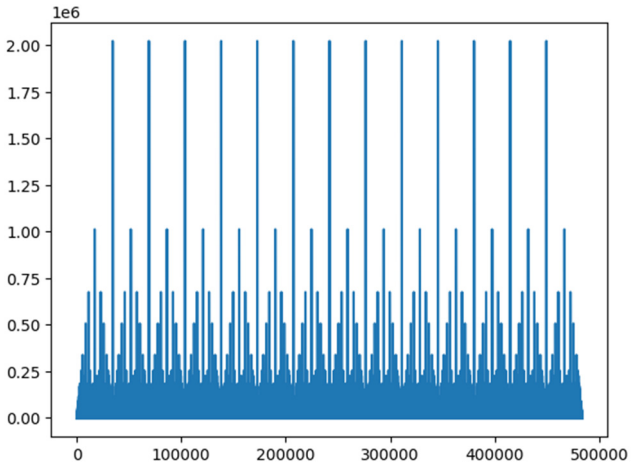

Рис. 5. Распределение количества фиксированных точек при p=1009, q=2003

Таблица 1

Значение *e* с максимальным количеством фиксированных точек

№	e	$e-1$	$t_1(p-1)$	$t_2(q-1)$
1	144145	144144	143·1008	72·2002
2	288289	288288	286·1008	144·2002
3	432433	432432	429·1008	216·2002
4	576577	576576	572·1008	288·2002
5	720721	720720	715·1008	360·2002
6	864865	864864	858·1008	432·2002
7	1009009	1009008	1001·1008	504·2002
8	1153153	1153152	1144·1008	576·2002
9	1297297	1297296	1287·1008	648·2002
10	1441441	1441441	1430·1008	720·2002
11	1585585	1585584	1573·1008	792·2002
12	1729729	1729728	1716·1008	864·2002
13	1873873	1873872	1859·1008	936·2002

Из таблицы 1 видно, что значение $e-1$ делится нацело на $p-1$ и $q-1$. Из-за чего и получается, что все точки будут открыты.

Рассмотрим распределение фиксированных точек при выбранном *e* и изменяемым *p, q*. Возьмем *e* из числа стандартных чисел, т.е. из чисел Ферма, вида $2^{2^i}+1, i \in \mathbb{N}$ равное 65537. Это число часто выбирается в качестве открытой экспоненты, так как оно простое и содержит всего две единицы в двоичном представлении, что упрощает возведение в степень при использовании алгоритма быстрого возведения в степень по модулю, а также обеспечивает достаточно большое значение секретной экспоненты *d*. Числа *p*, *q* будут выбираться из списка простых чисел в диапазоне от 40000 до 50000 (рис. 6).

Самый большой пик имеет значение 4203009, он получен при следующих значениях

$$p = 40961 = 2^{13} \cdot 5 + 1$$

$$q = 45569 = 2^{9} \cdot 89 + 1$$

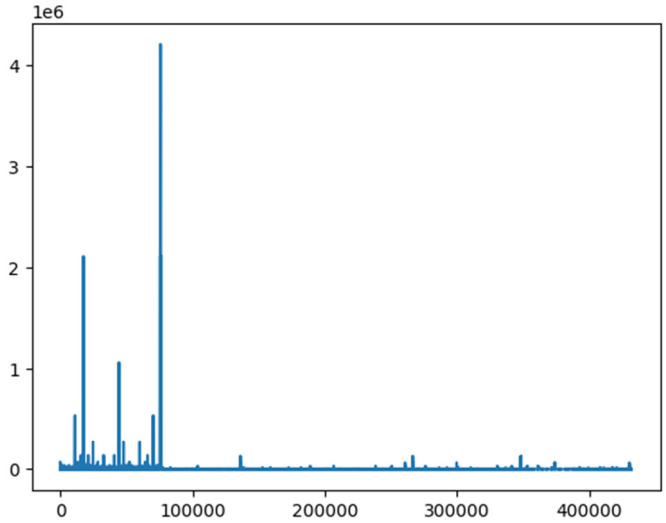

Рис. 6. Распределение количества фиксированных точек при *e*=65537, *p* и *q* ∈ (40000, 50000)

Такое большое количество фиксированных точек возникает из-за того, что *p*, *q* представляются в виде

$$2^j k + 1 \qquad (22)$$

где $j, k \in \mathbb{N}$.

В связи с чем

$$d_1 = 2^{\min(j_p, i)} \tag{23}$$

$$d_2 = 2^{\min(j_q, i)} \tag{24}$$

где j_p, j_q – степень двойки из *p, q* соответственно, i – степень двойки из *e*.

Из данного примера получаем для (23) и (24) следующие значения:

$d_1 = 2^{13} = 8192$

$d_2 = 2^9 = 512$

Посчитаем количество фиксированных точек:

$s = 1 + 8912 + 512 + 8192 \cdot 512 = 4203729$

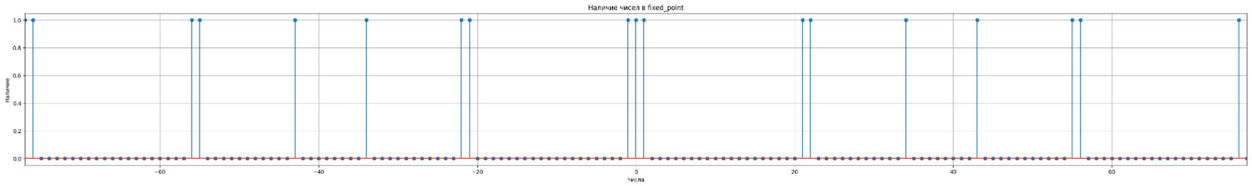

Рис. 7. Распределение стационарных фиксированных точек

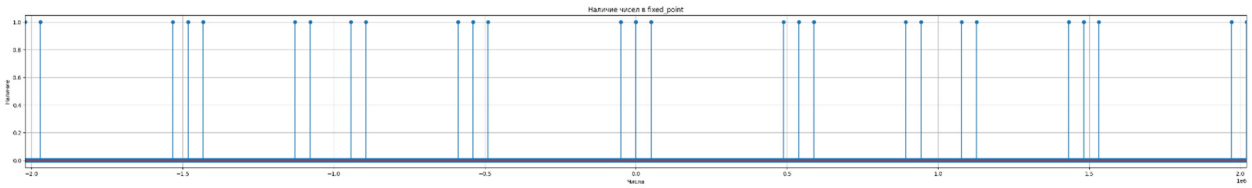

Рис. 8. Распределение точек при *p*=1009, *q*=2003, *e*=31

Так что, если предположить, что есть еще простые числа вида (6.1) и степень $j \geq 16$, количество фиксированных точек будет равно:

$$s = 1 + 2 \cdot 2^{16} + \left(2^{16}\right)^2 = 1 + 2^{17} + 2^{32} = 4295098369$$

7. Распределение стационарных фиксированных точек

Как уже говорилось, количество точек зависит от *e*. Но стационарные девять точек будут всегда, не зависимо от выбранной экспоненты.

Рассмотрим, случай, когда у нас девять точек. Три точки всегда известны, это 0, 1, *n*-1 – тривиальные точки. График обладает вертикальной симметрией, с осью проходящей через нуль, как показано на рис. 7.

Симметрия возникает из-за условий появления фиксированной точки и открытой экспоненты *e*. Так как *e* – нечетная получаем, что:

$$(-x)^e (\mathrm{mod}\, n) = -(x^e) (\mathrm{mod}\, n)$$

Следовательно, если в (3.1) взять вместо *m,* отрицательное значение, то получим

$$-m = (-m^e) (\mathrm{mod}\, n) = -(m^e) (\mathrm{mod}\, n)$$

Получается, если m была фиксированной точкой, то и $-m$ тоже фиксированная точка.

Когда имеются только стационарные фиксированные точки, решая уравнение из системы (4) относительно модуля *p*, получаем три корня, симметричных относительно нуля.

$m_p = 0, 1, p-1$

$-0 (\mathrm{mod}\, p) = 0 (\mathrm{mod}\, p)$

$$-1(\text{mod } p) = p - 1(\text{mod } p)$$
$$-(p-1)(\text{mod } p) = -p + 1(\text{mod } p) = 1(\text{mod } p)$$

Для q аналогичные рассуждения.

Из этого следует, что решение систем по КТО:

$$\begin{cases} m_{f1} = m_p \\ m_{f1} = m_q \end{cases}$$
$$\begin{cases} m_{f1} = -m_p \\ m_{f1} = -m_q \end{cases}$$

Будут противоположны:

$$m_{f1} = -m_{f2}(\text{mod } n)$$

Таким образом, решение системы (14) будет противоположно решению (15) по модулю n. Аналогично (16) противоположно (19), (17) противоположно (21), а (18) противоположно (20).

Итак, зная хотя бы одну фиксированную точку, отличную от нуля, можно найти вторую, противоположную данной.

Рассмотрим распределение точек следующих значений: $p = 1009, q = 2003, e = 31$.

По формуле (2) получаем количество фиксированных точек:

$$s = (1+6)\cdot(1+2) = 7\cdot 3 = 21$$

Также воспользовавшись методом, описанным в части 2, можно найти все эти точки. Распределение этих точек приведено на рис. 8. Точки приведены в листинге 1. Разложим каждую точку на простые множители. Как видно из листинга, точки 3, 6, 7, 9, 14, 16, 17, 20 имеют в своем разложение p или q.

Есть точки, которые являются простыми числами, такие как: 8, 10.

Отдельно стоит заметить, что часть точек отличается на 1, т.е. идут друг за другом и на рисунке 8 они сливаются. Это точки под номерами: 3 и 4, 6 и 7, 8 и 9, 14 и 15, 16 и 17, 19 и 20.

Количество точек, имеющих в себе p или q, зависит от d_1, d_2. Так как они появляются из систем (10) и (11). И, следовательно, их количество равно:

$$s_{p,q} = d_1 + d_2$$

В данном примере таких точек 8.

Листинг 1. *Фиксированные точки*

1. $0 = 0$
2. $1 = 1$
3. $50075 = 5^2 \cdot 2003^1$
4. $50076 = 2^2 \cdot 3^2 \cdot 13^1 \cdot 107^1$
5. $488731 = 227^1 \cdot 2153^1$
6. $538806 = 2^1 \cdot 3^1 \cdot 89^1 \cdot 1009^1$
7. $538807 = 269^1 \cdot 2003^1$
8. $588881 = 588881^1$
9. $588882 = 2^1 \cdot 3^1 \cdot 7^2 \cdot 2003^1$
10. $893339 = 893339^1$
11. $943414 = 2^1 \cdot 23^1 \cdot 20509^1$
12. $1077613 = 17^1 \cdot 63389^1$
13. $1127688 = 2^3 \cdot 3^1 \cdot 19^1 \cdot 2473^1$
14. $1432145 = 5^1 \cdot 11^1 \cdot 13^1 \cdot 2003^1$
15. $1432146 = 2^1 \cdot 3^1 \cdot 238691^1$
16. $1482220 = 2^2 \cdot 5^1 \cdot 37^1 \cdot 2003^1$

17. $1482221 = 13^1 \cdot 113^1 \cdot 1009^1$
18. $1532296 = 2^3 \cdot 191537^1$
19. $1970951 = 211^1 \cdot 9341^1$
20. $1970952 = 2^3 \cdot 3^1 \cdot 41^1 \cdot 2003^1$
21. $2021026 = 2^1 \cdot 7^1 \cdot 241^1 \cdot 599^1$

8. Взлом RSA с помощью стационарных фиксированных точек

Будем рассматривать стационарные точки, которые не являются тривиальными. Таких точек шесть. И получены они из решения систем (14), (15), (16), (17), (18) и (19).

Как уже говорилось, решение данных систем имеет вид:

- $m_1 = t_1 q$
- $m_2 = t_2 q$
- $m_3 = t_3 p + 1$
- $m_4 = t_4 q + 1$

Нетрудно заметить, что, разложив точки m_1 и m_2 на простые множители, получим значения p и q. В то время как разложение m_3 и m_4 не даст результата, в этих случаях надо раскладывать $m_3 - 1$ и $m_4 - 1$. Но сложность факторизации все еще достаточно большая. Так сложность факторизации методом Ферма [1] имеет экспоненциальную сложность. В работе Bahig [6] описан усовершенствованный алгоритм Ферма.

Можно воспользоваться алгоритмом Евклида для нахождения НОД [6]. Так как n, делится только на p и q, то результатом вычисления НОД для m_1, m_2 будут значения q, p соответственно. Для m_3 и m_4 вычисление НОД даст результат 1. Но можно вычислить $НОД(m_3 - 1, n)$ и $НОД(m_4 - 1, n)$, которые будут равны p, q соответственно.

Сложность вычисления НОД по усовершенствованному алгоритму Евклида, согласно [7] равно

$$O\left(\log_2 n\left(\log_2 n + k\log_2 n + \frac{n}{2}\right) + \frac{n}{2}\log_2\frac{n}{2}\right)$$

Заключение

Исследование фиксированных точек в алгоритме RSA демонстрирует критическую важность внимательного подхода к выбору параметров криптосистемы. Особенно это актуально в условиях вычислительных ограничений, когда могут быть выбраны относительно небольшие значения простых чисел p и q. При достаточно больших значениях p и q количество фиксированных точек будет минимальным, что обеспечивает высокую надежность системы. Однако, как показано в работе, при определенных комбинациях параметров количество фиксированных точек может значительно возрастать, что потенциально снижает криптостойкость.

Анализ методов нахождения фиксированных точек и их распределения подтверждает, что даже при минимальном количестве фиксированных точек (равном 9) существуют стационарные точки, которые могут быть использованы для получения информации о секретных параметрах p и q. В частности, показано, что некоторые нетривиальные фиксированные точки могут содержать в себе информацию, позволяющую, например, с помощью алгоритма Евклида вычислить p и q, хотя сложность факторизации остается высокой для больших чисел.

Таким образом, для поддержания криптостойкости алгоритма RSA крайне важно не только выбирать достаточно большие значения p и q, но и тщательно подходить к выбору открытой экспоненты e, избегая значений, которые могут приводить к аномально большому количеству фиксированных точек. Полученные в данной работе результаты предоставляют ценные сведения для разработчиков и исследователей в области криптографии, позволяя им принимать обоснованные решения при проектировании и использовании криптосистем на основе RSA.

Литература

1. *Ильин М.Е., Калинкина Т.И., Пржегорлинский В.Н.* Криптографическая защита информации в объектах информационной инфраструктуры // учеб. для студ. учреждений сред. проф. образования. М.: Издательский центр «Академия», 2019.

2. Boneh Dan «Twenty Years of Attacks on the RSA Cryptosystem» // Notices of the American Mathematical Society. 1999. № 46(3). С. 203-213.

3. *Blakley G. R. and Borosh I.* Rivest-Shamir-Adleman public key cryptosystems do not always conceal messages // Computers & Mathematics with Applications. 1979. № 5(3). С. 169-178.

4. *Andrzej Chmielowiec*. Fixed points of the RSA encryption algorithm // Theoretical Computer Science 2010 № 411(1). С. 288-292.

5. *Мартынов Л.М.* Алгебра и теория чисел для криптографии // учебное пособие для вузов. СПб. 2025. 456 с.

6. *Bahig H,* Speeding Up Fermat's Factoring Method using Precomputation // Annals of Emerging Technologies in Computing. 2022. № 6(2). С. 50-60.

7. *Амер И.Ф., Аль Халиди А.М*. Два быстрых метода нахождения наибольшего общего делителя // Вестник российского нового университета. серия: сложные системы: модели, анализ и управление. 2021. №1. С. 150-158.

АНАЛИЗ ПОДХОДОВ К ПРОГНОЗИРОВАНИЮ СОСТОЯНИЯ ОБОРУДОВАНИЯ В БЕСПРОВОДНЫХ СЕНСОРНЫХ СЕТЯХ И ВОЗМОЖНОСТЕЙ ИХ ИНТЕГРАЦИИ С ТЕХНОЛОГИЯМИ БЛОКЧЕЙНА

Фатхулин Тимур Джалилевич
Московский технический университет связи и информатики,
доцент кафедры ИАД, к.т.н., Москва, Россия
t.d.fatkhulin@mtuci.ru

Бобков Данила Борисович
Московский технический университет связи и информатики,
студентка группы МБД2431, Москва, Россия

Рахматова Азиза Акрамовна
Московский технический университет связи и информатики,
студентка группы М62402, Москва, Россия

Аннотация
В статье представлен аналитический обзор современных подходов к прогнозированию состояния оборудования в беспроводных сенсорных сетях, применяемых в системах промышленного интернета вещей. Рассматриваются методы статистического анализа, классического машинного обучения и глубоких нейронных сетей, включая рекуррентные, attention- и графовые модели, используемые для обработки многомерных сенсорных временных рядов.

Ключевые слова
беспроводные сенсорные сети; прогнозирование; состояния оборудования; предиктивное обслуживание; машинное обучение; глубокое обучение; временные ряды

Введение

Беспроводные сенсорные сети (Wireless Sensor Networks, WSN) широко применяются в системах мониторинга технического состояния оборудования в рамках концепций Industrial Internet of Things (IIoT) [1, 2]. Использование распределённых сенсорных узлов позволяет в реальном времени собирать данные о параметрах функционирования оборудования и реализовывать прогнозное обслуживание, направленное на снижение простоев и эксплуатационных затрат [3]. В последние годы для прогнозирования состояния оборудования и показателей остаточного ресурса активно применяются методы машинного и глубокого обучения, включая рекуррентные нейронные сети, attention-механизмы и графовые модели [4-6].

Вместе с тем практическое использование таких методов в WSN осложняется проблемами качества и доверия к сенсорным данным. Ограниченные ресурсы узлов, потери пакетов, деградация датчиков и дрейф данных приводят к снижению надёжности и воспроизводимости прогнозов [7, 8]. Дополнительную угрозу представляют атаки на сенсорные сети, связанные с подменой и повторной передачей данных [9]. В этих условиях особую актуальность приобретает задача обеспечения целостности, трассируемости и подтверждённого происхождения данных, используемых в алгоритмах прогнозирования.

В качестве возможного решения рассматриваются технологии блокчейна, обладающие свойствами неизменяемости, децентрализации и распределённого управления доверием [10-12]. Блокчейн может выступать в роли инфраструктуры доверия для сенсорных данных и результатов аналитики, однако его интеграция в WSN связана с дополнительными накладными расходами и архитектурными ограничениями [13]. Настоящая статья посвящена анализу современных подходов к прогнозированию состояния оборудования в беспроводных сенсорных сетях и рассмотрению возможностей и ограничений их интеграции с технологиями блокчейна.

Особенности беспроводных сенсорных сетей в задаче мониторинга состояния оборудования

Беспроводные сенсорные сети представляют собой распределённые системы, состоящие из множества автономных узлов, осуществляющих сбор и передачу измерений физических параметров оборудования. В задачах мониторинга технического состояния такие сети обеспечивают непрерывное наблюдение за вибрационными, температурными, акустическими и электрическими характеристиками, позволяя выявлять процессы деградации на ранних стадиях [1, 3]. В отличие от традиционных систем мониторинга, WSN характеризуются ограниченными вычислительными ресурсами, жёсткими энергетическими ограничениями и нестабильными каналами связи, что существенно влияет на выбор методов обработки данных и прогнозирования [7].

Сенсорные данные, формируемые в WSN, обладают высокой временной разрешающей способностью и выраженной неоднородностью, обусловленной различиями в типах датчиков, условиях эксплуатации и топологии сети. Кроме того, для таких данных характерны пропуски, выбросы и некорректные измерения, вызванные деградацией сенсорных узлов, интерференцией и потерями пакетов [8]. Эти особенности усложняют применение классических методов анализа временных рядов и требуют использования более устойчивых алгоритмов прогнозирования.

В работе показано, что при практической эксплуатации беспроводных сенсорных сетей ключевым ограничивающим фактором эффективности прогнозирования выступает не только точность используемых моделей, но и доверие к сенсорным данным, обусловленное их уязвимостью к пропускам, шуму, деградации датчиков и преднамеренным воздействиям.

Особое внимание уделено формализации проблемы доверия к данным и анализу технологий блокчейна как инфраструктурного слоя, обеспечивающего целостность, трассируемость и подтверждённое происхождение сенсорной информации и результатов аналитической обработки. Рассмотрены архитектурные подходы к интеграции методов машинного обучения и блокчейн-технологий в условиях ресурсных ограничений беспроводных сенсорных сетей, а также проанализированы связанные с этим накладные расходы, включая задержки, рост сетевого трафика и проблемы масштабируемости.

В результате показано, что наиболее перспективными являются гибридные on-chain/off-chain архитектуры, в которых блокчейн используется для регистрации метаданных и криптографических представлений данных, а вычислительно затратные операции прогнозирования выполняются на уровне edge- или облачных узлов. Полученные выводы могут служить основой для разработки доверенных и воспроизводимых систем прогнозного обслуживания оборудования в беспроводных сенсорных сетях и определения направлений дальнейших исследований в области интеграции машинного обучения и распределённых реестров

Классические методы прогнозирования состояния оборудования

Ранние подходы к прогнозированию состояния оборудования в сенсорных сетях основывались на статистических моделях и методах регрессионного анализа. К ним относятся авторегрессионные модели, методы главных компонент и линейные регрессионные зависимости, применяемые для выявления трендов и аномалий в измерениях [7]. Несмотря на простоту и интерпретируемость, такие методы демонстрируют ограниченную эффективность при работе с нелинейными и высокоразмерными данными, характерными для современных WSN.

С развитием машинного обучения получили распространение методы опорных векторов, случайных лесов и градиентного бустинга, применяемые для классификации состояний и оценки остаточного ресурса оборудования [3]. Эти методы обеспечивают более высокую точность по сравнению со статистическими моделями, однако требуют тщательной настройки признакового пространства и, как правило, не учитывают сложные временные и пространственные зависимости между сенсорными узлами.

Современные методы глубокого обучения

В последние годы основное внимание исследователей сосредоточено на применении методов глубокого обучения для прогнозирования состояния оборудования в WSN. Рекуррентные нейронные сети, в частности архитектуры LSTM и GRU, широко используются для анализа многомерных временных рядов и прогнозирования показателей остаточного ресурса [4, 5]. Эти модели способны учитывать долгосрочные временные зависимости, однако их эффективность снижается при наличии пропусков данных и нестабильных режимов работы оборудования.

Дальнейшее развитие получили attention-механизмы и Transformer-подходы, позволяющие более гибко моделировать временные зависимости и повышать устойчивость к шуму в данных [5]. Для задач, где существенную роль играют пространственные корреляции между сенсорными узлами, применяются графовые нейронные сети и пространственно-временные модели, в которых WSN представляется в виде графа [6]. Такие подходы демонстрируют высокую точность прогнозирования, однако требуют значительных вычислительных ресурсов и сложной процедуры обучения.

Проблемы надёжности и воспроизводимости прогнозов

Несмотря на достигнутые успехи, современные методы прогнозирования состояния оборудования в WSN остаются чувствительными к качеству входных данных. Пропуски измерений, аномалии сенсорных узлов и дрейф данных приводят к снижению стабильности прогнозов и затрудняют их воспроизводимость [8]. Дополнительным фактором риска является возможность подмены или искажения сенсорных данных, что особенно критично для распределённых беспроводных сетей [9].

Таким образом, повышение точности моделей машинного обучения само по себе не гарантирует надёжности прогнозирования в реальных условиях эксплуатации. Это обусловливает необходимость комплексного подхода, сочетающего методы интеллектуального анализа данных с механизмами обеспечения доверия, целостности и трассируемости сенсорной информации (табл. 1) [14-28].

Таблица 1

Эволюция методов анализа прогнозирования состояния оборудования

Класс методов	**Типовые модели**	**Учитываемые зависимости**	**Преимущества**	**Ограничения в WSN**
Статистические методы	AR, ARIMA, PCA, линейная регрессия	Временные (локальные)	Простота реализации, интерпретируемость	Низкая точность при нелинейностях, чувствительность к шуму
Классические ML-методы	SVM, Random Forest, Gradient Bosting	Зависит от признаков	Более высокая точность по сравнению со статистическими методами	Требуют ручного выделения признаков, слабая адаптация к дрейфу данных
Рекурентные нейроные сети	LSTM, GRU	Временные (долгосрочные)	Эффективны для многомерных временных рядов	Чувствительны к пропускам данных, высокая вычислительная сложность
Attention/Transformer модели	TemporalTransformer, Informer	Временные (глобальные)	Улучшенная обработка долгосрочных зависимостей, устойчивость к шуму	Высокие требования к ресурсам, сложность edge-реализации
Графовые нейронные сети	STGNN, GCN-LSTM	Пространственно-временные	Учитывают топологию WSN и корреляции между узлами	Сложность построения графа, масштабируемость
Edge-ориентированные модели	Lightweight DL, TinyML	Временные (локальные)	Низкая задержка, снижение трафика	Ограниченная точность, необходимость компромиссов

Как следует из таблицы 1, эволюция методов прогнозирования состояния оборудования в WSN демонстрирует переход от интерпретируемых, но ограниченных статистических моделей к более точным, но ресурсозатратным архитектурам глубокого обучения. Наиболее перспективными с точки зрения точности являются attention-механизмы и графовые нейронные сети, однако их практическое применение в беспроводных сенсорных сетях осложняется требованиями к вычислительным ресурсам и устойчивости к деградации данных. Это подчёркивает необходимость дополнительных механизмов, направленных на повышение доверия и воспроизводимости прогнозов, что создаёт предпосылки для интеграции методов машинного обучения с технологиями блокчейна.

Формализация проблемы доверия к данным в контексте методов прогнозирования

Сравнительный анализ методов прогнозирования состояния оборудования, представленный в таблице 1, показывает, что повышение точности прогнозирования достигается преимущественно за счёт увеличения сложности моделей и расширения учитываемых зависимостей. Однако вне зависимости от используемого класса моделей все рассмотренные подходы опираются на предположение о достоверности и корректности входных сенсорных данных. В условиях беспроводных сенсорных сетей данное предположение часто не выполняется, что приводит к систематическим ошибкам прогнозирования и снижению практической ценности интеллектуальных моделей.

С формальной точки зрения сенсорные данные в WSN могут рассматриваться как последовательность измерений:

$$X = \{xi(t)\},\ i = 1,\ldots,N \ ,$$

где $xi(t)$ – измерение, полученное от i -го сенсорного узла в момент времени t. В реальных условиях эксплуатации часть элементов X может быть искажена вследствие аппаратных сбоев, деградации датчиков, потерь пакетов или преднамеренных воздействий. При этом большинство моделей, представленных в таблице 1, минимизируют функцию ошибки вида $L(f(X),y)$ не обладая механизмами проверки целостности и происхождения элементов X.

Отсутствие формализованных механизмов доверия приводит к тому, что:

1. Невозможно однозначно установить происхождение данных, использованных для обучения или инференса модели;
2. Результаты прогнозирования не являются воспроизводимыми в условиях повторного анализа или аудита;
3. Аномалии и подмены данных интерпретируются как реальные процессы деградации, что особенно критично для глубоких моделей, чувствительных к распределению входных данных.

Таким образом, проблема доверия к данным в WSN носит не вспомогательный, а фундаментальный характер, затрагивая все классы методов прогнозирования, представленные в Таблице 1. Это позволяет рассматривать доверие к данным как отдельное системное свойство, ортогональное выбору конкретной модели машинного обучения. Следовательно, решение данной проблемы требует внедрения дополнительных инфраструктурных механизмов, обеспечивающих целостность, неизменяемость и трассируемость сенсорной информации на протяжении всего жизненного цикла данных.

В этой связи технологии блокчейна представляют собой перспективный инструмент для формирования доверенной среды обработки сенсорных данных, не зависящей от конкретной реализации прогностической модели. Анализ таких технологий и возможностей их интеграции с методами прогнозирования состояния оборудования в беспроводных сенсорных сетях рассматривается в следующем разделе статьи.

Блокчейн как инфраструктура доверия к сенсорным данным

Блокчейн-технологии представляют собой распределённые реестры, обеспечивающие неизменяемое и согласованное хранение данных между участниками системы без необходимости доверия к централизованному управляющему узлу. Ключевые свойства блокчейна – децентрализация, криптографическая защищённость, неизменяемость записей и возможность аудита – делают его перспективным инструментом для повышения доверия к данным, формируемым в беспроводных сенсорных сетях [10-12].

В контексте мониторинга состояния оборудования блокчейн может использоваться для регистрации происхождения сенсорных данных, временных меток, идентификаторов узлов и параметров качества измерений. Такой подход позволяет обеспечить воспроизводимость и проверяемость данных, используемых в алгоритмах прогнозирования, что особенно важно в условиях деградации сенсорных узлов и нестабильных каналов связи, характерных для WSN [11]. При этом блокчейн выступает в роли инфраструктурного слоя, ортогонального методам машинного обучения, и не влияет напрямую на архитектуру прогностических моделей.

На практике блокчейн редко интегрируется на уровне отдельных сенсорных узлов из-за ограниченных вычислительных и энергетических ресурсов. Более распространённым является архитектурный подход, при котором функции взаимодействия с блокчейном реализуются на уровне шлюзов или edge-устройств, агрегирующих данные от множества сенсоров [12]. Это позволяет снизить нагрузку на сеть и сохранить применимость блокчейна в реальных промышленных условиях.

Ограничения и накладные расходы применения блокчейна в WSN

Несмотря на потенциальные преимущества, использование блокчейн-технологий в беспроводных сенсорных сетях сопряжено с рядом существенных ограничений. Одним из ключевых факторов является дополнительная задержка, связанная с подтверждением транзакций и достижением консенсуса между узлами блокчейн-сети. Даже в permissioned-блокчейнах с оптимизированными механизмами консенсуса эта задержка может быть критичной для приложений, требующих близкого к реальному времени реагирования [13].

Дополнительные накладные расходы возникают за счёт увеличения объёма передаваемых данных и необходимости хранения состояния распределённого реестра. Частая регистрация сенсорных измерений или результатов обработки данных в блокчейне может привести к росту сетевого трафика и нагрузке на узлы, что противоречит энергетическим и пропускным ограничениям WSN [11]. В этой связи особую роль играет выбор стратегии ончейн-регистрации, предполагающей хранение в блокчейне только компактных криптографических представлений данных и метаданных.

Ещё одним ограничением является сложность масштабирования блокчейн-решений при увеличении числа сенсорных узлов и частоты измерений. Рост количества транзакций может негативно сказаться на пропускной способности системы и потребовать дополнительных вычислительных ресурсов на стороне шлюзов и edge-устройств [10]. Кроме того, управление криптографическими ключами и механизмами доступа в крупномасштабных WSN представляет собой отдельную инженерную задачу.

Таким образом, эффективность применения блокчейна в беспроводных сенсорных сетях определяется балансом между уровнем обеспечиваемого доверия и накладными расходами, связанными с задержками, энергопотреблением и масштабируемостью. Это указывает на необходимость тщательно продуманной архитектуры интеграции блокчейн-технологий с методами машинного обучения для прогнозирования состояния оборудования, анализ которой рассматривается в следующем разделе статьи.

Архитектурные подходы к интеграции ML и блокчейна в WSN

В современных исследованиях интеграция методов машинного обучения и блокчейн-технологий в беспроводных сенсорных сетях рассматривается как средство повышения доверия и воспроизводимости прогнозирования, а не как механизм улучшения точности моделей [10-12]. В таких системах машинное обучение используется для анализа сенсорных временных рядов и формирования прогнозов состояния оборудования, тогда как блокчейн выполняет роль инфраструктурного слоя, обеспечивающего контроль целостности, происхождения и жизненного цикла данных и моделей (табл. 2).

Таблица 2

Аналитическая оценка эффективности интеграции

Подход к интеграции	**Роль ML**	**Роль блокчейна**	**Преимущества**	**Ограничения**
Blockhain как журнал аудита	Централизованное прогнозирования	Регистрация результатов и метаданных	Простота реализации, низкие накладные расходы	Ограниченная децентрализация
Hybrid on-chain / off-chain	Edge/Cloud ML	Хранение хэшей данных и версий моделей	Баланс доверия и производительности	Требует продуманной архитектуры
Blockchain-assisted FL	Фередартивное обучение	Координация и аудит обучения	Повышенная приватность и доверие	Рост сложности и задержек
Blockchain-centric подходы	Вспомогательная роль ML	Управление данными и доступом	Высокий уровень доверия	Непреминимы для WSN

Наиболее распространённым является гибридный архитектурный подход, при котором вычислительно затратные операции обучения и инференса выполняются на edge- или облачных узлах, а блокчейн используется для регистрации ограниченного набора артефактов, включая хэши сенсорных данных, метаданные качества измерений и версии используемых моделей [11]. Такой подход позволяет минимизировать накладные расходы и сохранить применимость интеграции в условиях ограниченных ресурсов WSN. В ряде работ блокчейн также используется для координации распределённого и федеративного обучения, обеспечивая проверяемость вкладов отдельных участников без раскрытия исходных данных [12].

Сравнительный анализ существующих подходов показывает, что интеграция блокчейн-технологий и методов машинного обучения наиболее эффективна в архитектурах, где блокчейн используется для регистрации метаданных и артефактов обработки данных, а не для хранения или непосредственной обработки сенсорных потоков. Такой подход позволяет повысить воспроизводимость и надёжность прогнозов без существенного ухудшения эксплуатационных характеристик системы.

В то же время чрезмерная интеграция блокчейна, выражающаяся в высокой частоте транзакций или регистрации объёмных данных, приводит к росту задержек, сетевого трафика и энергопотребления, что делает такие решения малопригодными для беспроводных сенсорных сетей. Это подчёркивает необходимость компромиссного проектирования систем прогнозирования, в которых уровень обеспечиваемого доверия соотносится с допустимыми накладными расходами.

Заключение

В данной статье выполнен комплексный анализ современных подходов к прогнозированию состояния оборудования в беспроводных сенсорных сетях, а также рассмотрены возможности и ограничения их интеграции с технологиями блокчейна. Показано, что развитие методов машинного и глубокого обучения, включая рекуррентные нейронные сети, attention-механизмы и графовые модели, позволяет эффективно выявлять процессы деградации и прогнозировать отказы оборудования на основе многомерных сенсорных данных. Вместе с тем в условиях WSN практическая ценность таких моделей существенно ограничивается проблемами качества, целостности и воспроизводимости сенсорной информации, обусловленными ресурсными ограничениями узлов, нестабильностью каналов связи и потенциальными угрозами безопасности.

Проведённый обзор демонстрирует, что повышение точности моделей само по себе не обеспечивает надёжности прогнозирования в реальных условиях эксплуатации. Доверие к данным следует рассматривать как фундаментальное системное свойство, не зависящее от выбора конкретного класса алгоритмов машинного обучения и оказывающее прямое влияние на устойчивость и интерпретируемость прогнозов. В этом контексте технологии блокчейна могут быть использованы в качестве инфраструктуры доверия, обеспечивающей неизменяемость, трассируемость и подтверждённое происхождение сенсорных данных и результатов аналитической обработки.

Показано, что наибольшую практическую применимость для беспроводных сенсорных сетей имеют гибридные архитектуры интеграции машинного обучения и блокчейна, предполагающие выполнение вычислительно затратных операций обучения и инференса на уровне edge- или облачных узлов и регистрацию в распределённом реестре лишь ограниченного набора метаданных и криптографических представлений данных. Такой подход позволяет повысить воспроизводимость и надёжность прогнозов без существенного роста задержек и энергопотребления. В то же время чрезмерная интеграция блокчейна, выражающаяся в высокой частоте транзакций или хранении объёмных данных on-chain, приводит к ухудшению эксплуатационных характеристик и снижает применимость решений в условиях WSN.

Полученные результаты позволяют сделать вывод о необходимости системного проектирования доверенных систем прогнозирования состояния оборудования, в которых методы машинного обучения и блокчейн-технологии рассматриваются как взаимодополняющие компоненты. Перспективными направлениями дальнейших исследований являются количественная оценка компромиссов между уровнем обеспечиваемого доверия и накладными расходами, интеграция федеративного обучения, а также разработка стандартизованных архитектур для промышленного применения в средах IIoT и беспроводных сенсорных сетях.

Литература

1. *John D. Kelleher*, Deep Learning. The MIT Press Essential Knowledge series, MIT Press, 2019.

2. *Simon J.D. Prince*, Understanding Deep Learning. MIT Press, 2023.

3. *Фатхулин Т. Д., Юдин А. Д.* Методики оптимизации загрузки изображений в web-приложениях // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 1. С. 105-110. EDN TXTWFG.

4. *Фатхулин Т. Д., Фатхулина Г. Г., Рахматова А. А.* Интеграция технологии больших языковых моделей в образовательный процесс высшей школы // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 2. С. 107-110. EDN FOGQPZ.

5. *Киреев А. А., Фатхулин Т. Д.* Анализ средств автоматизированного выбора конфигурации сети // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 1. С. 15-19. EDN ETSHKC.

6. *Фатхулин Т. Д., Чепенко К. А*. Анализ технологий обнаружения дефектов фасадов зданий /// Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2025. № 1. С. 78-82. EDN BYMEPU.

7. *Леохин Ю. Л., Фатхулин Т. Д., Кожанов М. С.* Анализ и исследование применения нейросетевых технологий для генерации программного кода // Вестник Рязанского государственного радиотехнического университета. 2024. № 87. С. 41-53. DOI 10.21667/1995-4565-2024-87-41-53. EDN HKEOFX.

8. *Леохин, Ю. Л., Фатхулин Т. Д., Ментус М. В.* Разработка и применение методов распознавания зашумленных аудиофайлов посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2024. № 88. С. 65-73. DOI 10.21667/1995-4565-2024-88-65-73. EDN NMXASI.

9. *Мяличева А. А., Фатхулин Т. Д.*Анализ методов машинного обучения для прогнозирования дефектов в исходном коде // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 16-19. EDN IVJCZF.

10. *Маслов К. В., Фатхулин Т. Д., Иванов Д. А.* Анализ технологий автоматизации бизнес-процессов и разработки программного обеспечения с использованием low-code платформ // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 1. С. 6-11. EDN HDBOYM.

11. *Фатхулин Т. Д., Исаев А. В*. Анализ моделей arima и lstm, используемых для прогнозирования криптовалют и определения портфеля инвестиций // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2024. № 2. С. 20-25. EDN ODWOPA.

12. *Леохин Ю. Л., Фатхулин Т. Д.* Разработка методов и алгоритма формализации текстового запроса к онлайн-сервисам, генерирующим изображения посредством нейросетевых технологий // Вестник Рязанского государственного радиотехнического университета. 2023. № 85. С. 82-95. DOI 10.21667/1995-4565-2023-85-82-95. EDN PZWYZV.

13. *Фатхулин Т. Д., Лушин Е. А.* Анализ развития автоматической генерации кода для web-сервисов // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2023. № 1. С. 128-132. EDN JUEGXP.

14. *Митрофанов А. О., Степанов М. Н., Фатхулин Т. Д.* Анализ нейросетевых методов генерации изображения по текстовому запросу // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2022. № 1. С. 19-23. EDN CWRLQA.

15. *Фатхулин Т. Д., Хорикова С. Г., Щитов В. М*. Анализ ключевых особенностей технологии программно-конфигурируемых оптических сетей (SDON) // Труды Северо-Кавказского филиала Московского технического университета связи и информатики. 2021. № 1. С. 29-34. EDN SMTDAF.

16. *Фатхулина Г. Г.* Разработка технологического уровня когнитивного обучения иностранному языку магистрантов гуманитарного вуза // Современные методы и технологии преподавания иностранных языков : Сборник научных статей XVI Международная научно-практическая конференция, Чебоксары, 17-18 октября 2019 года / Ответственные редакторы: Н.В. Кормилина, Н.Ю. Шугаева. Чебоксары: Чувашский государственный педагогический университет им. И.Я. Яковлева, 2019. С. 125-129. EDN BWPGLK.

17. *Фатхулина Г. Г.* Обучение иноязычному чтению на основе теории межкультурной коммуникации // Лингводидактические особенности обучения иностранным языкам в неязыковых вузах : Материалы II Международной научно-практической конференции, Москва, 25 апреля 2019 года. М.: Канцлер, 2019. С. 221-226. EDN VZHZPT.

18. *Фатхулина Г. Г.* Содержание обучения фонетическому аспекту английского языка в свете теории межкультурной коммуникации // Вопросы лингводидактики и межкультурной коммуникации в контексте современных исследований : сборник научных статей XI Международной научно-практической конференции, Чебоксары, 26 апреля 2019 года / отв. ред. Н. В. Кормилина, Н. Ю. Шугаева. Чебоксары: Чувашский государственный педагогический университет им. И.Я. Яковлева, 2019. – С. 372-376. EDN IVAMWG.

19. *Фатхулина Г. Г.* Роль глоссария в овладении студентами иноязычной лексикой // Вопросы лингводидактики и межкультурной коммуникации : Сборник научных статей, Чебоксары, 23-24 октября 2015 года; Отв. ред.: Н. В. Кормилина, Н. Ю. Шугаева. Чебоксары: Чувашский государственный педагогический университет им. И.Я. Яковлева, 2015. С. 193-197. EDN UYLLQR.

20. *Фатхулина Г. Г.* Развивающий потенциал современных технологий обучения иностранному языку в вузе // Высшее образование для XXI века : XII Международная научная конференция: Доклады и материалы. Круглый стол «Оптимизация преподавания иностранного языка в вузе», Москва, 03-05 декабря 2015 года / Отв. ред. С. Ф. Щербак. М.: Московский гуманитарный университет, 2015. С. 21-26. EDN VNBMGB.

21. *Фатхулина Г. Г.* Применение технологии активного слушания в преподавании иностранного языка студентам гуманитарного вуза // Актуальные проблемы лингводидактики и методики обучения иностранным языкам : сборник научных статей. Чебоксары: Чувашский государственный педагогический университет им. И.Я. Яковлева, 2015. С. 281-284. EDN TYAKQZ.

22. *Fatkhulina G. G.* A cognitive EFL teaching techniques for University students // Современное языковое образование: инновации, проблемы, решения : Сборник научных трудов. М.: Московский государственный гуманитарный университет им. М.А. Шолохова, 2014, pp. 157-162. EDN TDAXGD.

23. *Вишневский В. М., Леохин Ю. Л., Фатхулин Т. Д., Занегин А. В*. Методы машинного обучения в решении задачи прогнозирования спроса на отдельные виды товаров // T-Comm: Телекоммуникации и транспорт. 2024. Т. 18, № 10. С. 34-43. DOI 10.36724/2072-8735-2024-18-10-34-43. EDN COBEAG.

24. *Леохин Ю. Л., Фатхулин Т. Д., Маслов К. В.* Разработка методов системного анализа бизнес- процессов в банковской сфере для принятия решений о кредитовании различных организаций // Наукоемкие технологии в космических исследованиях Земли. 2025. Т. 17, № 5. С. 59-71. DOI 10.36724/2409-5419-2025-17-5-59-71. EDN VXBFTH.

25. *Леохин Ю. Л., Дымкова С. С., Фатхулин Т. Д.* Методы машинного обучения в прикладных задачах прогнозирования динамично изменяющихся данных // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19, № 8. С. 49-63. DOI 10.36724/2072-8735-2025-19-8-49-63. EDN ULVCHG.

26. *Leokhin Yu. L., Dymkova S. S., Fatkhulin T. D.* Research and development of image improvement tools // T-Comm: Телекоммуникации и транспорт. 2025. Vol. 19, No. 4, pp. 45-56. DOI 10.36724/2072-8735-2025-19-4-45-56. EDN FUINEN.

27. *Леохин Ю. Л., Дымкова С. С., Фатхулин Т. Д., Зозуля И. С.* Методы и алгоритмы интеллектуальной поддержки принятия управленческих решений в организационных системах торговых компаний // T-Comm: Телекоммуникации и транспорт. 2025. Т. 19, № 12. С. 44-50. DOI 10.36724/2072-8735-2025-19-12-44-50. EDN XXFTQJ.

28. *Леохин Ю. Л., Фатхулин Т. Д., Занегин А. В*. Модификация метода градиентного усиления для прогнозирования спроса на отдельные виды товаров // Наукоемкие технологии в космических исследованиях Земли. 2025. Т. 17, № 2. С. 32-41. DOI 10.36724/2409-5419-2025-17-2-32-41. EDN PNUPKY.