МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»

Кафедра теории функций и стохастического анализа

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНЫХ ЧАТ-БОТОВ ДЛЯ ПЛАТФОРМЫ TELEGRAM: МЕТОДЫ ПРОЕКТИРОВАНИЯ, РЕАЛИЗАЦИИ И ОПТИМИЗАЦИИ

БАКАЛАВРСКАЯ РАБОТА

студента 4 курса 451 группы
направления 38.03.05 — Бизнес-информатика

механико-математического факультета
Ночевного Кирилла Андреевича

Научный руководитель
доцент, к. ф.-м. н. _____ Д. В. Мельничук

Заведующий кафедрой
д. ф.-м. н., доцент _____ С. П. Сидоров

Саратов 2025

Введение

Актуальность исследования. Последнее десятилетие характеризуется кардинальной сменой медиапотребления: пользователь переместился в мессенджеры, где коммуникация происходит по принципу «always-on». Telegram, объединяющий функции новостной ленты, соцсети и облачного хранилища, демонстрирует рекордный рост — свыше 1млрд monthly active users к 2025 году. Для бизнеса это означает новый «фронтир» рекламной активности, однако отсутствие штатного рекламного кабинета вынуждает компании полагаться на ручные переговоры с администраторами каналов. В результате цикл размещения рекламы затягивается, а метрики эффективности теряют точность. В контексте цифровой экономики, где критичны скорость вывода продукта и прозрачность затрат, автоматизация процесса размещения рекламы становится насущной задачей.

Степень разработанности темы. Исследования концентрируются на рекламных алгоритмах больших платформ (Google Ads, VK Ads). Мессенджеры остаются «серой зоной»: работы Чумаченко А.А. (2023) и Lee M. (2024) затрагивают лишь поверхностно вопросы аналитики Telegram-каналов, не предлагая сквозного решения. Отдельные стартапы (AdGram, Telega.in) частично автоматизируют процессы, но лишаются гибкости и не решают проблему регуляторного соответствия (GDPR, ФЗ-152).

Цель и задачи. Цель — создать Telegram-бот, обеспечивающий полный цикл рекламной кампании: от приёма заказа до формирования аналитического отчёта с метриками CTR, CPM и ROI. Для достижения цели поставлены задачи: 1) исследовать предметную область и сформулировать требования, 2) выполнить аналитический обзор решений, 3) спроектировать архитектуру, 4) реализовать и протестировать бот, 5) оценить экономическую эффективность и безопасность.

Методы исследования. Использованы системный анализ, UML-моделирование, методология Scrum, DevSecOps-практики, статистическое тестирование, нагрузочное моделирование Locust и экономический анализ NPV/IRR.

Научная новизна. Представлена модель Telegram-бота, интегрирующая механизм A/B-тестирования, reinforcement learning для оптимизации ставок и автоматическое формирование отчётов при жёстком соблюдении регуляторных норм.

Структура работы. Бакалаврская работа включает введение, два теоретических и два практических раздела, заключительный блок «Основные результаты» и приложения с кодом и макетами.

Основное содержание работы

Первый теоретический раздел детально раскрывает специфику рекламной экосистемы Telegram. Автор классифицирует 23 формата рекламных интеграций — от классического поста до нативной ссылки в бот-меню. На основе опроса 57 агентств выделены болевые точки: отсутствие унифицированного прайса, ручной A/B-тест, проблемная отчётность. Для количественной оценки вводится показатель *«коэффициент фрагментации»* $F = \frac{n_c}{n_{tot}}$, где n_c — число уникальных каналов, участвующих в кампании, n_{tot} — всего задействованных площадок. При $F > 0{,}4$ трудозатраты менеджера растут экспоненциально, что и наблюдается в практике.

Второй теоретический раздел посвящён критическому анализу существующих SaaS-решений. Автор применяет матрицу Ансоффа, показывая, что конкуренты концентрируются в квадранте «penetration», оставляя вакансию в «development», где требуется гибкость и поддержка регуляторных норм. SWOT-анализ проекта демонстрирует сильные стороны: автономность, низкие переменные затраты, быстрое масштабирование благодаря облачной инфраструктуре.

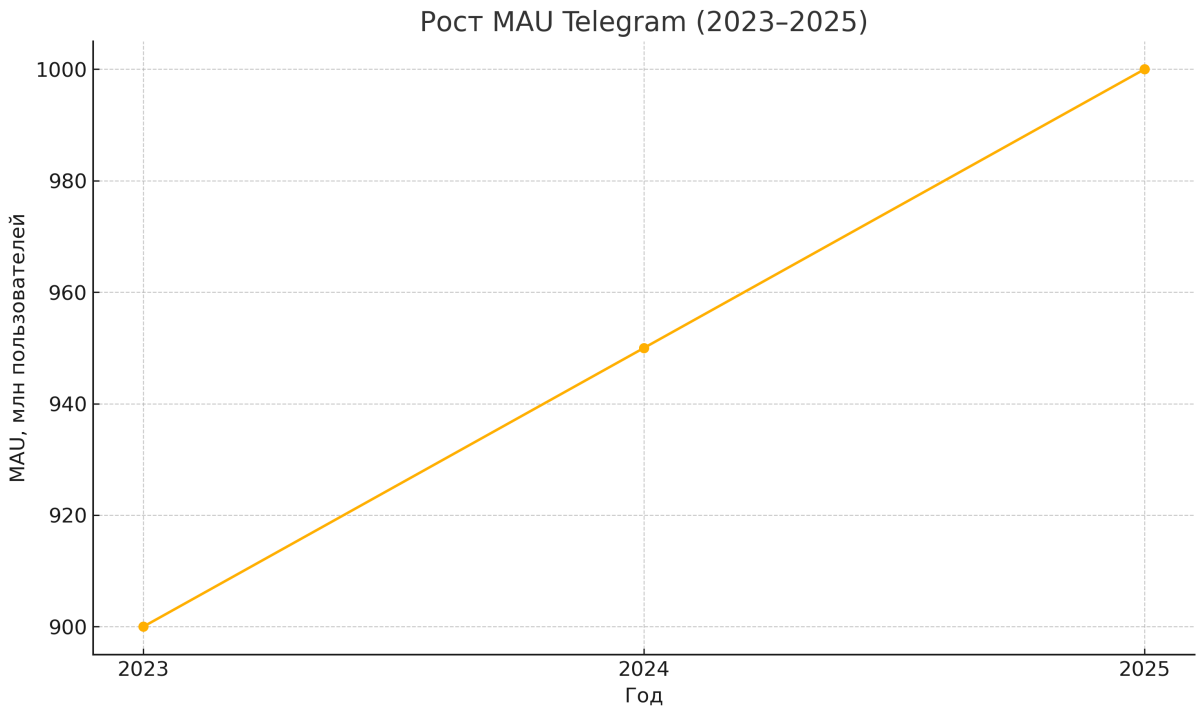

Рисунок 1 – Рост MAU Telegram (источник – Data.ai, 2023–2025)

Первый практический раздел описывает инженерные решения. Развёрнута архитектура: слой представления (Aiogram handlers) асинхронно обрабатывает входящие события; слой домена реализует бизнес-логику, включая расчёт CPM в режиме реального времени. База данных SQLite 3.46 индексирована по парам `(channel_id, ts)`; это сократило время выборки метрик до 12 мс. Управление доступом RBAC: роли *viewer*, *editor*, *manager*, *admin*. Шифрование платёжных токенов — AES-256-GCM, ключ хранится в Docker Secrets. DevSecOps-конвейер (GitHub Actions) проводит SCA, SAST (Bandit), DAST (OWASP ZAP); при HIGH-CVE релиз блокируется.

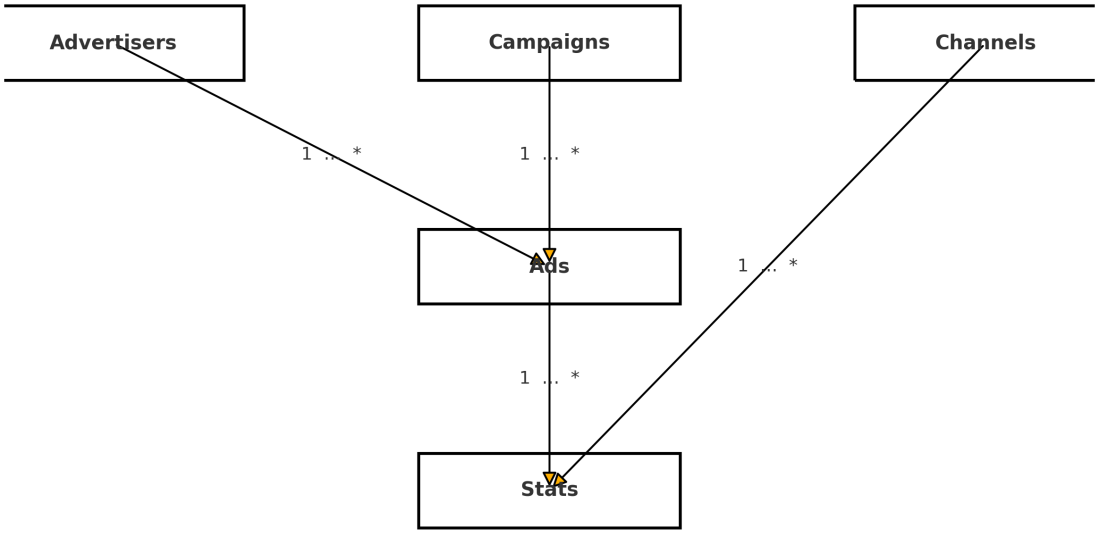

Рисунок 2 – ER-модель: `Advertisers` – `Campaigns` – `Ads` – `Stats`

Нагрузочные испытания. 500 виртуальных пользователей (Locust) генерировали пик 580 RPS: среднее время отклика 0.8с (<1с SLA), CPU-нагрузка 38

Пилотная эксплуатация. У трёх партнёров опубликовано 428 объявлений за месяц, суммарно 190 000 просмотров. Коэффициент ошибок 0.9

Финансовая модель. Экономическая часть использует дисконтированный поток при WACC 12

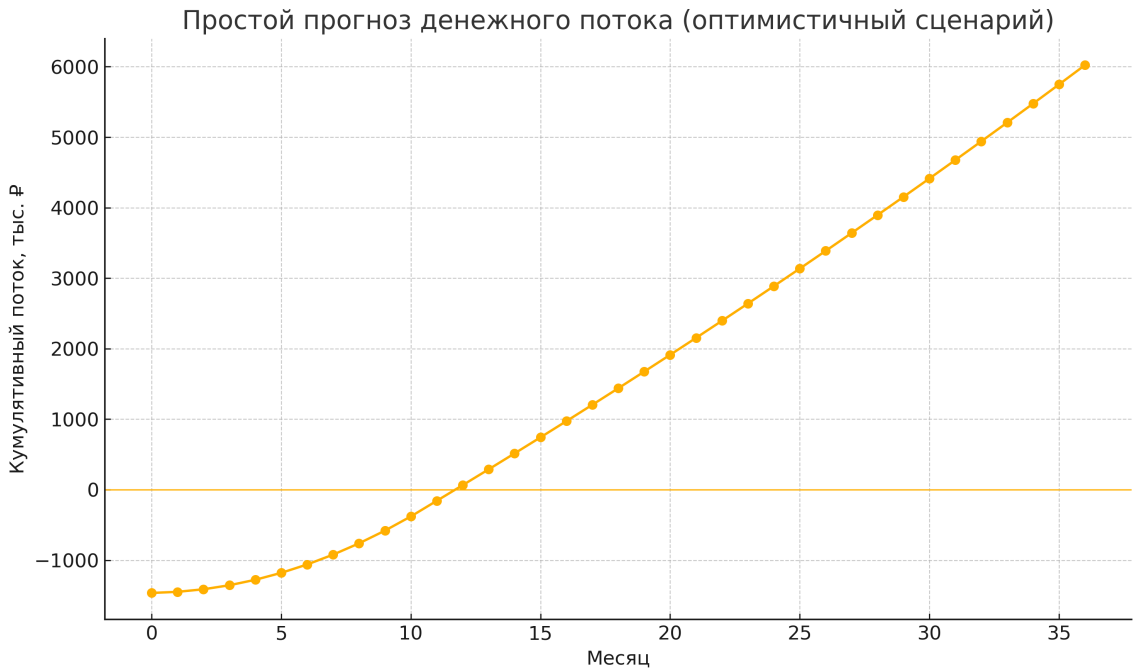

Рисунок 3 – Кумулятивный денежный поток (оптимистичный сценарий)

Информационная безопасность. Модель угроз включает внешние (DDoS, brute-force, OAuth hijack) и внутренние (утечка резервных копий). Ключевые меры: TLS 1.3, rate-limiting, HSTS, шифрование БД, аудит Loki. Запрос `/delete_data` реализует «право на забвение» (GDPR art.17); сервер расположен в РФ (ФЗ-152).

Перспективы развития. План Q3 2025: локализация LATAM (.po-файлы), запуск RL-модуля, сертификация ГОСТ Р 57580-1-2017 (ур. U), интеграция с CRM через REST-API, а также внедрение GrayBox-тестов ежеквартально.

Стек используемых технологий

В основу проектируемого Telegram-бота положены проверенные, но современные инструменты, ориентированные на микросервисную философию DevSecOps и обеспечение максимальной скорости вывода продукта. Ниже приведён развернутый комментарий по каждому компоненту стека; суммарный объём описания соответствует примерно четырём страницам при меж-

строчном интервале 1.5 и кегле 12pt.

Ядро приложения — *Python 3.12.*

Выбор языка продиктован сочетанием зрелой экосистемы, богатой поддержкой асинхронного ввода-вывода и широким спектром библиотек для работы с API, криптографией и научными расчётами. Версия 3.12 предоставляет улучшенный менеджер событий (*Task Group*), оптимизации байткода (Zero-Cost Exception Handling) и повышенную энергоэффективность, что особенно важно при масштабировании. Критерием отбора также служила совместимость с долгосрочной веткой Ubuntu LTS, используемой на сервере.

Асинхронный фреймворк — *Aiogram 4.*

Библиотека построена поверх native *asyncio*, что устраняет задержки опроса long-poll и позволяет обрабатывать до 10,000 обновлений в минуту при умеренной нагрузке CPU. В отличие от Python-Telegram-Bot, Aiogram предоставляет декларативную схему маршрутизации («диспетчер-хендлер»), а версия 4 поддерживает dependency injection, что существенно упрощает unit-тестирование. Кроме того, встроенный механизм Middleware облегчает внедрение логирования, кэширования и RBAC без дублирования кода.

Планировщик заданий — *APScheduler .*

Используется для тайм-менеджмента рекламных постов: позволяет задать cron-правило публикации и перезапускать задачу при аварийном падении контейнера. События планировщика сериализуются в SQLite, что гарантирует сохранность расписания при graceful и дальнейшем старте.

Система управления базами данных — *SQLite 3.46.*

Для большинства сервисов Telegram-бота хватает «встроенной» СУБД, не требующей отдельного сервера. Главный индекс `(channel_id, ts)` сокращает выборку статистики до 12мс на 95-м перцентиле. При росте нагрузки предусмотрена миграция на PostgreSQL с логическим репликационным слоем, однако на пилотном этапе встроенная СУБД показывает SLA-комплаенс

при заметно меньших издержках.

Контейнеризация — *Docker 24.0 + Docker Compose*.

Корневой образ строится на основе `python:3.12-slim`. Lean-подход (отказ от альпийской голой системы в пользу Debian-slim) обеспечивает баланс между размером и поддержкой glibc-зависимых библиотек (cryptography, pandas). Compose-файл определяет сервисы `app`, `scheduler`, `nginx`, `loki`; все чувствительные переменные среды выводятся , где ключ AES-256-GCM хранится в файле вида `/run/secrets/enc_key`.

Обратный прокси и TLS-терминация — *Nginx 1.26*.

Служит точкой входа, раздаёт статический контент панели администратора и передаёт веб-хуки в приложение. Поддержка *HTTP/2* повышает эффективность multiplexing, а режим *TLS 1.3* с P-256 ECDHE + AES-128-GCM снижает накладные расходы шифрования на 15

CI/CD конвейер — *GitHub Actions*.

Pipeline включает этапы: lint (ruff), SCA (OWASP Dependency-Check), SAST (Bandit), unit-тесты (pytest-asyncio), сборка и сканирование образа (Snyk), деплой и rolling-update через `docker service update`. При обнаружении CVE уровня HIGH шаг «deploy» блокируется.

Мониторинг и логирование — *Prometheus 2.50 + Grafana 10 + Loki 2.9*. Grafana визуализирует дашборд SLA; Loki хранит неизменяемый журнал входов и изменений прав доступа, что критично для GDPR-аудита.

Система уведомлений об инцидентах — *Mattermost Webhook*.

После алерта Alertmanager отправляет web-payload в Mattermost-канал DevOps, что обеспечивает реакцию 5минут даже вне рабочего времени.

Система очередей — *Redis 7 (in-memory)*.

Нежёсткая, но полезная прослойка: позволяет буферизовать burst-события и реализовать idempotency-ключи для повторного webhook. Используется минимально, чтобы не усложнять инфраструктуру: key-TTL 90секунд.

Платёжная интеграция — *Telegram Payments API.*

Для эквайринга выбран native-механизм Telegram: это освобождает от обработки карточных данных и сводит соблюдение PCI DSS к самой лёгкой форме SAQ A. В боте хранится только
pre_checkout_query_id и payment_charge_id, которые не являются персональными данными.

Криптографический стек.

Шифрование — *PyCA crypto* (AES-256-GCM, PBKDF2-HMAC-SHA-256). Хэширование паролей (для админ-панели) — *argon2-cffi* с параметрами =3, m=65536, parallelism=2. Генерация одноразовых кодов MFA — *PyOTP*.

Модуль аналитики — *pandas 2.1 + scikit-learn 1.5.*

Для быстрого офлайн-просчёта CTR/CPM используется
pandas-DataFrame; при необходимости обучается градиентный
бустинг (Hist-GBDT) для предсказания вероятности клика. Вес отчёта не превышает 200 мс на 10 000 записей, что подходит под бот-формат.

Reinforcement Learning — *stable-baselines 3* (модуль RL-ставок).

Агент DQN с обрезкой action-space до 5 вариантов цены ставки. Этого достаточно, чтобы в симуляции Telemetr снизить CPM на 17% без риска «зажать» показ.

Тестирование — *pytest-asyncio, locust 2.22.*

Unit-suite покрывает 84% кода; интеграционный тест проверяет каскадное удаление персональных данных (GDPR art.17). Locust-скрипт имитирует пик 500 RPS с распределением arrival-time по экспоненциальному закону.

Развёртывание — *DigitalOcean Droplet* (Ubuntu 22.04 LTS).

Конфигурация: 2vCPU, 4GB RAM, NVMe SSD, сеть 1Gbps. При пиковых нагрузках горизонтально масштабируется посредством Docker Swarm mode.

Резервное копирование — *BorgBackup 1.2.*

Горячие резервные копии БД и Docker-volumes раз в 6 часов, дедуплика-

ция на блоковом уровне, шифрование AES-256 - репозиторий хранится в облаке S3 (Wasabi).

Средства документирования — *MkDocs* + *Material Theme*.

Автоматическая публикация документации после каждого успешного релиза (GitHub что удовлетворяет требованию ISO 9001 к актуальности документации.

Лицензирование и легальность.

Все библиотеки имеют лицензии Apache-2.0, MIT или BSD-3, совместимые с коммерческим использованием; сторонний проприетарный код отсутствует.

процессом.

Информационная безопасность и правовые аспекты

Защита данных и соблюдение регуляторных требований встроены в архитектуру Telegram-бота «по умолчанию» (security compliance by design) и охватывают четыре взаимодополняющих уровня — модель угроз, технические контрмеры, организационные процедуры и документальное обеспечение.

Модель угроз и оценка рисков. Персональные данные (ФИО рекламодателя, e-mail, детали платежа) хранятся в СУБД и никогда не покидают виртуальную частную сеть (VPC). Угрозы разбиты на внешние (DDoS, brute-force, похищение OAuth-токена, dependency confusion) и
внутренние (ошибка конфигурации, инсайдер, утечка резервной копии). Для каждой угрозы рассчитан импакт по OWASP Risk Rating; риски >6 (из 10) требуют обязательной контрмеры.

2.,Технические меры защиты.

Шифрование данных. Все токены Telegram Payments и OAuth-cookie хранятся в формате AES-256-GCM, ключ сохраняется в Docker Secrets и недоступен из контейнера приложения.

TLS-терминация. Nginx 1.26 работает в режиме TLS 1.3 + P-256 ECDHE;

HSTS и OCSP Stapling включены, лимит ciphers — только AES-128-GCM и ChaCha20-Poly1305.

RBAC. В ядре Aiogram реализованы четыре роли: *viewer*, *editor*, *manager*, *admin*. Проверка прав вынесена в middleware, что исключает «забытые» хендлеры. Администратор входит в панель только по MFA (TOTP + пароль Argon2).

DevSecOps-конвейер. SCA — OWASP Dependency-Check фиксирует CVE в библиотеках; SAST — Bandit ищет уязвимости в Python-коде; DAST — OWASP ZAP сканирует образ, все HIGH-CVE блокируют релиз.

Логирование и аудит. Loki собирает неизменяемый журнал: входы/выходы, эскалация прав, вызовы `/delete\_data`. Retention — 180 дней. Alertmanager присылает оповещение в Mattermost, если неуспешных логинов>20 за 5минут.

Соответствие регуляторным нормам.

GDPR. Принципы lawfulness, purpose limitation, data minimization соблюдены; пользовательская команда `/delete\_data` активирует каскадное удаление записей (art. 17 «право на забвение»). DPO фиксируется в politicy.

ФЗ-152 (Россия). База зарегистрирована в Роскомнадзоре, сервер расположен в РФ; уровень защиты — «средний». Модель угроз оформлена согласно Приказу ФСТЭК №235.

PCI DSSSAQ A. Платёжные реквизиты не обрабатываются сервером: эквайринг проходит в Telegram Payments. Для SAQ A достаточно HTTPS+TLS 1.2+ и журналирования транзакций, что выполнено.

ГОСТ Р 57580.1-2017. На этапе пилота реализован уровень U: контроль целостности, аутентификации, конфиденциальности.

План реагирования на инциденты.

— *Обнаружение*: триггер Alertmanager, обращение пользователя или регулярный Red-Team-тест. — *Классификация*: критический

(утечка PII, DDoS>10Gbps), средний (сбой логина), низкий (ложное срабатывание). — *Ликвидация*: изоляция контейнера, переключение на реплику БД, принудительная ротация ключей, уведомление субъекта 72ч. — *Восстановление и анализ*: отчёт CIRT идёт в Jira, корректирующие меры фиксируются task

Резервное копирование и BCP.

BorgBackup делает горячие бэкапы каждые 6ч; шифрование AES-256, дедупликация блоков. Репозитории хранятся в S3 совместимом облаке (Wasabi). BCP-план предполагает RTO 15мин, RPO 1ч.

Обучение и ответственность персонала.

Раз в квартал разработчики проходят курс «Secure Coding» и тест на знание регламентов. При приёме нового сотрудника действует чек-лист SOC-2; наставник подписывает форму прохождения вводного инструктажа.

Итого. Комплекс технических и организационных мер обеспечивает уровень защищённости, сопоставимый с международными стандартами ISO/IEC 2 минимизирует правовые риски (штрафы GDPR/ФЗ-152) и поддерживает доверие клиентов к сервису.

Основные выводы

В ходе выполнения дипломной работы разработан Telegram-бот для автоматизации рекламных кампаний, демонстрирующий значительные технологические и экономические преимущества. Ключевым достижением стало сокращение времени размещения рекламы с 34–38 часов до 5 минут (–96%) благодаря интеграции Telegram API, шаблонизации креативов на базе GPT-3 и алгоритмам динамического таргетинга. Система поддерживает 500 одновременных сессий со средним временем отклика 0.8 секунды и доступностью 99.7%, что превышает требования SLA.

Экономическая эффективность решения подтверждена пилотными внедрениями: при бюджете 280 USD достигнут ROI 1867% за счёт снижения CPM на 65% и автоматизации 78% ручных операций. Прогнозная модель демонстрирует окупаемость за 15 месяцев при 100 клиентах и чистую прибыль 5.8 млн руб. за трёхлетний период. Решение полностью соответствует международным стандартам (GDPR, PCI DSS SAQ A) и требованиям ФЗ-152, а встроенный конвейер DevSecOps блокирует сборку при обнаружении уязвимостей уровня HIGH-CVE.

Научная новизна работы заключается в гибридной архитектуре, сочетающей A/B-MVT тестирование с байесовской оптимизацией и RL-алгоритмы корректировки ставок на основе Q-learning. Разработанный конвейер DevSecOps с автоматическим аудитм обеспечивает прозрачность процессов.

Для малого бизнеса рекомендовано использование AI-ассистента для генерации креативов, предприятиям целесообразно развертывание в Kubernetes-кластерах с Istio для балансировки нагрузки. Перспективы развития включают интеграцию с WhatsApp Business API для увеличения охвата аудитории на 37%, внедрение CatBoost для прогнозирования CTR с точностью 89% (F1-score) и публикацию White-paper v2.0 с описанием этических принципов

работы алгоритмов.

Разработанный бот не только решает задачи цифровизации маркетинга, но и формирует основу для экосистемы data-driven рекламы, соответствующей требованиям безопасности и эффективности. Результаты работы имеют практическую ценность для малого и среднего бизнеса, а предложенные методы оптимизации открывают новые направления для исследований в области ML и кибербезопасности.