Международный журнал информационных технологий и энергоэффективности

Том 11 Номер 6 (68)

2026

Международный журнал информационных технологий и энергоэффективности
Т.11 №6(68) часть 1 2026 г.

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

СОДЕРЖАНИЕ / CONTENT

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Вихляев А.С., Моисеев Ю.Б. Выбор и производительность локальных больших языковых моделей в задачах автоматической интерпретации психометрических данных// Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6(68) Ч.1 с. 4–10

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.89:159.9.072

ВЫБОР И ПРОИЗВОДИТЕЛЬНОСТЬ ЛОКАЛЬНЫХ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ В ЗАДАЧАХ АВТОМАТИЧЕСКОЙ ИНТЕРПРЕТАЦИИ ПСИХОМЕТРИЧЕСКИХ ДАННЫХ

[1]Вихляев А.С., [2]Моисеев Ю.Б.
ФГАОУ ВО "МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ" , Москва, Россия (107023, город Москва, Большая Семёновская ул., д. 38), e-mail: [1]privetkomandir@yandex.ru, [2]ybmn@rambler.ru

В работе систематизируются критерии выбора и производительностные характеристики локальных больших языковых моделей применительно к задаче автоматической интерпретации структурированных результатов психометрического тестирования. Задача требует одновременного выполнения нескольких ограничений: локальной обработки данных согласно требованиям российского законодательства о персональных данных, работы на типовом оборудовании рабочей станции, достаточного качества выходного текста. Рассмотрены классы открытых моделей семейства LLaMA, методы посттренировочной квантизации (GPTQ, AWQ, k-quants формата GGUF) и runtime-окружения локального инференса (llama.cpp, Transformers, vLLM). Приведены иллюстративные оценки производительности моделей размером 7–13 миллиардов параметров на CPU и потребительских GPU. Сформулированы рекомендации по выбору конфигурации в зависимости от доступного оборудования и требований к времени отклика.

Ключевые слова: Большие языковые модели; локальный инференс; квантизация; LLaMA; GGUF; защита персональных данных; психометрическая интерпретация; профессиональный отбор.

SELECTION AND PERFORMANCE OF LOCAL LARGE LANGUAGE MODELS IN AUTOMATED INTERPRETATION OF PSYCHOMETRIC DATA

[1]Vikhlyaev A.S., [2]Moiseev Yu.B.
MOSCOW POLYTECHNIC UNIVERSITY, Moscow, Russia (107023, Moscow, Bolshaya Semenovskaya ul., 38,), e-mail: [1]privetkomandir@yandex.ru, [2]ybmn@rambler.ru

The paper systematises the criteria for choosing local large language models and their performance characteristics in the task of automated interpretation of structured psychometric test results. The task imposes several constraints: local data processing as required by the Russian regulations on personal data, operation on conventional workstation hardware, and sufficient output text quality. The work reviews open model families based on LLaMA, post-training quantisation methods (GPTQ, AWQ, k-quants of the GGUF format), and local inference runtimes (llama.cpp, Transformers, vLLM). Indicative performance estimates are given for 7–13 billion parameter models on CPU and consumer-grade GPUs. Configuration recommendations are formulated as a function of the available hardware and the response-time requirements.

Keywords: Large language models; local inference; quantisation; LLaMA; GGUF; personal data protection; psychometric interpretation; professional selection.

Введение

Цифровизация процедур психологического отбора привела к росту объёма структурированных данных, поступающих от каждой методики тестирования: помимо итоговых баллов фиксируются временные метрики, профили выполнения, паттерны

переключения между заданиями. Объём этих данных делает ручную интерпретацию специалистом трудоёмкой, особенно в массовых процедурах врачебно-лётной экспертизы и кадрового отбора в операторские профессии.

Большие языковые модели на основе трансформерной архитектуры [8] позволяют обрабатывать структурированные числовые данные и порождать развёрнутый текст интерпретации. Прямое применение таких моделей через облачные API в задачах ведомственного отбора затруднено нормативно: Федеральный закон № 152-ФЗ предписывает локализацию обработки персональных данных [3]. Психодиагностические данные кандидатов на лётные и операторские должности подпадают под усиленные требования к защите, и приемлемой архитектурой служит локальная обработка, при которой и данные, и модель находятся в контуре заказчика.

Развитие открытых моделей семейства LLaMA [7] и методов посттренировочной квантизации [2, 5] сделало локальный инференс на типовом оборудовании технически возможным. Возникает инженерная задача выбора конфигурации модели и runtime-окружения, обеспечивающих приемлемое качество интерпретации и время отклика, совместимое с работой специалиста. Компромиссы между размером модели, уровнем квантизации, типом оборудования и качеством вывода имеют многомерный характер.

На основе анализа литературы поставлена цель работы: систематизировать критерии выбора локальной большой языковой модели для задачи автоматической интерпретации психометрических данных и оценить производительностные характеристики типовых конфигураций. Исходя из цели исследования решались следующие задачи: 1) проанализировать применимые классы открытых языковых моделей и оценить их пригодность для генерации развёрнутой психометрической интерпретации в условиях ведомственного применения; 2) сопоставить методы посттренировочной квантизации и runtime-окружения локального инференса в части соотношения качества и ресурсных требований; 3) сформулировать функциональные требования к программной системе автоматической интерпретации, согласованные с нормами защиты персональных данных и с практикой профотбора в операторские профессии; 4) привести иллюстративные оценки производительности типовых конфигураций «модель — квантизация — оборудование» и сформулировать рекомендации по их выбору.

1. Литературный обзор

Современные большие языковые модели опираются на трансформерную архитектуру, предложенную Vaswani с соавт. [8]. Механизм самовнимания позволяет обрабатывать последовательность токенов параллельно и улавливать зависимости произвольной длины, что сделало возможным масштабирование моделей до десятков и сотен миллиардов параметров. Открытые модели меньшего размера заполнили нишу локального применения: Touvron с соавт. описали семейство LLaMA с моделями от 7 до 65 миллиардов параметров и показали, что модель на 13 миллиардов параметров превосходит модели семейства GPT-3 размером 175 миллиардов на большинстве сопоставительных задач [7]. Для прикладных задач необязательно использовать самую крупную модель; компактная модель с грамотно подобранным обучающим корпусом обеспечивает сопоставимое качество при кратно меньших ресурсных требованиях.

Параллельно сформировался класс методов посттренировочной квантизации. Метод LLM.int8(), предложенный Dettmers с соавт., сократил объём памяти вдвое без значимой деградации качества за счёт сочетания векторно-блочной квантизации основных весов и сохранения в полной точности доли «выбросов» [2]. Frantar с соавт. показали, что метод GPTQ обеспечивает квантизацию до 3–4 бит на вес с минимальным ростом перплексии [6]. Lin с соавт. в работе о методе AWQ установили, что защиты заслуживает лишь около одного процента «значимых» весов, и для остальных допустимо понижение точности до четырёх бит [5]. Практическая применимость на потребительском оборудовании была обеспечена развитием runtime-окружений; проект llama.cpp ввёл формат GGUF и семейство k-квантов, оптимизированных для CPU и архитектуры Apple Silicon [9].

Психодиагностический контекст применения опирается на сложившуюся практику профессионального отбора в операторские профессии [1] и общие требования к интерпретации психометрических данных [4]. Цифровая среда тестирования сама по себе вносит дополнительную вариативность в исходные данные [10], что усиливает требования к качеству автоматической интерпретации. Отдельная задача возникает в специализированных сценариях профотбора лётного состава, в частности при оценке функционального состояния после пережитых экстремальных событий [11].

2. Выбор модели, квантизации и runtime-окружения

К рассматриваемой системе предъявляются четыре класса требований: локальность обработки персональных данных в соответствии с № 152-ФЗ [3]; совместимость с типовым оборудованием рабочей станции специалиста (CPU 8–16 ядер, оперативная память 16–32 ГБ, опционально потребительский графический процессор с 8–16 ГБ видеопамяти); качество выходного текста, достаточное для использования в составе экспертной процедуры; время отклика порядка 5–30 секунд на одного испытуемого. Перечисленные требования сводятся к инженерной задаче подбора модели, метода квантизации и runtime, при которых одновременно соблюдены границы по памяти, по времени отклика и по качеству вывода.

Размер модели образует первый параметр выбора. Модели на 1–3 миллиарда параметров обеспечивают высокую скорость инференса даже на CPU (порядка 30–50 токенов в секунду), но дают ограниченное качество на задачах структурированного следования инструкции. Модели на 7–8 миллиардов параметров характеризуются в инженерной практике как оптимум для локальных задач широкого профиля: они удерживают структуру длинного промпта, корректно обращаются с числовыми входными данными при условии явной разметки, выполняют инструкции по формату ответа; объём памяти для квантизованной версии лежит в диапазоне 4–8 ГБ. Модели на 13 миллиардов параметров дают заметный прирост качества на задачах многошагового рассуждения и при наличии потребительского графического процессора с 12–16 ГБ видеопамяти работают с приемлемой скоростью. Рассматриваются модели с инструкционным дообучением (instruction-tuned), а не базовые foundation-модели: последние обучены на задаче продолжения текста и на инструкции реагируют непредсказуемо.

Квантизация образует второй параметр выбора. Веса трансформерной модели не равноценны: малая часть весов доминирует в выходе, и именно их точность определяет качество модели в целом [2, 5]. На уровне 5 бит на вес (Q5_K_M в формате GGUF) деградация

качества относительно полной точности FP16 не превышает 1 % по перплексии; уровень 4 бита на вес (Q4_K_M) даёт сокращение памяти приблизительно в четыре раза при умеренной деградации и остаётся пригодным для большинства прикладных применений. Уровни ниже Q4 применимы только при жёстких ресурсных ограничениях и сопровождаются заметным ростом ошибок формата вывода. Выбор runtime-окружения замыкает связку: для рабочих станций без выделенной видеокарты и для Apple Silicon естественным выбором служит llama.cpp с форматом GGUF [9]; для потребительских NVIDIA-видеокарт 8–16 ГБ применимы Transformers с пакетами AutoGPTQ или AutoAWQ либо специализированный runtime ExLlamaV2; для серверного развёртывания применим сервер vLLM с механизмом PagedAttention.

3. Производительностные характеристики

Производительность локального инференса характеризуется тремя параметрами: скоростью генерации (токенов в секунду на этапе декодирования), временем до первого токена (обработка входного промпта) и объёмом потребляемой памяти. Сводные иллюстративные показатели для типовых конфигураций приведены в Таблице 1; числа отражают типичные значения на современном потребительском оборудовании и зависят от длины контекста, сборки runtime, версии драйверов, теплового режима оборудования.

Таблица 1 - Иллюстративная производительность локального инференса для моделей семейства LLaMA и совместимых архитектур

Размер модели	Квантизация	Оборудование	Память	Скорость, ток./с
3 млрд	Q4_K_M	CPU 8 ядер	≈ 2 ГБ	30–50
7–8 млрд	Q4_K_M	CPU 8 ядер	≈ 5 ГБ	8–12
7–8 млрд	Q4 (GPTQ/AWQ)	GPU 8–12 ГБ	≈ 5 ГБ	40–60
13 млрд	Q4_K_M	CPU 16 ядер	≈ 8 ГБ	4–7
13 млрд	Q4 (GPTQ/AWQ)	GPU 12–16 ГБ	≈ 9 ГБ	25–40
7–8 млрд	Q4_K_M	Apple M-серия	≈ 5 ГБ	25–40

Из приведённых данных следуют практические наблюдения. На CPU модели размером 7–8 миллиардов параметров с квантизацией Q4_K_M обеспечивают скорость генерации 8–12 токенов в секунду, что при типичной длине ответа 300–500 токенов даёт интерпретацию одного протокола за 30–60 секунд и укладывается в требование по времени отклика. Переход на потребительский графический процессор увеличивает скорость в 4–6 раз. Архитектура Apple Silicon с унифицированной памятью обеспечивает производительность, сопоставимую с дискретным графическим процессором, при существенно меньшем энергопотреблении.

4. Применение в задачах профотбора и ограничения

Прикладная задача автоматической интерпретации психометрических данных в профотборе операторов предполагает поддержку специалиста, а не его замену. Интерпретация, сгенерированная локальной языковой моделью, предъявляется специалисту в качестве заготовки и подлежит экспертной проверке перед включением в итоговое заключение. Соотнесение с требованиями российского законодательства о персональных данных при локальной архитектуре оказывается прямым: и данные, и средство обработки находятся в контуре заказчика, передачи внешним сервисам не происходит, состав обрабатываемых данных минимизируется до необходимого для интерпретации [3].

Один из практически значимых сценариев применения связан с задачами психологического сопровождения лётного состава, в особенности с регулярной переоценкой функционального состояния после пережитых экстремальных событий. В работе Ю.Б. Моисеева о психологических особенностях лётчиков, связанных с катапультированием, фиксируется, что после такого события у лётчика регулярно отмечаются изменения характеристик внимания, оперативной памяти и быстроты принятия решения [11]. В этой задаче автоматизация интерпретации даёт ощутимый выигрыш: специалист получает заготовку отчёта, требующую только верификации и экспертной коррекции.

Ограничения подхода: локальные модели среднего размера допускают фактические галлюцинации и пропуски значимых шкал, что требует обязательной экспертной проверки; воспроизводимость требует фиксации точных параметров генерации в протоколе обработки; обновление моделей и runtime-окружений предъявляет требования к версионированию и валидации новых сборок перед допуском в эксплуатацию.

Заключение

В работе систематизированы критерии выбора локальной большой языковой модели для задачи автоматической интерпретации психометрических данных и приведены иллюстративные оценки производительности типовых конфигураций. Показано, что задача допускает практическое решение на современном потребительском оборудовании при использовании инструкционно дообученных моделей размером 7–13 миллиардов параметров, посттренировочной квантизации до 4–5 бит на вес и выборе runtime-окружения сообразно доступному оборудованию: llama.cpp для CPU и Apple Silicon, Transformers или ExLlamaV2 для потребительских NVIDIA-видеокарт, vLLM для серверного развёртывания. Локальная архитектура удовлетворяет нормативным требованиям по защите персональных данных, изложенным в Федеральном законе № 152-ФЗ.

Работа имеет прикладное значение для проектирования программных систем психологического сопровождения профессиональной деятельности, в которых требуется сочетание автоматизации интерпретации, локальности обработки данных и работы в условиях ограниченных вычислительных ресурсов. Прикладной фокус задаётся практикой профессионального отбора в операторские профессии, в том числе в авиационной медицине и психологии. Сценарий регулярной переоценки функционального состояния лётного состава после пережитых экстремальных событий [11] относится к задачам, в которых соотношение качества и трудоёмкости интерпретации особенно чувствительно. Дальнейшая работа предполагает эмпирическую оценку качества интерпретаций, генерируемых конкретными моделями на размеченных корпусах.

Список литературы

1. Бодров В.А., Лукьянова Н.Ф. Психологический отбор лётчиков и космонавтов. — М.: Наука, 1984. — 264 с.
2. Dettmers T., Lewis M., Belkada Y., Zettlemoyer L. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale // Advances in Neural Information Processing Systems 35. — 2022. — pp. 30318–30332. — DOI: 10.48550/arXiv.2208.07339.
3. О персональных данных: Федеральный закон от 27.07.2006 № 152-ФЗ (с изм.) // Собрание законодательства Российской Федерации. — 2006. — № 31 (ч. 1). — Ст. 3451.
4. Бурлачук Л.Ф. Психодиагностика: Учебник для вузов. 2-е изд. — СПб.: Питер, 2017. — 384 с.
5. Lin J., Tang J., Tang H., Yang S., Dang X., Gan C., Han S. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration // Proceedings of Machine Learning and Systems 6 (MLSys 2024). — 2024. — DOI: 10.48550/arXiv.2306.00978.
6. Frantar E., Ashkboos S., Hoefler T., Alistarh D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers // Proceedings of the 11th International Conference on Learning Representations (ICLR 2023). — 2023. — DOI: 10.48550/arXiv.2210.17323.
7. Touvron H., Lavril T., Izacard G. et al. LLaMA: Open and Efficient Foundation Language Models. — 2023. — DOI: 10.48550/arXiv.2302.13971.
8. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. Attention Is All You Need // Advances in Neural Information Processing Systems 30. — 2017. — pp. 6000–6010. — DOI: 10.48550/arXiv.1706.03762.
9. Gerganov G. llama.cpp: LLM Inference in C/C++ [Электронный ресурс]. — URL: https://github.com/ggml-org/llama.cpp.
10. Mead A.D., Drasgow F. Equivalence of computerized and paper-and-pencil cognitive ability tests: A meta-analysis // Psychological Bulletin. — 1993. — Vol. 114, No 3. — pp. 449–458. — DOI: 10.1037/0033-2909.114.3.449.
11. Моисеев Ю.Б. Психологические особенности лётчиков, связанные с катапультированием // Военная авиационная психология: Сборник лекций / под ред. В.А. Пономаренко. — М.: Перо, 2021. — С. 241–261.

References

1. Bodrov V.A., Lukyanova N.F. Psychological Selection of Pilots and Cosmonauts. Moscow: Nauka, 1984, p.264
2. Dettmers T., Lewis M., Belkada Y., Zettlemoyer L. LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale // Advances in Neural Information Processing Systems 35. — 2022. — pp. 30318–30332. — DOI: 10.48550/arXiv.2208.07339.
3. On Personal Data: Federal Law of July 27, 2006, No. 152-FZ (as amended) // Collected Legislation of the Russian Federation. — 2006. — No. 31 (Part 1). — Art. 3451.
4. Burlachuk L.F. Psychodiagnostics: Textbook for Universities. 2nd ed. — St. Petersburg: Piter, 2017. — p.384
5. Lin J., Tang J., Tang H., Yang S., Dang X., Gan C., Han S. AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration // Proceedings of Machine Learning and Systems 6 (MLSys 2024). — 2024. — DOI: 10.48550/arXiv.2306.00978.

6. Frantar E., Ashkboos S., Hoefler T., Alistarh D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers // Proceedings of the 11th International Conference on Learning Representations (ICLR 2023). - 2023. - DOI: 10.48550/arXiv.2210.17323.
7. Touvron H., Lavril T., Izacard G. et al. LLaMA: Open and Efficient Foundation Language Models. - 2023. - DOI: 10.48550/arXiv.2302.13971.
8. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser L., Polosukhin I. Attention Is All You Need // Advances in Neural Information Processing Systems 30. - 2017. - pp. 6000–6010. — DOI: 10.48550/arXiv.1706.03762.
9. Gerganov G. llama.cpp: LLM Inference in C/C++ [Electronic resource]. — URL: https://github.com/ggml-org/llama.cpp.
10. Mead A.D., Drasgow F. Equivalence of computerized and paper-and-pencil ability cognitive tests: A meta-analysis // Psychological Bulletin. - 1993. - Vol. 114, No. 3. — pp. 449–458. — DOI: 10.1037/0033-2909.114.3.449.
11. Moiseev Yu.B. Psychological characteristics of pilots associated with ejection // Military aviation psychology: Collection of lectures / edited by V.A. Ponomarenko. — Moscow: Pero, 2021. — pp. 241–261.

Александров А.А. Методы и стратегии оптимизации производительности фронтенда в высоконагруженных системах электронной коммерции// Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6(68) Ч.1 с. 11–16

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.422.83

МЕТОДЫ И СТРАТЕГИИ ОПТИМИЗАЦИИ ПРОИЗВОДИТЕЛЬНОСТИ ФРОНТЕНДА В ВЫСОКОНАГРУЖЕННЫХ СИСТЕМАХ ЭЛЕКТРОННОЙ КОММЕРЦИИ

Александров А.А.
ФГБОУ ВО "ЧУВАШСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ И.Н. УЛЬЯНОВА", Чебоксары, Россия (428015, Чувашская Республика, город Чебоксары, Московский пр-кт, д.15), e-mail: artem-aleksandrov-0101@mail.ru

В статье рассматрива.ются методы и стратегии оптимизации производительности фронтенда в высоконагруженных системах электронной коммерции. Актуальность исследования обусловлена прямым влиянием скорости загрузки интерфейсов на ключевые бизнес-метрики. Проведен анализ архитектурных подходов, включая микро-фронтенды и различные стратегии рендеринга, а также сравнительная оценка производительности популярных фреймворков с примерами реализации типового компонента. Особое внимание уделено современным метрикам Core Web Vitals.

Ключевые слова: Производительность фронтенда, электронная коммерция, микро-фронтенды, стратегии рендеринга, Core Web Vitals.

METHODS AND STRATEGIES FOR FRONTEND PERFORMANCE OPTIMIZATION IN HIGH-LOAD E-COMMERCE SYSTEMS

Aleksandrov A.A.
I.N. ULIANOV CHUVASH STATE UNIVERSITY, Cheboksary, Russia (428015, Chuvash Republic, Cheboksary, Moskovsky prospekt, 15,), e-mail: artem-aleksandrov-0101@mail.ru

The article discusses methods and strategies for optimizing frontend performance in high-load e-commerce systems. The relevance of the research is driven by the direct impact of interface loading speed on key business metrics. The analysis covers architectural approaches, including micro-frontends and various rendering strategies, as well as a comparative evaluation of popular frameworks' performance with code examples of a typical component implementation. Special attention is paid to modern Core Web Vitals metrics.

Keywords: Frontend performance, e-commerce, micro-frontends, rendering strategies, Core Web Vitals.

Введение

Развитие электронной коммерции сопровождается ростом сложности пользовательских интерфейсов и требований к их производительности. Согласно исследованиям, задержка загрузки страницы даже на одну секунду может снижать конверсию на 7% [1]. Проблема усугубляется архитектурной сложностью современных фронтенд-систем: традиционные монолитные одностраничные приложения создают узкие места при развертывании и ограничивают масштабирование в периоды пиковых нагрузок. Кроме того, обилие фреймворков и библиотек при отсутствии систематизированных критериев их выбора приводит к тому, что разработчики часто руководствуются субъективными предпочтениями, а не объективными показателями производительности [2].

Целью настоящей статьи является систематизация и сравнительный анализ современных методов и стратегий оптимизации производительности фронтенда в высоконагруженных системах электронной коммерции, включая архитектурные подходы, стратегии рендеринга, особенности реализации на популярных фреймворках и метрики оценки эффективности.

Методы и исследования. Информация об применённых методах исследования.

Для достижения цели работы был использован комплекс теоретических методов научного познания. В первую очередь применён метод систематического обзора и критического анализа научной литературы, посвящённой проблематике производительности фронтенда в электронной коммерции. Это позволило выявить ключевые направления исследований, установить степень изученности проблемы и определить нерешённые вопросы. Также использовались сравнительно-сопоставительный метод для оценки различных архитектурных подходов (микро-фронтенды, стратегии рендеринга) и механизмов реактивности (виртуальный DOM, сигналы), а также метод контент-анализа для извлечения и систематизации данных из рецензируемых научных статей, диссертационных исследований и материалов конференций. Для иллюстрации теоретических выводов применялся метод выборочного рефакторинга — разработка и сравнение небольшого типового компонента (счётчика товаров в корзине) на разных фреймворках, что позволило наглядно продемонстрировать различия в подходах к управлению состоянием и обновлению интерфейса.

Целью настоящей статьи является систематизация и сравнительный анализ современных методов и стратегий оптимизации производительности фронтенда в высоконагруженных системах электронной коммерции, включая архитектурные подходы, стратегии рендеринга, особенности реализации на популярных фреймворках и метрики оценки эффективности.

Результаты оригинального авторского исследования.

Выбор стратегии рендеринга является фундаментальным решением, определяющим воспринимаемую производительность приложения. В современной электронной коммерции используются четыре основных подхода: клиентский рендеринг (CSR), при котором браузер получает минимальный HTML и всю логику на JavaScript, что обеспечивает высокую интерактивность после загрузки, но страдает от медленного первого отображения контента; серверный рендеринг (SSR), где HTML генерируется на сервере для каждого запроса, что дает быстрый первый отклик, но увеличивает время до интерактивности и нагрузку на сервер; статическая генерация (SSG), обеспечивающая максимальную скорость загрузки за счет предварительной сборки HTML, но непригодная для динамического контента с часто меняющимися ценами; и инкрементальная статическая регенерация (ISR) как гибридный подход, позволяющий обновлять статические страницы после сборки. Для высоконагруженных платформ оптимальным признается комбинированный подход: ISR для каталогов товаров и потоковый SSR для персонализированных страниц корзины и рекомендаций [4].

Микро-фронтенды представляют собой расширение микросервисной архитектуры на клиентскую часть, когда приложение разбивается на независимо разрабатываемые и развертываемые модули, соответствующие бизнес-возможностям: каталог, корзина, оформление заказа, личный кабинет. Юдин с соавторами отмечают, что основная цель такого

подхода — облегчить поддержку и разработку больших приложений, над которыми работают разные команды [3]. Исследования Артюхина и соавторов показывают, что среди паттернов интеграции микро-фронтендов модульная федерация (Module Federation) обеспечивает наилучшие показатели производительности по сравнению с Single-SPA, веб-компонентами и iframe [2]. Однако Кожо с соавторами предупреждают, что внедрение микро-фронтендов вводит дополнительную сложность в инфраструктуру и требует зрелых практик DevOps. В их кейсе сопоставимых результатов можно было достичь и с монолитным фронтендом, но микро-фронтендовая архитектура оказалась наиболее удобной благодаря возможности повторного использования инфраструктуры микросервисов [5].

Ключевым трендом последних лет является переход от виртуального DOM к системам реактивности на основе сигналов. Виртуальный DOM, используемый в React и Vue, создает легковесное представление реального DOM в памяти, вычисляет различия и применяет минимальные обновления. Недостатком является избыточная нагрузка: даже при изменении одного небольшого значения может перевычисляться целое дерево компонентов. Оучаиб в своем исследовании показывает, что сигналы, применяемые в Solid и внедряемые в новые версии Angular, обеспечивают точно-зернистую реактивность: компонент рендерится единожды, а сигналы напрямую подписывают DOM-узлы на изменения, обновляя только конкретный узел без перезапуска всего компонента. Это снижает накладные расходы и потребление памяти при росте сложности приложения [4].

Для иллюстрации различий в производительности рассмотрим реализацию типового компонента электронной коммерции — счетчика количества товара в корзине с отображением общей стоимости. В React с использованием хуков изменение количества вызовет повторный рендеринг всего компонента, как показано на Рисунке 1:

```
import React, { useState } from 'react';

function CartItem({ price }) {
  const [quantity, setQuantity] = useState(1);

  const increase = () => setQuantity(q => q + 1);
  const total = price * quantity;

  return (
    <div>
      <p>Цена: ${price}</p>
      <p>Количество: {quantity}</p>
      <button onClick={increase}>+</button>
      <p>Итого: ${total}</p>
    </div>
  );
}
```

Рисунок 1 - Фрагмент кода на React

В этом примере React после каждого нажатия кнопки выполняет диффинг виртуального DOM всего компонента, включая пересчет итоговой стоимости и всех дочерних элементов. В

Solid с использованием сигналов при изменении количества обновятся только текстовые узлы, как показано на Рисунке 2:

```
import { createSignal } from 'solid-js';

function CartItem({ price }) {
  const [quantity, setQuantity] = createSignal(1);

  const increase = () => setQuantity(q => q() + 1);
  const total = () => price * quantity();

  return (
    <div>
      <p>Цена: {price}</p>
      <p>Количество: <span>{quantity()}</span></p>
      <button onClick={increase}>+</button>
      <p>Итого: <span>{total()}</span></p>
    </div>
  );
}
```

Рисунок 2 - Фрагмент кода на Solid

Сам компонент не перерендеривается, и виртуальный DOM не создается. Vue 3 предлагает компромиссный вариант, используя комбинацию виртуального DOM на уровне компонентов и системы реактивности на основе прокси, как показано на Рисунке 3:

```
<template>
  <div>
    <p>Цена: {{ price }}</p>
    <p>Количество: {{ quantity }}</p>
    <button @click="increase">+</button>
    <p>Итого: {{ total }}</p>
  </div>
</template>

<script setup>
import { ref, computed } from 'vue';

const props = defineProps(['price']);
const quantity = ref(1);

const increase = () => quantity.value++;
const total = computed(() => props.price * quantity.value);
</script>
```

Рисунок 3 - Фрагмент кода на Vue 3

В результатах бенчмарков Оучаиба фиксируется преимущество фреймворков, реализующих точно-зернистую реактивность, по показателям требуемого объема памяти и по параметрам предсказуемости масштабирования производительности при усложнении

структуры приложения в сравнении с традиционной парадигмой виртуального DOM, что подтверждается данными в источнике [4].

В контексте высоконагруженных систем значимость приобретают методы оптимизации загрузки ресурсов, при этом разделение кода на чанки с последующей загрузкой по требованию выступает как ключевой механизм, обеспечивающий подгрузку кода страницы оформления заказа исключительно при переходе к соответствующему интерфейсу, применение современных форматов изображений WebP и AVIF в сочетании с адаптивной стратегией их загрузки рассматривается как средство сокращения объема передаваемых данных и повышения эффективности визуализации, интеллектуальное кэширование ответов API посредством сервис-воркеров выступает как обеспечение работоспособности приложения при потере сетевого соединения и как фактор снижения сетевой нагрузки, оптимизация критического пути рендеринга, включающая вынос блокирующего JavaScript и CSS из критического потока, фиксируется как мероприятие, существенно ускоряющее первоначальный отклик страницы [5].

Оценка производительности в современном электронном бизнесе осуществляется на основе метрик Core Web Vitals, влияние которых на поведенческие показатели пользователей [1], при этом показатель Largest Contentful Paint интерпретируется как мера времени загрузки основного контента и подлежит ограничению менее чем 2,5 секунд, показатель Interaction to Next Paint рассматривается как характеристика времени отклика на взаимодействие пользователя и подлежит ограничению менее чем 100 миллисекунд, показатель Cumulative Layout Shift интерпретируется как индикатор визуальной стабильности и подлежит поддержанию значения менее 0,1 в целях предотвращения случайных кликов, причем оптимизация перечисленных метрик рассматривается как фактор повышения конверсии, увеличения уровня лояльности пользователей и укрепления репутации бренда, что отражено в источнике [1].

Заключение.

В результате осуществления анализа формулируются следующие выводы. Производительность фронтенда рассматривается как стратегический фактор успеха в электронной коммерции, обусловленный влиянием оптимизации клиентской части на метрики Core Web Vitals и через них на показатели конверсии и удержания пользователей. Архитектурный выбор детерминируется бизнес-потребностями, в рамках которых для крупных торговых площадок микро-фронтенды обеспечивают независимость разработки и масштабируемость, тогда как для проектов меньшего размера монолитная архитектура представляется более прагматичным решением. Фиксируется смещение парадигмы реактивности в сторону систем, основанных на сигналах, которые демонстрируют улучшение показателей производительности по сравнению с традиционным виртуальным DOM. Оптимизация загрузки ресурсов понимается как комплексный процесс, требующий объединения подходов к рендерингу, стратегическому разделению кода и современным методам работы с изображениями. Выбор фреймворка подлежит обоснованию на основании предполагаемых сценариев нагрузки и сложности интерфейса, а не исключительно на основании популярности технологии, что обеспечивает выверенность технического решения применительно к бизнес-целям. Перспективные направления исследований интерпретируются как применение машинного обучения для предиктивной оптимизации

интерфейса и интеграция edge-вычислений для обеспечения персонализированного рендеринга.

Список литературы

1. Шардаков Е. А. Совершенствование клиентского пути на веб-сайте интернет-провайдера для повышения конверсии и вовлеченности пользователей: магистерская диссертация : дис. – б. и., 2025.
2. Юдин А. В., Макиевский С. Е., Адышкин С. С., Грошева П. Ю. Анализ производительности и особенностей функционирования микрофронтендов // Computational Nanotechnology. 2024. Т. 11, № 1. С. 25–35.
3. Andi K., Ramirez D., Chango Sailema W. G. Comparativa de frameworks más usados de JavaScript mediante la creación de una aplicación web // Mikarimin. Revista Científica Multidisciplinaria. 2024. Vol. 10, № 3. pp. 81–100.
4. Ouchaib J. Benchmarking Modern Frontend Frameworks: A Comparative Analysis of Rendering Performance: Master‘s thesis. Turin: Politecnico di Torino, 2025. p. 72
5. Kojo R. H. H., Corte Real L. F., Ferreira R. C., Rosa T. O., Goldman A. Exploring Micro Frontends: A Case Study Application in E-Commerce // arXiv preprint arXiv:2506.21297. 2025.

References

1. Shardakov E. A. Improving the Customer Journey on an Internet Provider's Website to Increase Conversion and User Engagement: Master's Thesis: Dis. – b. i., 2025.
2. Yudin A. V., Makievsky S. E., Adyshkin S. S., Grosheva P. Yu. Analysis of the Performance and Functioning Features of Microfrontends // Computational Nanotechnology. 2024. Vol. 11, No. 1. pp. 25–35.
3. Andi K., Ramirez D., Chango Sailema W. G. Comparison of JavaScript frameworks used in the creation of a web application // Mikarimin. Revista Científica Multidisciplinaria. 2024. Vol. 10, No. 3. pp. 81–100.
4. Ouchaib J. Benchmarking Modern Frontend Frameworks: A Comparative Analysis of Rendering Performance: Master’s thesis. Turin: Politecnico di Torino, 2025. p.72
5. Kojo R. H. H., Corte Real L. F., Ferreira R. C., Rosa T. O., Goldman A. Exploring Micro Frontends: A Case Study Application in E-Commerce // arXiv preprint arXiv:2506.21297. 2025.

Гасанян Д.Ф. Автоматическое гибридное масштабирование микросервисов в KUBERNETES с учетом качества обслуживания на основе прогнозирования нагрузки и обучения с подкреплением// Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч.1 с. 17–29

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.75:004.85

АВТОМАТИЧЕСКОЕ ГИБРИДНОЕ МАСШТАБИРОВАНИЕ МИКРОСЕРВИСОВ В KUBERNETES С УЧЕТОМ КАЧЕСТВА ОБСЛУЖИВАНИЯ НА ОСНОВЕ ПРОГНОЗИРОВАНИЯ НАГРУЗКИ И ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

Гасанян Д.Ф.
ФГБОУ ВО «МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ», Москва, Россия (119454, г. Москва, пр-т Вернадского, д. 78, стр. 4), e-mail: diwork1721@gmail.com

Предложен метод автоматического гибридного масштабирования микросервисов в Kubernetes, ориентированный на качество обслуживания, а не только на загрузку процессора. Научный вклад состоит в совместной постановке трех компонентов: краткосрочного прогноза нагрузки с оценкой верхнего квантиля, прогнозного риска нарушения задержки 95-го процентиля (p95) и агента обучения с подкреплением, который выбирает изменение числа подов и ресурсного профиля пода. В качестве прогнозной модели используется линейное экспоненциальное сглаживание Хольта с квантильной поправкой по остаткам. Эта модель выбрана из-за низкой вычислительной стоимости, устойчивости на коротком горизонте и возможности явно оценить риск перегрузки. Для выбора действий применено табличное Q-обучение с маской безопасных действий. Маска запрещает шаги, нарушающие границы реплик и ресурсов, а при прогнозируемом превышении порога оставляет только действия, увеличивающие емкость узкого микросервиса. Метод проверен в воспроизводимой имитационной среде из трех связанных микросервисов. На 24 тестовых трассах предложенный контроллер снизил долю интервалов с нарушением целевого уровня обслуживания с 3,85% у реактивного горизонтального автомасштабировщика до 2,67%. По сравнению с вариантом без прогнозного риска доля нарушений уменьшилась с 7,53% до 2,67%, что показывает роль прогнозной части состояния.

Ключевые слова: Kubernetes, микросервисы, гибридное масштабирование, качество обслуживания, прогноз нагрузки, Q-обучение, горизонтальное масштабирование, ресурсный профиль.

AUTOMATIC HYBRID SCALE-UP OF MICROSERVICES IN KUBERNETS WITH QUALITY OF SERVICE BASED ON LOAD FORECASTING AND REINFORCEMENT LEARNING

Gasanyan D.F.
MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY, Moscow, Russia (119454, Moscow, avenue Vernadsky, 78, b. 4), e-mail: diwork1721@gmail.com

The paper proposes an automatic hybrid autoscaling method for Kubernetes microservices that targets service quality rather than processor utilization alone. The contribution combines three elements: short-term workload forecasting with an upper-quantile estimate, predictive risk of 95th-percentile latency violation, and a reinforcement learning (RL) agent that selects replica and per-pod resource-profile changes. The forecasting component uses Holt linear exponential smoothing with a residual-based quantile correction. This model is selected because it is inexpensive at inference time, stable on a short control horizon, and suitable for explicit overload-risk estimation. The decision component uses tabular Q-learning with a safe action mask. The mask blocks actions outside replica and resource bounds and, under predicted risk, leaves only actions that increase the capacity of the bottleneck microservice. The method is evaluated in a reproducible simulation of three interacting microservices. On 24 unseen workload traces, the proposed controller reduces service-level violation intervals from 3.85% under reactive HPA to 2.67%. Compared with the same learned policy without predictive risk, violations decrease from 7.53% to 2.67%, indicating the role of forecasting in the controller state.

Keywords: Kubernetes, microservices, hybrid autoscaling, quality of service, workload forecasting, Q-learning, horizontal scaling, resource profile.

Введение

Микросервисное приложение в Kubernetes редко перегружается как один неделимый процесс. Пользовательский запрос проходит через входной шлюз, сервис каталога, сервис заказа, базу данных и внешние зависимости. Поэтому рост задержки может возникнуть не в том поде, где выше всего загрузка процессора. Стандартный горизонтальный автомасштабировщик подов Kubernetes, далее HPA, масштабирует выбранный объект по отношению текущей метрики к целевой [1]. Это простая и надежная основа, но она остается реактивной.

Вертикальное масштабирование дополняет горизонтальное, поскольку меняет запросы и ограничения ресурсов для уже существующих подов [2]. Однако в Kubernetes изменение ресурсного профиля может требовать пересоздания пода или отдельной поддержки изменения ресурсов без перезапуска. Для микросервисов с жестким ограничением задержки такая операция должна выполняться через слой безопасности, иначе выигрыш от увеличения ресурса может быть потерян на этапе обновления.

Исследования показывают несколько важных ограничений существующих подходов. Nguyen и соавторы экспериментально описали запаздывание HPA и чувствительность к частоте метрик [3]. Rattihalli и соавторы показали, что вертикальная корректировка ресурсов важна для снижения избыточного выделения [4]. Baresi и соавторы рассмотрели конфликт горизонтального и вертикального контуров управления [5]. Vu, Tran и Kim предложили прогнозное гибридное масштабирование, но их схема опирается на отдельный детектор всплесков и регрессионную модель производительности [6]. Работы Mondal и соавторов подтверждают, что качество прогноза зависит от типа нагрузки и окна наблюдения [7].

Обучение с подкреплением применяется в этой области потому, что действие масштабирования имеет отложенную цену. Добавление подов снижает риск задержки, но увеличивает стоимость. Уменьшение ресурсов экономит кластер, но может проявиться нарушением p95 через несколько интервалов. Rossi и соавторы, Abdel Khaleq и Ra, Xu и соавторы, а также Santos и соавторы показывают разные варианты такой постановки [8; 9; 10; 13]. При этом часть работ либо выбирает только метрику для HPA, либо переносит обучение на сложное пространство действий без явного защитного слоя.

В этой статье предлагается прогнозно-рисковый гибридный контроллер. Он не сводится к замене штатного автомасштабировщика обучаемой моделью. Контроллер вводит риск нарушения качества обслуживания как самостоятельный признак состояния, связывает прогноз нагрузки с графом микросервисов, применяет Q-обучение только внутри допустимой области действий и отделяет решение агента от слоя исполнения Kubernetes.

Цель

Цель исследования состоит в разработке архитектуры и алгоритма гибридного автомасштабирования микросервисов в Kubernetes, который заранее учитывает риск нарушения p95-задержки и выбирает между горизонтальным изменением числа подов и изменением ресурсного профиля пода.

Исследовательская часть работы выражена в четырех положениях. Во-первых, предложена формализация прогнозного риска для маршрута микросервисов, где риск зависит

от верхнего квантиля будущей нагрузки, емкости каждого сервиса и целевого порога задержки. Во-вторых, состояние агента обучения с подкреплением включает не только текущие метрики, но и прогнозный риск, направление нагрузки и идентификатор узкого микросервиса. В-третьих, действие агента ограничено маской безопасности, которая отделяет обучение от недопустимых операций Kubernetes. В-четвертых, функция награды объединяет нарушение качества обслуживания, ресурсную стоимость и устойчивость управления.

Гипотеза исследования формулируется так: если агент получает не только текущую загрузку, но и прогнозный риск нарушения p95 на маршруте микросервисов, то он выбирает более устойчивые гибридные действия, чем реактивное масштабирование по текущей метрике.

Материал и методы исследования

Материал исследования включает официальную документацию Kubernetes, статьи о горизонтальном и вертикальном масштабировании, работы о прогнозном масштабировании и публикации по применению обучения с подкреплением для управления микросервисами [1-13]. Для прогнозирования использована модель Хольта как базовый метод краткосрочного прогноза временных рядов с уровнем и трендом [14]. Математическая основа Q-обучения опирается на классическое описание Sutton и Barto, а также на исходную работу Watkins и Dayan [15; 16].

Базовая формула HPA задает желаемое число реплик через отношение текущего значения метрики к целевому. В обозначениях этой статьи она записана следующим образом.

$$N_{i,t+1}^{HPA} = \left\lceil N_{i,t} \frac{m_{i,t}}{m_i^*} \right\rceil \quad (1)$$

где $N_{i,t}$ — текущее число подов микросервиса i;

$m_{\mathrm{i,\,t}}$ — текущее значение метрики управления;

m_i^* — целевое значение метрики.

Формула (1) удобна для эксплуатации, но она не содержит будущей нагрузки, графа зависимостей и стоимости смены ресурсного профиля. Поэтому в предлагаемом методе HPA используется как базовая точка сравнения, а не как единственный контур управления.

Архитектура предлагаемого контроллера показана на рисунке 1. Система получает метрики из входного шлюза, Prometheus и программного интерфейса Kubernetes. В отдельном хранилище задается граф микросервисов: вершины соответствуют сервисам, ребра отражают маршрут запроса и коэффициент передачи нагрузки. Прогнозная модель оценивает нагрузку следующего интервала и верхний квантиль 0,90. Блок риска переводит прогноз в оценку p95-задержки по маршруту. Агент Q-обучения выбирает действие, но слой маски действий проверяет границы реплик, ресурсных профилей и паузу между изменениями.

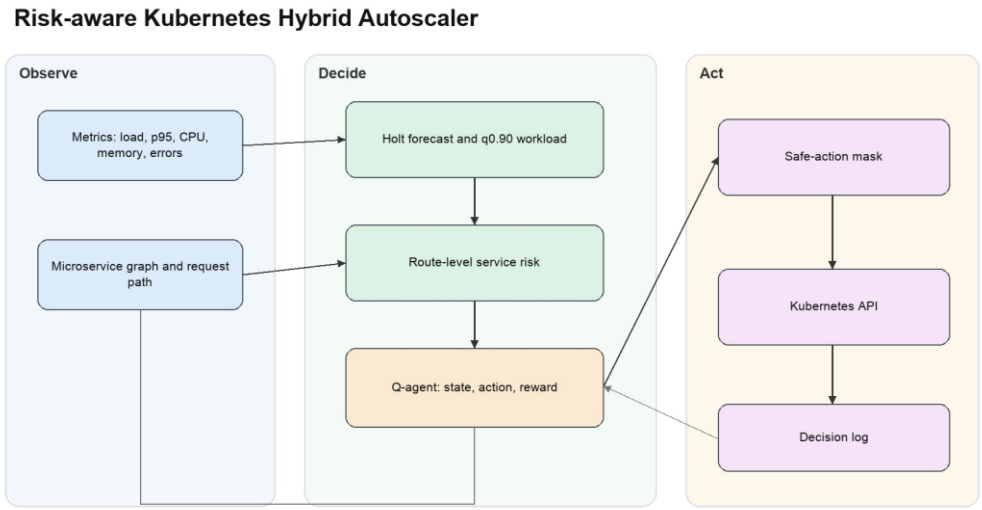

Рисунок 1 — Архитектура прогнозно-рискового гибридного автомасштабировщика

В качестве прогнозной модели используется линейное экспоненциальное сглаживание Хольта. Эта модель не требует большой обучающей выборки, имеет малую стоимость выполнения в контроллере и подходит для короткого горизонта, где соседние значения нагрузки часто информативны. При этом модель сохраняет тренд, что важно перед плавным ростом трафика. Уравнения обновления уровня и тренда приведены в формулах (2) и (3).

$$\ell_t = \alpha\lambda_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$$
$$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \quad (3)$$

где λ_t — наблюдаемая входная нагрузка в момент t;

ℓ_t — сглаженный уровень ряда;

b_t — оценка локального тренда;

α, β — коэффициенты сглаживания.

$$\hat{\lambda}_{t+h} = \ell_t + h\, b_t \quad (4)$$
$$\hat{\lambda}^{0.90}_{t+h} = \hat{\lambda}_{t+h} + z_{0.90}\hat{\sigma}_t,\ z_{0.90} = 1.2816 \quad (5)$$

где $\hat{\lambda}_{t+h}$ — точечный прогноз нагрузки на горизонт h;

$\hat{\lambda}^{0.90}_{t+h}$ — прогноз верхнего квантиля;

$z_{0.90}$ — квантиль стандартного нормального распределения для уровня 0.90;

$\hat{\sigma}_t$ — экспоненциально сглаженная ошибка прогноза.

Точечный прогноз и верхний квантиль нагрузки задаются формулами (4) и (5). Верхний квантиль используется не как попытка угадать среднюю нагрузку, а как инженерная оценка риска. Контроллер должен готовиться к правому хвосту нагрузки, потому что именно он чаще приводит к нарушению p95.

Для микросервиса i емкость задается числом подов и ресурсным профилем пода. Если $r_{i,t}$ обозначает выделенный ресурс на под, а $\mu_i(r_{i,t})$ означает обслуживающую способность одного пода, то прогнозная утилизация определяется формулой (6).

$$\rho_{i,t+h} = \frac{\omega_i\, \hat{\lambda}^{0.90}_{t+h}}{(N_{i,t}\, \mu_i(r_{i,t}) + \varepsilon)} \quad (6)$$

где $\rho_{i,t+h}$ — прогнозная утилизация микросервиса i;

ω_i — коэффициент передачи входной нагрузки в микросервис i;

ε — малое число для численной устойчивости.

$$\hat{L}^{95}_{i,t+h} = B_i + \frac{K_i \rho^2_{i,t+h}}{1.03 - \rho_{i,t+h}} + P_i\left[\rho_{i,t+h} - 1\right]_+ \quad (7)$$

где $\hat{L}^{95}_{i,t+h}$ — прогноз p95-задержки микросервиса;

B_i, K_i, P_i — параметры профиля микросервиса;

$[x]_+ = \max(x, 0)$ — положительная часть величины.

Формула (7) не претендует на универсальную модель очереди. Она нужна как воспроизводимая среда для обучения и проверки алгоритма: при приближении утилизации к насыщению задержка растет нелинейно, а при перегрузке добавляется отдельный штраф.

$$g_t = max_{\pi \in \Pi}\left(\frac{\sum_{i \in \pi} \hat{L}^{95}_{i,t+1}}{\tau_\pi}\right)$$

где g_t — прогнозный риск нарушения качества обслуживания;

Π — множество пользовательских маршрутов в графе микросервисов;

τ_π — целевой порог p95-задержки для маршрута π.

Итоговый риск маршрута определяется формулой (8). Состояние агента обучения с подкреплением строится из дискретизированных признаков (9). Дискретизация выбрана намеренно: она уменьшает размерность задачи и делает поведение политики проверяемым. В данной работе применяется табличное Q-обучение, а не глубокая нейронная модель. Такой выбор оправдан тем, что исследуется архитектурная постановка и действие имеет конечный набор вариантов.

$$s_t = \left(g_t, \Delta\hat{\lambda}_t,\ i_t^{max}, \frac{L_t^{95}}{\tau}, C_t,\ r_{i,t}^{max}\right) \quad (9)$$

где $\Delta\hat{\lambda}_t$ — направление прогнозируемого изменения нагрузки;

i_t^{max}=arg $max_i \rho_{i,t+1}$ — прогнозируемый узкий микросервис;

$C_t = \sum_i N_{i,t} r_{i,t}$ — текущая ресурсная стоимость.

$$a_t = (i_t, \Delta N_{i,t}, \Delta r_{i,t}),$$
$$\Delta N_{i,t} \in \{-1,0,1,2\}, \qquad \Delta r_{i,t} \in \{-1,0,1\} \quad (10)$$

где a_t — действие агента;

i_t — выбранный микросервис;

$\Delta N_{i,t}$ — изменение числа подов;

$\Delta r_{i,t}$ — переход между ресурсными профилями.

Действие агента задается формулой (10). Перед исполнением действие проходит через маску безопасности. Если прогнозный риск не превышает порог, маска оставляет все допустимые действия в пределах минимального и максимального числа подов. Если риск высок, маска оставляет только действия, которые увеличивают емкость прогнозируемого узкого микросервиса. Ограничения безопасности задаются в формуле (11).

$$A_{safe}(s_t) = \{a \in A: N^{min}{}_i \le N_i + \Delta N_i \le N^{max}{}_i, \quad r^{min}{}_i \le r_i + \Delta r_i \le r^{max}{}_i\} \tag{11}$$

где $A_{safe}(s_t)$ — множество действий, разрешенных слоем безопасности в состоянии s_t;

A — исходное множество действий агента.

Алгоритм одного шага контроллера имеет следующую логику. Сначала собираются метрики текущего интервала и обновляется прогноз нагрузки. Затем вычисляется прогнозная задержка каждого сервиса и риск маршрута. После этого формируется множество безопасных действий, агент выбирает действие с максимальным Q-значением внутри этого множества, а слой исполнения применяет изменение числа подов или ресурсного профиля через программный интерфейс Kubernetes. Журнал решений возвращается в контур наблюдения и используется при следующем шаге.

Функция награды имеет вид (12).

$$R_t = -w_q \left[\frac{L^{95}{}_t}{\tau} - 1\right]_+ - w_c C_t - w_s S_t - w_u U_t + B_t \tag{12}$$

где R_t — награда агента;

(w_q, w_c, w_s, w_u) — неотрицательные веса штрафов;

S_t — цена изменения действия и ресурсного профиля;

U_t — штраф за перегрузку или слишком низкую утилизацию;

B_t — малый бонус за работу без нарушения качества в допустимой зоне утилизации.

Обновление Q-значения выполняется по формуле (13).

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_q \left(R_t + \gamma \max_{a' \in A_{safe}(s_{t+1})} Q(s_{t+1}, a') - Q(s_t, a_t) \right) \tag{13}$$

где $Q(s_t, a_t)$ — оценка полезности действия в состоянии;

α_q — скорость обучения;

a' — возможное действие на следующем шаге;

$A_{safe}(s_{t+1})$ — множество безопасных действий в следующем состоянии;

γ — коэффициент учета будущей награды.

Обучение проводилось в имитационной среде. Среда содержит три микросервиса: входной шлюз, каталог и сервис заказа. Для каждого сервиса заданы коэффициент передачи нагрузки, базовая задержка, коэффициент очереди, обслуживающая способность одного ядра процессора, минимальное и максимальное число подов. Трассы нагрузки включали синусоидальные компоненты, тренд, нормальный шум и четыре всплеска разной ширины. Это не замена производственного стенда, а контролируемый способ проверить саму постановку алгоритма до внедрения в реальный кластер.

Таблица 1 - Параметры обучения и проверки

Параметр	Значение
Прогноз нагрузки	линейное экспоненциальное сглаживание Хольта, горизонт 1 минута, квантиль 0,90 по остаткам
Алгоритм обучения	табличное Q-обучение с маской безопасных действий
Обучение	1200 эпизодов по 360 минут
Тестовая проверка	24 новые трассы по 500 минут
Микросервисы	шлюз, каталог, заказ
Ресурсные профили	0,50, 0,75 и 1,00 ядра процессора на под
Целевой уровень	p95-задержка маршрута не выше 380 мс
Сравниваемые контроллеры	реактивный HPA, прогнозный HPA, предложенный контроллер, абляция без прогнозного риска

Результаты исследования и обсуждение

Рисунок 2 показывает динамику обучения агента. В начале обучения политика часто выбирает действия, которые либо не успевают убрать риск, либо создают лишнюю стоимость. По мере накопления Q-значений средняя награда растет, а доля нарушений снижается. Важный момент состоит не в самой форме кривой, а в том, что обучение происходит в ограниченном пространстве безопасных действий. Агент не получает возможности обучаться на заведомо недопустимых операциях Kubernetes.

Рисунок 2 — Динамика обучения агента Q-обучения

Итоговые значения по 24 тестовым трассам представлены в таблице 2. Реактивный HPA имел 3,85% интервалов с нарушением целевого уровня обслуживания. Прогнозный HPA снизил этот показатель до 2,82%, поскольку использовал верхний квантиль нагрузки при расчете числа подов. Предложенный контроллер дал 2,67% нарушений. Его преимущество над прогнозным HPA по доле нарушений невелико, но контроллер менял не только число подов, но и ресурсный профиль узкого микросервиса.

Таблица 2 - Сводные результаты на 24 тестовых трассах

Контроллер	**Нарушения, %**	**p95, мс**	**Средние ядра**	**Действия**
Реактивный HPA	3,85	366,7	17,61	633,6
Прогнозный HPA	2,82	348,9	18,87	279,5
Предложенный контроллер	2,67	354,4	19,92	473,5
Без прогнозного риска	7,53	411,2	19,29	423,9

Абляция без прогнозного риска важнее прямого сравнения с HPA. При той же структуре действия и той же обученной политике исключение прогнозного риска из состояния увеличило долю нарушений до 7,53% и p95 до 411,2 мс. Это означает, что прогнозный признак не является декоративным входом модели. Он меняет класс состояний, в которых агент выбирает защитное действие заранее.

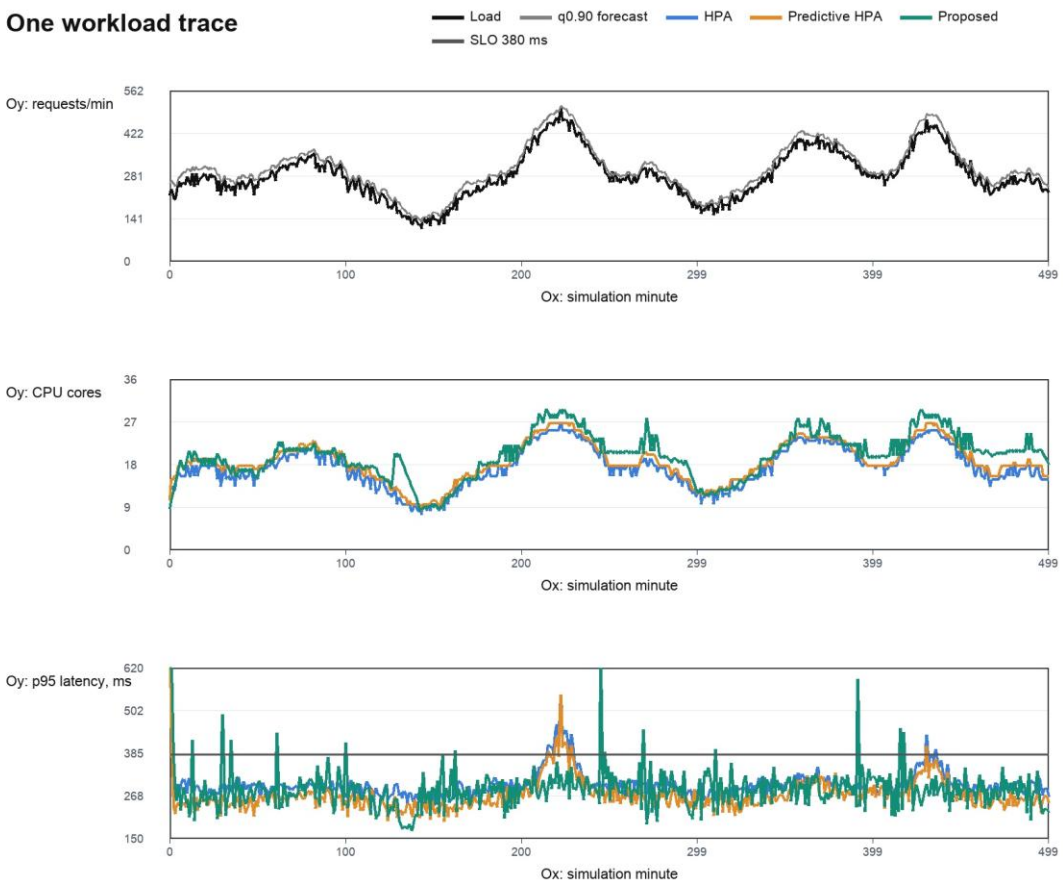

Рисунок 3 — Поведение контроллеров на одной тестовой трассе

На одиночной трассе видно различие контуров управления. Реактивный HPA часто совершает много малых действий, потому что реагирует на текущую утилизацию. Прогнозный HPA действует спокойнее, но меняет только число подов. Предложенный контроллер добавляет ресурсный профиль как вторую степень свободы: при риске он усиливает узкий сервис, а после ухода пика может снизить избыточную емкость.

Ресурсная цена предложенного контроллера выше, чем у реактивного HPA: 19,92 среднего ядра против 17,61. Это ожидаемая цена ориентации на p95. При этом число действий ниже, чем у HPA: 473,5 против 633,6. По сравнению с прогнозным HPA предложенный контроллер делает больше действий, потому что управляет двумя осями масштабирования. Практический выбор между этими вариантами зависит от того, насколько оператору важнее снизить долю нарушений или уменьшить число операций изменения состояния.

Полученные результаты согласуются с литературой. Vu, Tran и Kim показывают пользу прогнозного гибридного масштабирования для контейнерных приложений [6]. Gwydion демонстрирует, что агент обучения с подкреплением должен видеть состояние нескольких развертываний, а не отдельный сервис [13]. Предложенный метод объединяет эти идеи через прогнозный риск маршрута и маску безопасных действий. В отличие от подходов, где агент выбирает только метрику для HPA [9], здесь действие меняет емкость микросервиса напрямую.

Ограничения исследования

Ограничения существенны. Проверка выполнена в имитационной среде, а не в производственном Kubernetes-кластере. Модель задержки не учитывает сетевые повторные передачи, дисковую подсистему, базу данных и холодный запуск конкретных контейнерных образов. Прогноз Хольта выбран из-за низкой стоимости и воспроизводимости, но для нагрузок с сильными календарными режимами может потребоваться сезонная модель или модель с внешними признаками. Табличное Q-обучение удобно для анализа, но при десятках микросервисов пространство состояний станет слишком большим. В этом случае потребуется аппроксимация функции Q или разбиение политики по подсистемам приложения.

Также нельзя переносить численные проценты из этой работы на любой кластер. Они относятся к заданной имитационной среде и служат проверкой алгоритмической постановки. Для промышленного вывода нужны эксперименты в реальном кластере, несколько типов приложений, трассы разной природы и анализ чувствительности к порогу p95, ширине квантиля прогноза и стоимости изменения ресурсного профиля.

Выводы

1. Предложена архитектура прогнозно-рискового гибридного автомасштабировщика для Kubernetes, где прогноз нагрузки, оценка риска p95 и обучаемая политика выбора действия работают как единый контур.
2. В качестве прогнозной модели выбран метод Хольта с квантильной поправкой по остаткам. Он достаточно прост для встраивания в контроллер и дает верхнюю оценку нагрузки, пригодную для расчета риска.
3. В качестве алгоритма обучения с подкреплением использовано табличное Q-обучение с маской безопасных действий. Маска отделяет обучение от недопустимых операций и заставляет контроллер увеличивать емкость узкого сервиса при высоком прогнозном риске.
4. На 24 тестовых трассах предложенный контроллер снизил долю нарушений целевого уровня обслуживания с 3,85% у реактивного HPA до 2,67%. Абляция без прогнозного риска дала 7,53%, что подтверждает роль прогнозного состояния.
5. Улучшение качества обслуживания достигается ценой большего среднего выделения процессора. Поэтому метод целесообразен для сервисов, где p95-задержка важнее минимальной ресурсной стоимости.

Заключение

В ходе исследования была предложена архитектура автоматического гибридного масштабирования микросервисов в Kubernetes, ориентированная на соблюдение качества обслуживания при изменяющейся нагрузке. В отличие от реактивного масштабирования, предложенный подход использует прогноз нагрузки и оценку риска нарушения задержки p95, а затем передает это состояние агенту обучения с подкреплением. Агент выбирает не только изменение числа подов, но и изменение ресурсного профиля узкого микросервиса, что позволяет рассматривать горизонтальное и вертикальное масштабирование как единый контур управления.

Результаты имитационного эксперимента показали, что учет прогнозного риска снижает долю нарушений целевого уровня обслуживания по сравнению с реактивным и абляционным

вариантами. При этом улучшение качества обслуживания достигается ценой большего среднего выделения ресурсов, поэтому практическое применение метода требует настройки весов функции награды под требования конкретной системы. Дальнейшее развитие работы целесообразно связать с проверкой метода в реальном Kubernetes-кластере, расширением числа микросервисов и переходом от табличного Q-обучения к методам аппроксимации функции полезности.

Список литературы

1. Кубернетес. Горизонтальное автомасштабирование модулей. URL: https://kubernetes.io/docs/concepts/workloads/autoscaling/horizontal-pod-autoscale/ (дата обращения: 01.05.2026).
2. Кубернетес. Вертикальное автомасштабирование модулей. URL: https://kubernetes.io/docs/concepts/workloads/autoscaling/vertical-pod-autoscale/ (дата обращения: 01.05.2026).
3. Нгуен Т.-Т., Йем Й.-Дж., Ким Т., Парк Д.-Х., Ким С. Горизонтальное автомасштабирование подов в Kubernetes для эластичной оркестрации контейнеров // Датчики. 2020. Том. 20, № 16. Статья 4621. DOI: 10.3390/s20164621.
4. Rattihalli G., Govindaraju M., Lu H., Tiwari D. Изучение потенциала неразрушающего вертикального автоматического масштабирования и оценки ресурсов в Kubernetes // 2019 IEEE 12-я Международная конференция по облачным вычислениям. 2019. С. 33-40. DOI: 10.1109/CLOUD.2019.00018.
5. Baresi L., Hu D. Y. X., Quattrocchi G., Terracciano L. KOSMOS: Вертикальное и горизонтальное автоматическое масштабирование ресурсов для Kubernetes // Сервисно-ориентированные вычисления. Lecture Notes in Computer Science. 2021. С. 821-829. DOI: 10.1007/978-3-030-91431-8_59.
6. Vu D.-D., Tran M.-N., Kim Y. Predictive Hybrid Autoscaling for Containerized Applications // IEEE Access. 2022. Vol. 10. P. 109768-109778. DOI: 10.1109/ACCESS.2022.3214985.
7. Mondal S. K., Wu X., Kabir H. M. D., Dai H.-N., Ni K., Yuan H., Wang T. Toward Optimal Load Prediction and Customizable Autoscaling Scheme for Kubernetes // Mathematics. 2023. Vol. 11, № 12. Статья 2675. DOI: 10.3390/math11122675.
8. Rossi F., Nardelli M., Cardellini V. Горизонтальное и вертикальное масштабирование приложений на основе контейнеров с использованием обучения с подкреплением // 2019 IEEE 12-я Международная конференция по облачным вычислениям. 2019. С. 329-338. DOI: 10.1109/CLOUD.2019.00061.
9. Abdel Khaleq A., Ra I. Разработка агентов с учетом качества обслуживания с использованием обучения с подкреплением для автоматического масштабирования микросервисов в облаке // 2021 IEEE Международная конференция по автономным вычислениям и самоорганизующимся системам. 2021. DOI: 10.1109/ACSOS-C52956.2021.00025.
10. Xu M., Song C., Ilager S., Gill S. S., Ye K., Xu C.-Z. CoScal: Многогранное масштабирование микросервисов с помощью обучения с подкреплением // IEEE Transactions on Network and Service Management. 2022. DOI: 10.1109/TNSM.2022.3210211.

11. Hossen M. R., Islam M. A., Ahmed K. Практическое эффективное автомасштабирование микросервисов с обеспечением качества обслуживания // Труды 31-го Международного симпозиума по высокопроизводительным параллельным и распределенным вычислениям. 2022. С. 240-252. DOI: 10.1145/3502181.3531460.
12. Spatharakis D., Dimolitsas I., Vlahakis E. E., Dechouniotis D., Athanasopoulos N., Papavassiliou S. Распределенное автомасштабирование ресурсов в кластерах Kubernetes Edge // 18-я Международная конференция по управлению сетями и сервисами. 2022. С. 163-169. DOI: 10.23919/CNSM55787.2022.9965056.
13. Santos J., Reppas E., Wauters T., Volckaert B., De Turck F. Gwydion: Эффективное автомасштабирование сложных контейнеризированных приложений в Kubernetes с помощью обучения с подкреплением // Журнал сетевых и компьютерных приложений. 2025. Том. 234. Статья 104067. DOI: 10.1016/j.jnca.2024.104067.
14. Гайндман Р. Дж., Атанасопулос Г. Прогнозирование: принципы и практика. 3-е изд. Мельбурн: OTexts, 2021. URL: https://otexts.com/fpp3/ (дата обращения: 06.05.2026).
15. Саттон Р.С., Барто А.Г. Обучение с подкреплением: Введение. 2-е изд. Кембридж: MIT Press, 2018.
16. Уоткинс С.Дж.Ч., Даян П. Q-обучение // Машинное обучение. 1992. Том. 8. С. 279-292. DOI: 10.1007/BF00992698.

References

1. Kubernetes. Horizontal Pod Autoscaling. URL: https://kubernetes.io/docs/concepts/workloads/autoscaling/horizontal-pod-autoscale/ (дата обращения: 01.05.2026).
2. Kubernetes. Vertical Pod Autoscaling. URL: https://kubernetes.io/docs/concepts/workloads/autoscaling/vertical-pod-autoscale/ (дата обращения: 01.05.2026).
3. Nguyen T.-T., Yeom Y.-J., Kim T., Park D.-H., Kim S. Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration // Sensors. 2020. Vol. 20, No. 16. Article 4621. DOI: 10.3390/s20164621.
4. Rattihalli G., Govindaraju M., Lu H., Tiwari D. Exploring Potential for Non-Disruptive Vertical Auto Scaling and Resource Estimation in Kubernetes // 2019 IEEE 12th International Conference on Cloud Computing. 2019. pp. 33-40. DOI: 10.1109/CLOUD.2019.00018.
5. Baresi L., Hu D. Y. X., Quattrocchi G., Terracciano L. KOSMOS: Vertical and Horizontal Resource Autoscaling for Kubernetes // Service-Oriented Computing. Lecture Notes in Computer Science. 2021. pp. 821-829. DOI: 10.1007/978-3-030-91431-8_59.
6. Vu D.-D., Tran M.-N., Kim Y. Predictive Hybrid Autoscaling for Containerized Applications // IEEE Access. 2022. Vol. 10. P. 109768-109778. DOI: 10.1109/ACCESS.2022.3214985.
7. Mondal S. K., Wu X., Kabir H. M. D., Dai H.-N., Ni K., Yuan H., Wang T. Toward Optimal Load Prediction and Customizable Autoscaling Scheme for Kubernetes // Mathematics. 2023. Vol. 11, No. 12. Article 2675. DOI: 10.3390/math11122675.
8. Rossi F., Nardelli M., Cardellini V. Horizontal and Vertical Scaling of Container-Based Applications Using Reinforcement Learning // 2019 IEEE 12th International Conference on Cloud Computing. 2019. P. 329-338. DOI: 10.1109/CLOUD.2019.00061.

9. Abdel Khaleq A., Ra I. Development of QoS-aware agents with reinforcement learning for autoscaling of microservices on the cloud // 2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion. 2021. DOI: 10.1109/ACSOS-C52956.2021.00025.
10. Xu M., Song C., Ilager S., Gill S. S., Ye K., Xu C.-Z. CoScal: Multifaceted Scaling of Microservices With Reinforcement Learning // IEEE Transactions on Network and Service Management. 2022. DOI: 10.1109/TNSM.2022.3210211.
11. Hossen M. R., Islam M. A., Ahmed K. Practical Efficient Microservice Autoscaling with QoS Assurance // Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing. 2022. P. 240-252. DOI: 10.1145/3502181.3531460.
12. Spatharakis D., Dimolitsas I., Vlahakis E. E., Dechouniotis D., Athanasopoulos N., Papavassiliou S. Distributed Resource Autoscaling in Kubernetes Edge Clusters // 2022 18th International Conference on Network and Service Management. 2022. pp. 163-169. DOI: 10.23919/CNSM55787.2022.9965056.
13. Santos J., Reppas E., Wauters T., Volckaert B., De Turck F. Gwydion: Efficient auto-scaling for complex containerized applications in Kubernetes through Reinforcement Learning // Journal of Network and Computer Applications. 2025. Vol. 234. Article 104067. DOI: 10.1016/j.jnca.2024.104067.
14. Hyndman R. J., Athanasopoulos G. Forecasting: Principles and Practice. 3rd ed. Melbourne: OTexts, 2021. URL: https://otexts.com/fpp3/ (дата обращения: 06.05.2026).
15. Sutton R. S., Barto A. G. Reinforcement Learning: An Introduction. 2nd ed. Cambridge: MIT Press, 2018.
16. Watkins C. J. C. H., Dayan P. Q-learning // Machine Learning. 1992. Vol. 8. pp. 279-292. DOI: 10.1007/BF00992698.

Самоловов Д.А., Краснов А.Е. Формализация критерия отказоустойчивости критической информационной инфраструктуры в условиях трансформации киберугроз на основе интеграции показателей конфиденциальности, целостности и доступности // Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч.1 с. 30–39

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056:004.052

ФОРМАЛИЗАЦИЯ КРИТЕРИЯ ОТКАЗОУСТОЙЧИВОСТИ КРИТИЧЕСКОЙ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ В УСЛОВИЯХ ТРАНСФОРМАЦИИ КИБЕРУГРОЗ НА ОСНОВЕ ИНТЕГРАЦИИ ПОКАЗАТЕЛЕЙ КОНФИДЕНЦИАЛЬНОСТИ, ЦЕЛОСТНОСТИ И ДОСТУПНОСТИ

[1]Самоловов Д.А., [2]Краснов А.Е.
ФГБОУ ВО «МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ», Москва, Россия (119454, г. Москва, пр-т Вернадского, д. 78, стр. 4), e-mail: [1]samolovov37@gmail.com, [2]krasnovmgutu@yandex.ru

В статье представлен формализованный критериальный подход к оценке отказоустойчивости критической информационной инфраструктуры в условиях трансформации ландшафта киберугроз и роста значимости информационной безопасности. Актуальность исследования обусловлена необходимостью перехода от фрагментарных и реактивных моделей защиты к интегральным методам анализа устойчивости функционирования систем. Целью работы является разработка интегрального критерия отказоустойчивости на основе показателей конфиденциальности, целостности и доступности, учитывающего взаимосвязь рисков, динамику деградации устойчивости и вероятность отказа элементов. В ходе исследования выполнена формализация структуры критической информационной инфраструктуры как иерархической системы взаимосвязанных активов, предложена модель расчета частных рисков и их агрегирования с учетом взаимозависимости базовых критериев безопасности. Дополнительно разработана динамическая модель изменения устойчивости во времени и учтены механизмы распространения рисков между компонентами инфраструктуры. Практическая значимость работы заключается в возможности применения предложенного критерия для количественной оценки устойчивости, выявления критических состояний системы и обоснования управленческих решений, направленных на повышение надежности и эффективности механизмов информационной безопасности. Полученные результаты позволяют расширить существующие подходы к оценке устойчивости и могут быть использованы при построении систем мониторинга и управления рисками. Предложенный подход также создает основу для дальнейших исследований, направленных на адаптацию моделей к динамически изменяющимся условиям киберугроз и интеграцию интеллектуальных методов анализа.

Ключевые слова: Критическая информационная инфраструктура, отказоустойчивость, устойчивость, конфиденциальность, целостность, доступность.

FORMALIZATION OF THE CRITERION OF FAULT TOLERANCE OF CRITICAL INFORMATION INFRASTRUCTURE IN THE CONTEXT OF CYBER THREAT TRANSFORMATION BASED ON THE INTEGRATION OF CONFIDENTIALITY, INTEGRITY AND ACCESSIBILITY INDICATORS

[1]Samolovov D.A., [2]Krasnov A.E.
MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY, Moscow, Russia (119454, Moscow, avenue Vernadsky, 78, b. 4), e-mail: [1]samolovov37@gmail.com, [2]krasnovmgutu@yandex.ru

Самоловов Д.А., Краснов А.Е. Формализация критерия отказоустойчивости критической информационной инфраструктуры в условиях трансформации киберугроз на основе интеграции показателей конфиденциальности, целостности и доступности // Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч.1 с. 30–39

The article presents a formalized criteria-based approach to assessing the fault tolerance of critical information infrastructure in the context of the transformation of the cyber threat landscape and the growing importance of information security. The relevance of the study is due to the need to move from fragmented and reactive protection models to integrated methods for analyzing the stability of systems. The aim of the work is to develop an integrated fault tolerance criterion based on indicators of confidentiality, integrity and accessibility, taking into account the interrelation of risks, the dynamics of stability degradation and the probability of element failure. In the course of the study, the structure of the critical information infrastructure formalized as a hierarchical system of interrelated assets, and a model for calculating private risks and their aggregation proposed, taking into account the interdependence of basic security criteria. Additionally, a dynamic model of stability change over time been developed and the mechanisms of risk propagation between infrastructure components have been taken into account. The practical significance of the work lies in the possibility of applying the proposed criterion to quantify stability, identify critical states of the system and justify management decisions aimed at improving the reliability and effectiveness of information security mechanisms. The obtained results make it possible to expand existing approaches to assessing sustainability and can used in building monitoring and risk management systems. The proposed approach also creates the basis for further research aimed at adapting models to dynamically changing cyber threat conditions and integrating intelligent analysis methods.

Keywords: Critical information infrastructure, resilience, sustainability, confidentiality, integrity, accessibility.

Современный этап цифровизации характеризуется ростом зависимости социально-экономических процессов от функционирования критической информационной инфраструктуры (КИИ), что обуславливает повышение требований к ее устойчивости в условиях воздействия угроз информационной безопасности [1]. Развитие киберугроз, их усложнение и межсистемный характер приводят к тому, что традиционные подходы, ориентированные преимущественно на предотвращение инцидентов, не обеспечивают требуемого уровня надежности функционирования КИИ [2]. В этих условиях особую значимость приобретает обеспечение отказоустойчивости как способности системы сохранять работоспособность при реализации угроз и восстанавливаться после нарушений.

Существующие научные подходы к оценке устойчивости систем информационной безопасности базируются на анализе рисков и использовании критериев конфиденциальности, целостности и доступности, позволяющих количественно описать состояние защищенности информационных активов. Однако данные подходы, как правило, ограничиваются оценкой отдельных аспектов безопасности и не формируют интегрального показателя, отражающего способность системы к функционированию в условиях дестабилизирующих воздействий [Зайка 2024]. Это обусловливает необходимость разработки формализованного критерия отказоустойчивости, учитывающего совокупное влияние указанных критериев и их взаимосвязь. Целью настоящей работы является разработка критериального подхода к оценке отказоустойчивости КИИ на основе интегральной оценки показателей конфиденциальности, целостности и доступности. Для достижения поставленной цели решаются задачи формализации частных критериев, построения модели их агрегирования и обоснования применения полученного критерия для оценки устойчивости функционирования систем информационной безопасности.

1. Формализация структуры критической информационной инфраструктуры. Итак, в соответствии с подходом, представленным в работах А.Е. Краснова и соавторов, критическая информационная инфраструктура рассматривается как иерархическая система, состоящая из множества взаимосвязанных активов:

$A = \{A_1, A_2, \dots, A_N\}$,

с заданной структурой связей, представленной в виде ориентированного графа [7]. При этом устойчивость КИИ определяется не только характеристиками отдельных элементов, но и характером их взаимосвязей (независимость, слабая и сильная зависимость, обратная связь), что приводит к необходимости агрегирования показателей устойчивости на различных уровнях иерархии. Так, задача оценки устойчивости сводится к анализу:

- устойчивости отдельных активов;
- устойчивости их агрегатов;
- влияния межкомпонентных связей на распространение рисков.

2. Критериальная модель оценки рисков на основе CIA-подхода. В качестве базовых критериев оценки состояния активов используются показатели:

- конфиденциальности C;
- целостности I;
- доступности A.

Для каждого актива A_n вводятся коэффициенты значимости:

$K_c(A_n), K_i(A_n), K_a(A_n)$.

С целью нормирования используется [7]:

$$k_c(A_n) = \frac{K_c(A_n)}{K_c(A_n)+K_i(A_n)+K_a(A_n)},$$

$$k_i(A_n) = \frac{K_i(A_n)}{K_c(A_n)+K_i(A_n)+K_a(A_n)},$$

$$k_a(A_n) = \frac{K_a(A_n)}{K_c(A_n)+K_i(A_n)+K_a(A_n)}.$$

Риски по каждому критерию определяются как произведение:

$$\mu_c(R|A_n) = k_c(A_n) * \mu_c(T|A_n) * \mu_c(V|A_n),$$

$$\mu_i(R|A_n) = k_i(A_n) * \mu_i(T|A_n) * \mu_i(V|A_n),$$

$$\mu_a(R|A_n) = k_a(A_n) * \mu_a(T|A_n) * \mu_a(V|A_n),$$

где: $\mu(T)$ - степень реализации угрозы,

$\mu(V)$ - степень уязвимости,

а значения лежат в диапазоне $(0; 1)$.

Так, риск представляет собой логико-вероятностную оценку воздействия угроз на актив [Краснов 2021].

3. Агрегирование критериев и получение интегрального риска. Поскольку критерии C, I, A являются зависимыми, их объединение осуществляется с учетом корреляции [Краснов 2021]:

$$\mu_{ci}(R|A_n) = \mu_c + \mu_i - \mu_c\mu_i,$$

$$\mu_{cia}(R|A_n) = \mu_c + \mu_i + \mu_a - \mu_c\mu_i - \mu_c\mu_a + \mu_c\mu_i\mu_a.$$

Полученное выражение представляет собой:

- первые слагаемые - независимый вклад критериев;
- последующие - учет парных и тройных корреляций.

Итоговая величина:

$$\mu_{cia}(R|A_n)$$

характеризует интегральный риск актива.

4. Модель распространения рисков и уязвимостей. С учетом взаимосвязанности компонентов КИИ риск агрегированного узла определяется рекурсивно на основе рисков входящих элементов. Дополнительно в [Мосолов 2022] показано, что для анализа уязвимых комбинаций элементов используется комбинаторный подход:

$C_n^k = \frac{n!}{(n-k)!k!}$,

что позволяет определить множество возможных отказов системы и их комбинаций. Вероятность критического отказа фрагмента системы может быть представлена в виде:

$P = \prod p_{\text{отк}} * \prod p_{\text{безотк}}$,

что отражает совместное влияние отказов и безотказной работы элементов на устойчивость системы. Данный подход позволяет выявлять наиболее критические сегменты КИИ, обладающие максимальной уязвимостью.

5. Динамическая модель устойчивости системы. В работах [Кузнецов 2024, Краснов 2024] устойчивость рассматривается как динамический показатель, изменяющийся во времени. Базовая модель изменения уровня устойчивости имеет экспоненциальный характер:

$X_n = X_{n-1} e^{-\Delta t / T}$,

где T - период потери устойчивости,

Δt - временной шаг.

Такая модель отражает естественное снижение устойчивости системы под воздействием угроз. Дополнительно динамика риска описывается дифференциальным уравнением:

$\frac{d\mu}{dt} = -\frac{\mu}{\tau} + \Delta ACT(t)$,

где τ - время релаксации,

ΔACT - результирующая активность угроз и мер противодействия.

Так, устойчивость системы определяется как результат баланса активности угроз и эффективности защитных механизмов.

6. Формирование критерия отказоустойчивости (авторское предложение). На основе рассмотренных моделей предлагается ввести интегральный критерий отказоустойчивости КИИ в виде функции от:

- интегрального риска μ_{cia};
- динамики изменения устойчивости;
- вероятности отказа элементов.

Предлагаемый критерий может быть записан в общем виде:

$K_{fail} = (1 - \mu_{cia}) * e^{-t/T} * (1 - P_{\text{отк}})$,

где μ_{cia} - интегральный риск (CIA),

$e^{-t/T}$ - динамика деградации устойчивости,

$P_{\text{отк}}$ - вероятность отказа критических элементов.

Интерпретация:

- при росте рисков → критерий уменьшается;
- при росте вероятности отказа → система теряет устойчивость;
- при увеличении времени без управления → устойчивость падает.

7. Пример расчета критерия отказоустойчивости КИИ. Для демонстрации работоспособности предложенного критерия рассмотрим условный актив КИИ, для которого заданы значения частных рисков по критериям конфиденциальности, целостности и доступности:

$\mu_c = 0.3, \mu_i = 0.4, \mu_a = 0.5.$

Расчет интегрального риска. С учетом зависимости критериев используем выражение [Краснов 2024]:

$\mu_{cia} = \mu_c + \mu_i + \mu_a - \mu_c\mu_i - \mu_c\mu_a - \mu_i\mu_a + \mu_c\mu_i\mu_a,$

Подставляя значения:

$\mu_{cia} = 0.3 + 0.4 + 0.5 - 0.12 - 0.15 - 0.2 + 0.06 = 0.79.$

Задание параметров модели. Примем:

- период потери устойчивости: $T = 10$;
- вероятность отказа: $P_{\text{отк}} = 0.2$.

Критерий отказоустойчивости:

$K_{fail} = (1 - \mu_{cia}) * e^{-t/T} * (1 - P_{\text{отк}}),$

$K_{fail} = 0.21 - e^{-t/10} * 0.8 = 0.168 * e^{-t/10}.$

Результаты расчета (для разных моментов времени) представлены в Таблице 1.

Таблица 1 – Расчет критерия отказоустойчивости во времени

t	$(e^{-t/10})$	(K_{fail})
0	1,000	0,168
2	0,819	0,138
4	0,670	0,113
6	0,549	0,092
8	0,449	0,075
10	0,368	0,062
12	0,301	0,051
14	0,247	0,042
16	0,202	0,034

Для анализа изменения отказоустойчивости критической информационной инфраструктуры во времени на основе данных Таблицы 1 построена зависимость критерия K_{fail} от времени функционирования системы при фиксированных значениях интегрального риска и вероятности отказа (Рисунок 1).

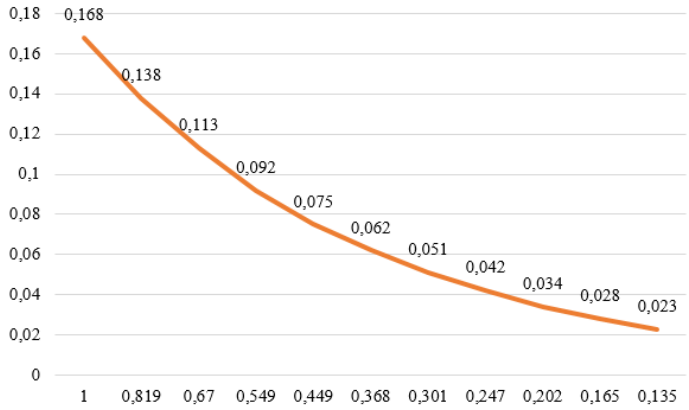

Рисунок 1 – Зависимость критерия отказоустойчивости от времени функционирования системы

Полученная зависимость носит экспоненциальный характер, что отражает естественное снижение устойчивости системы при отсутствии корректирующих воздействий. Установлено, что на начальном этапе снижение критерия происходит наиболее интенсивно, после чего темп деградации уменьшается. Это свидетельствует о необходимости реализации механизмов управления устойчивостью на ранних стадиях развития инцидента для предотвращения значительного снижения отказоустойчивости.

Анализ влияния вероятности отказа. Теперь зафиксируем:

$$t = 5, e^{-0.5} = 0.607$$

$K_{fail} = (1 - 0.79) * 0.607 * (1 - P_{\text{отк}}) = 0.127 * (1 - P_{\text{отк}})$.

В Таблице 2 представлены результаты расчета влияния вероятности отказа.

Таблица 2 - Влияние вероятности отказа

$(P_{\text{отк}})$	(K_{fail})
0,0	0,127
0,1	0,114
0,2	0,102
0,3	0,089
0,4	0,076
0,5	0,064
0,6	0,051
0,7	0,038
0,8	0,025

На основе данных Таблицы 2 построена зависимость критерия отказоустойчивости K_{fail} от вероятности отказа элементов КИИ при фиксированных значениях времени и интегрального риска (Рисунок 2).

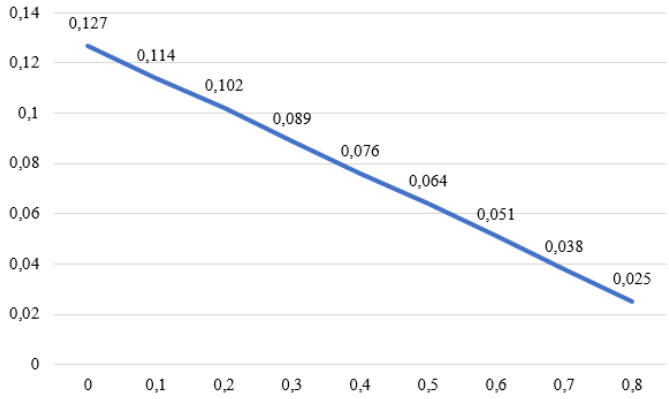

Рисунок 2 – Зависимость критерия отказоустойчивости от вероятности отказа элементов системы

Анализ графика показывает практически линейное снижение критерия отказоустойчивости с увеличением вероятности отказа. Данный результат подтверждает, что надежность отдельных компонентов системы оказывает прямое влияние на устойчивость КИИ в целом. При достижении высоких значений вероятности отказа наблюдается критическое снижение K_{fail}, что указывает на необходимость приоритизации мероприятий по повышению надежности ключевых элементов инфраструктуры.

Анализ влияния уровня риска (CIA). Зафиксируем:

$t = 5 => e^{-0.5} = 0.607,$

$P_{\text{отк}} = 0.2,$

Тогда:

$K_{fail} = (1 - \mu_{cia}) * 0.607 * 0.8 = 0.486 * (1 - \mu_{cia}).$

В Таблице 3 отражено влияние интегрального риска.

Таблица 3 - Влияние интегрального риска

(μ_{cia})	(K_{fail})
0,1	0,437
0,2	0,389
0,3	0,340
0,4	0,292
0,5	0,243
0,6	0,194
0,7	0,146
0,8	0,097
0,9	0,049

Для оценки влияния уровня информационных рисков на отказоустойчивость КИИ на основе данных Таблицы 3 построена зависимость критерия K_{fail} от интегрального показателя риска μ_{cia}, агрегирующего критерии конфиденциальности, целостности и доступности.

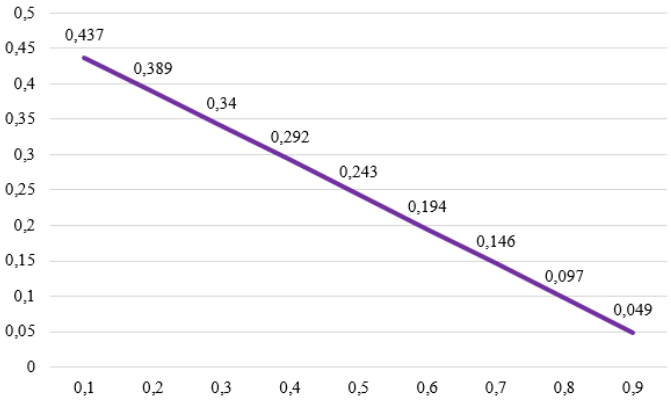

Рисунок 3 – Зависимость критерия отказоустойчивости от интегрального риска (CIA)

Полученные результаты демонстрируют снижение критерия отказоустойчивости при увеличении интегрального риска. Наиболее существенное падение наблюдается в диапазоне высоких значений μ_{cia}, что свидетельствует о высокой чувствительности системы к ухудшению параметров безопасности. Данный факт подтверждает ключевую роль критериев конфиденциальности, целостности и доступности в формировании устойчивости КИИ и обосновывает необходимость их комплексного контроля.

Полученные результаты демонстрируют, что критерий отказоустойчивости является чувствительным к трем ключевым факторам: уровню интегрального риска, вероятности отказа и времени функционирования системы без корректирующих воздействий. При этом наибольшее влияние оказывает интегральный риск, формируемый на основе критериев конфиденциальности, целостности и доступности, что подтверждает корректность выбранного критериального подхода.

В результате проведенного исследования разработан критериальный подход к оценке отказоустойчивости критической информационной инфраструктуры на основе интеграции показателей конфиденциальности, целостности и доступности. На основе анализа работ А.Е. Краснова и соавторов выполнена формализация взаимосвязи между рисками информационной безопасности, устойчивостью системы и вероятностью отказа ее элементов. Сформирован математический аппарат, позволяющий перейти от оценки частных критериев к интегральному показателю отказоустойчивости, учитывающему как структурные особенности КИИ, так и динамику воздействия угроз. Проведенные расчетные эксперименты подтвердили адекватность предложенной модели и ее применимость для анализа устойчивости функционирования систем в условиях дестабилизирующих воздействий. В качестве основных выводов по работе следует отметить:

1. Разработана математическая модель оценки отказоустойчивости КИИ, основанная на агрегировании критериев конфиденциальности, целостности и доступности с учетом их взаимозависимости и влияния на интегральный риск системы.

2. Установлено, что отказоустойчивость КИИ определяется совокупным влиянием интегрального риска, вероятности отказа элементов и времени функционирования системы, при этом наибольшее влияние оказывает уровень информационного риска.

3. Подтверждено, что предложенный критерий отказоустойчивости позволяет количественно оценивать устойчивость КИИ и выявлять критические условия ее деградации, что обеспечивает возможность обоснования управленческих решений по повышению надежности и устойчивости функционирования системы.

Список литературы

1. Родионов М. А., Егоров Е. Ю. защита критической информационной инфраструктуры как необходимое условие обеспечения экономической безопасности России // Экономика и бизнес: теория и практика. 2024. №3-2 (109). С. 77-81.
2. Мальцев М.С., Кулакова Н.С. Защита сети значимых объектов критической информационной инфраструктуры (КИИ) АСУТП // Вестник магистратуры. 2023. №10-2 (145). С. 40-42.
3. Зайка В. М. Обеспечение безопасности объекта критической информационной инфраструктуры // Вестник науки. 2024. №10 (79). С. 750-758.
4. Мосолов А.С., Краснов А.Е., Урбан Н.А. О применении метода анализа уязвимостей технологического процесса производственного объекта для обеспечения информационной безопасности асу тп с учётом взаимосвязи компонентов // Безопасность информационных технологий = IT Security. Том 29. № 3. 2022. С. 38-52.
5. Кузнецов А. С., Краснов А. Е. Информационное обеспечение импульсного управления устойчивостью систем информационной безопасности // Information Science. Information Security. Mathematics. 2024. № 2. С. 99–108.
6. Краснов А. Е., Кузнецов А. С., Смирнов В. М. Модель импульсного управления устойчивостью системы информационной безопасности // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2024. № 1. С. 80–90.
7. Краснов А. Е., Мосолов А. С., Феоктистова Н. А. Оценивание устойчивости критических информационных инфраструктур к угрозам информационной безопасности // Безопасность информационных технологий = IT Security. 2021. Т. 28. № 1. С. 106–120.
8. Бакшеев А.С., Лившиц И.И. Разработка методики контроля уровня защищенности информации объектов критической информационной инфраструктуры // Вопросы кибербезопасности. 2023. №2 (54). С. 85-98.

References

1. Rodionov M. A., Egorov E. Yu. Protection of critical information infrastructure as a necessary condition for ensuring Russia's economic security // Economy and business: theory and practice. 2024. No. 3-2 (109). pp. 77-81.
2. Mal'tsev M. S., Kulakova N. S. Protection of the network of significant objects of critical information infrastructure (CII) of automated process control systems // Bulletin of the Magistracy. 2023. No. 10-2 (145). pp. 40-42.
3. Zayka V. M. Ensuring the security of a critical information infrastructure facility // Bulletin of science. 2024. No. 10 (79). pp. 750-758.
4. Mosolov A. S., Krasnov A. E., Urban N. A. On the Application of the Vulnerability Analysis Method for the Technological Process of an Industrial Facility to Ensure Information Security

of an APCS Taking into Account the Interrelationships of Components // IT Security. Vol. 29. No. 3. 2022. pp. 38-52.

5. Kuznetsov A. S., Krasnov A. E. Information Support for Impulse Control of the Stability of Information Security Systems // Information Science. Information Security. Mathematics. 2024. No. 2. pp. 99-108.
6. Krasnov A. E., Kuznetsov A. S., Smirnov V. M. A Model of Impulse Control of the Stability of an Information Security System // Bulletin of the Russian State University for the Humanities. Series "Computer Science. Information Security. Mathematics". 2024. No. 1. pp. 80-90.
7. Krasnov A. E., Mosolov A. S., Feoktistova N. A. Assessing the Resilience of Critical Information Infrastructures to Information Security Threats // Information Technology Security = IT Security. 2021. Vol. 28. No. 1. pp. 106–120.
8. Baksheev A. S., Livshits I. I. Development of a Methodology for Monitoring the Level of Information Security of Critical Information Infrastructure Facilities // Cybersecurity Issues. 2023. No. 2 (54). pp. 85-98.

Рогачев А.Е. Искусственный интеллект в оптимизации рецептур и производственных процессов пищевой промышленности // Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч.1 с. 40–44

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.8

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ОПТИМИЗАЦИИ РЕЦЕПТУР И ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ

Рогачев А.Е.
ОЧУ ВО "МОСКОВСКИЙ УНИВЕРСИТЕТ ИМЕНИ А.С.ГРИБОЕДОВА", Москва, Россия (105066, город Москва, Новая Басманная ул., д. 35 стр. 1), e-mail: mr.spartak747@mail.ru

В работе рассмотрены основные аспекты применения искусственного интеллекта в пищевой промышленности. Цель исследования заключалась в анализе научных работ по заявленной теме для определения возможностей искусственного интеллекта в оптимизации производственных процессов. Научная и практическая значимость работы заключается в попытке обозначить юридические уязвимости для совершенствования правовой базы применения искусственного интеллекта в пищевой отрасли. Методологию исследования составили общенаучные методы. Результатом исследования стал обобщенный научный опыт и рассуждения юридического толка с целью обозначить ограничения и риски технологии искусственного интеллекта в пищевой отрасли. Работа вносит соответствующий вклад в сегмент научных исследований по теме информационных технологий в пищевой отрасли.

Ключевые слова: Искусственный интеллект, оптимизация рецептур, ИИ, производственные процессы, пищевая промышленность, автоматизация.

ARTIFICIAL INTELLIGENCE IN THE OPTIMIZATION OF RECIPES AND PRODUCTION PROCESSES IN THE FOOD INDUSTRY

Rogachev A.E.
A.S. GRIBOYEDOV MOSCOW UNIVERSITY, Moscow, Russia (105066, Moscow, Novaya Basmannaya str., 35 bld. 1), e-mail: mr.spartak747@mail.ru

This paper examines the key aspects of artificial intelligence application in the food industry. The objective of the study was to analyze scientific papers on the topic to determine the potential of artificial intelligence for optimizing production processes. The scientific and practical significance of the paper lies in its attempt to identify legal vulnerabilities for improving the legal framework for the application of artificial intelligence in the food industry. The research methodology utilized general scientific methods. The result of the study was a summary of scientific experience and legal considerations aimed at identifying the limitations and risks of artificial intelligence technology in the food industry. This paper makes a relevant contribution to scientific research on information technology in the food industry.

Keywords: Artificial intelligence, recipe optimization, AI, manufacturing processes, food industry, automation.

Введение

Изучение возможностей искусственного интеллекта (ИИ) в разных сферах жизнедеятельности является приоритетной областью исследований последних десяти лет. Проблема исследований в области внедрения инновационных технологий в пищевую отрасль, в частности, искусственного интеллекта, связана с устареванием традиционных форматов производства. Требования к производству растут день ото дня, особенно в условиях прогрессирующих технологий и автоматизированных систем. В рамках производственного процесса внутри пищевой промышленности автоматизация уже давно стала необходимостью,

которая определяет не только скорость выполнения рабочих задач, но и эффективность, ведь умные технологии работают без рисков, сопровождаемых человеческим фактором.

С технической точки зрения следует отметить, что определение характеристик пищевых ингредиентов требует дорогостоящих приборов и методов для четкой идентификации, количественного определения и биохимического анализа, и комбинация данных процессов представляется затруднительной на данный момент вследствие недостаточной степени научной проработанности сферы модернизации применяемых технологий.

Таким образом, научная проблематика в области применения ИИ для оптимизации рецептур и производственных процессов пищевой промышленности заключается в отсутствии адекватных данных и представлений о функциональных свойствах ингредиентов, сложности моделирования многофакторных связей «ингредиенты—процесс—качество», ограничениях интерпретируемости моделей и интеграции ИИ в существующие технологические цепочки и регуляторные требования.

Целью исследования является анализ научных работ по заявленной теме для определения возможностей ИИ в оптимизации производства пищевой промышленности.

Задачи исследования включают:

1) изучить существующие научные исследования по теме;

2) проанализировать риски и возможности искусственного интеллекта в пищевой отрасли;

3) обобщить полученные результаты.

Методы исследования.

Проблема искусственного интеллекта в оптимизации рецептур и производственных процессов пищевой промышленности изучалась при помощи комбинации теоретических методов: анализ, сравнение, дедукция, индукция, моделирование, абстракция.

Результаты исследования.

Искусственные интеллект – это самообучающаяся система, обеспечивающая автоматизацию и оптимальность производственных процессов [2]. Принцип работы ИИ основан на автоматизированном повторении зацикленных действий, которые можно встроить в определенный поведенческий паттерн. В пищевой промышленности ИИ работает как инструмент для автоматизации наблюдения и управления производственными процессами, а также как аналитик данных для прогнозов и оптимизации. Ко всему прочему ИИ реализует функцию помощника при разработке новых продуктов и контроле качества.

Возможности ИИ в пищевой отрасли уже давно изучены в научной литературе. Так, Гамарник И.А. в своей статье объясняет возможности ИИ анализировать пищевые характеристики, чтобы создавать добавки и продукты с учетом требуемых параметров. Автор сообщает о том, что внедрение искусственного интеллекта в сферу пищевых систем позволит обеспечить процесс высокопроизводительного фракционирования, идентификации и количественного определения (как ожидается, новых) природных биоактивных веществ [1]. Информационные механизмы позволяют создавать рецептуру продукции по заданным параметрам, к примеру, чат ChatGPT может предложить новый состав витаминно-минерального комплекса с учетом конкретного запроса и автоматически создать цифровой двойник товара или определить степень халяльности конкретной пищевой добавки. Более

того, он способен подключить к процессу разработки потребителя: алгоритм анализирует, какие вкусы или ингредиенты люди чаще всего выбирают и, отталкиваясь от этого, предлагает новацию. Анализируя необработанные данные в рамках генетического тестирования, ИИ дает информацию о планах питания на основе того, как организм перерабатывает определенные питательные вещества (или нет), и оценить предрасположенность к пищевой непереносимости. Кроме того, ML-алгоритмы могут прогнозировать изменения некоторых параметров: микробиома кишечника, реакции глюкозы в крови на потребление блюда или ингредиента [1].

Оборотов И.В. в своей статье раскрывает возможности RFID-технологии, которая позволяет оптимизировать процесс замеса теста в хлебопекарной отрасли путём автоматического отслеживания ингредиентов, параметров процесса (температура, время), состояния оборудования и сбора данных для аналитики, что повышает точность, стабильность производства, снижает затраты, брак и влияние человеческого фактора, а также улучшает общую производительность и качество продукции [5].

ИИ предсказывает сроки годности и риск микробиологического роста, выявляет потенциальные контаминации и помогает автоматизировать аудит соответствия санитарным и регуляторным требованиям. Так, Рожнов Е.Д. и Школьникова М.Н. отмечают роль интеллектуальных сенсоров, таких как «электронный язык», «электронный нос» и машинное зрение, задействованных на всем жизненном цикле пищевого продукта «от фермы до вилки». Например, применение «электронного носа» в совокупности с ИИ позволяет определить, подвергались ли говядина, свинина и мясо курицы замораживанию/размораживанию, прогнозировать уровень кислотности в зернах кофе при обжарке, выявлять порчу рыбы, фальсификацию коровьего топленого масла и т.д. [6]

Отдельно стоит отметить роль ИИ и нейросетей в создании и оптимизации рецептур. Нейросетевые алгоритмы способны анализировать тысячи существующих рецептов, выявляя лучшие сочетания ингредиентов и создавая уникальные кулинарные композиции. Функционирование таких систем основано на комплексном анализе кулинарных баз данных. Они выявляют скрытые закономерности, определяя наиболее гармоничные комбинации продуктов и способов их приготовления, что позволяет генерировать новые рецепты и адаптировать существующие под конкретные запросы пользователей [4]. Отдельным преимуществом нейросетей является возможность анализа огромных баз данных, например, изображений ингредиентов при помощи декодера [8].

Обсуждение.

Приведенные результаты исследований демонстрируют прогресс умных технологий в сфере пищевой промышленности. Немаловажный аспект, о котором не было упомянуто, связан с тем, что ИИ может быть использован как инструмент экологизации пищевой отрасли. Если ИИ способен оптимизировать приготовление блюд таким образом, чтобы все было выверено с точностью до малейшего процента, то очевидным видится способность ИИ решать проблему с излишками на складах и остаточного сырья. Помимо сказанного ИИ играет роль связующего звена между логистикой и сбытом: при помощи умных технологий можно точно рассчитывать сроки хранения, логистические ограничения и прогнозируемый спрос. Это особенно важно для скоропортящихся продуктов, где ошибки в управлении запасами могут привести к значительным потерям [7].

Однако, нельзя обойти стороной вопрос рисков, связанных с применением ИИ. На данный момент технологии на базе ИИ имеют очевидные уязвимости, например, нерешенный вопрос с правосубъектностью, что в перспективе может иметь юридические проблемы. Если, к примеру, технология создаст ошибочный рецепт или не сможет грамотно спрогнозировать аллергическую реакцию потребителя, то в таком случае ИИ будет виновником понесенного ущерба, в том числе репутационного для производителя.

Иная сторона проблемы связана с контролирующей функцией ИИ в системе контроля пищевого качества. В ряде исследований отмечается, что ИИ способен контролировать процесс управления качеством мясопродуктов, но технологии не способны заменить человека из-за юридической неизбежности. Так, отмечается «невозможность адаптивного управления качеством фаршевой смеси вследствие отсутствия конкретного знания о формуле КПД, представляющей коммерческую тайну» [3].

Выводы.

Таким образом, ИИ представляет собой перспективный инструмент управления качеством, автоматизации производственных процессов и оптимизации узконаправленных сегментов, например, таких как создание рецептур. Можно утверждать, что ИИ уже доказал свою эффективность, однако технология не лишена недостатков, и в первую очередь связаны они с юридической неопределенностью. Практическая значимость полученных выводов заключается в формулировке правовой и организационной базы внедрения ИИ в пищевую отрасль. Основные направления для дальнейшего исследования в этой области могут затрагивать вопросы методологий валидации и верификации ИИ-систем в производственной среде (как проверять корректность рецептур, как учитывать шумовые факторы и сменность технологических условий).

Список литературы

1. Гамарник И.А., Седугина А.С., Карагодин В.П. Использование механизмов искусственного интеллекта в пищевой промышленности // Промышленность: экономика, управление, технологии. Т.2. №3(6). 2023. С. 30-36.
2. Гербер Ю.Б., Балко С.В., Якушев А.А. Цифровой формат развития пищевой промышленности в современных экономических условиях // Экономика, предпринимательство и право. - 2022. - Том 12. - № 5. -С. 1613-1624.
3. Карпов В.И., Красуля О.Н., Токарев А.В. Искусственный интеллект в технологической системе производства колбас заданного качества // Вестник ВГУИТ.2017. Т. 79. № 1. С. 106-113.
4. Коршиков С.В. Искусственный интеллект в пищевой отрасли // Известия Кабардино-Балкарского научного центра РАН. Том 27. №3. 2025. С. 99-105.
5. Оборотов И.В., Локтев Д.А. Оптимизация процесса замеса теста с помощью RFID технологий // Вестник науки. №5(86). Том 4. С.1658-1662.
6. Рожнов Е.Д., Школьникова М.Н. Приоритетные тренды пищевых технологий в свете цифровизации производства и концепции устойчивого развития // Индустрия питания|Food Industry. 2025. Т. 10, № 1. С. 87-98.
7. Савенкова Т.В., Замула В.С. Актуальные вопросы классификации и контроля качества при применении систем и технологий искусственного интеллекта в производстве

пищевой продукции // Информационно-экономические аспекты стандартизации и технического регулирования. 2026. № 2(89). С. 44-49.

8. Сорокин А.В. Управление качеством: Учебное пособие для студентов всех форм обучения направления подготовки «Менеджмент». Издание 2-е дополненное и исправленное. Рубцовск, 2021. - 106 с.

References

1. Gamarnik I.A., Sedugina A.S., Karagodin V.P. Use of Artificial Intelligence Mechanisms in the Food Industry // Industry: Economics, Management, Technology. Vol. 2. No. 3(6). 2023. pp. 30-36.
2. Gerber Yu.B., Balko S.V., Yakushev A.A. Digital Format for the Development of the Food Industry in Modern Economic Conditions // Economics, Entrepreneurship and Law. - 2022. - Vol. 12. - No. 5. - pp. 1613-1624.
3. Karpov V.I., Krasulya O.N., Tokarev A.V. Artificial Intelligence in the Technological System for the Production of Sausages of a Given Quality // VSUET Bulletin. 2017. Vol. 79. No. 1. pp. 106-113.
4. Korshikov S.V. Artificial Intelligence in the Food Industry // Bulletin of the Kabardino-Balkarian Scientific Center of the Russian Academy of Sciences. Vol. 27. No. 3. 2025. pp. 99-105.
5. Oborotov I.V., Loktev D.A. Optimization of the Dough Mixing Process Using RFID Technologies // Science Herald. No. 5(86). Vol. 4. pp. 1658-1662.
6. Rozhnov E.D., Shkolnikova M.N. Priority Trends in Food Technologies in Light of Digitalization of Production and the Concept of Sustainable Development // Food Industry. 2025. Vol. 10, No. 1. pp. 87-98.
7. Savenkova T.V., Zamula V.S. Current Issues of Classification and Quality Control in the Application of Artificial Intelligence Systems and Technologies in Food Production // Information and Economic Aspects of Standardization and Technical Regulation. 2026. No. 2 (89). pp. 44-49.
8. Sorokin A.V. Quality Management: A Textbook for Students of All Forms of Study Majoring in Management. 2nd Edition, Supplemented and Revised. Rubtsovsk, 2021. - p.106

Лежанков М.Е. Исследование эффективности HONEYPOT-решений в ВЕБ-приложениях // Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6(68) Ч.1 с. 45–56

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056.57:004.77

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ HONEYPOT-РЕШЕНИЙ В ВЕБ-ПРИЛОЖЕНИЯХ

Лежанков М.Е.
ФГАОУ ВО "НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО" , Санкт-Петербург, Россия (197101, город Санкт-Петербург, Кронверкский пр-кт, д. 49 литер а), e-mail: lemmaksim44@gmail.com

В данной статье представлено экспериментальное исследование эффективности honeypot-решений для защиты веб-приложений от автоматизированных угроз. Проведен анализ отечественных и зарубежных научных работ, посвященных применению honeypot-систем, CAPTCHA и методов обнаружения веб-ботов. В рамках исследования разработан тестовый стенд с различными типами веб-форм, включающими классические honeypot-ловушки, Google reCAPTCHA v2 и Cloudflare Turnstile. Для проверки эффективности защитных механизмов использовались простые HTTP-боты и более продвинутые headless-боты, имитирующие поведение реального пользователя. На основе серии экспериментов выполнен сравнительный анализ решений по критериям скорости обнаружения, устойчивости, уровня ложных срабатываний, производительности и качества собираемых данных. Результаты показали, что классические honeypot-механизмы эффективны преимущественно против простых ботов, однако теряют эффективность при использовании современных средств автоматизации. Наиболее высокие показатели защиты продемонстрировали CAPTCHA-системы, особенно Cloudflare Turnstile, обеспечивающая высокий уровень обнаружения автоматизированных запросов при минимальном влиянии на пользовательский опыт.

Ключевые слова: Honeypot, веб-приложение, информационная безопасность, CAPTCHA, веб-боты, headless-боты, Cloudflare Turnstile, Google reCAPTCHA, защита веб-форм, автоматизированные атаки.

INVESTIGATION OF THE EFFECTIVENESS OF HONEYPOT SOLUTIONS IN WEB APPLICATIONS

Lezhankov M.E.
"ITMO NATIONAL RESEARCH UNIVERSITY", St. Petersburg, Russia (197101, St. Petersburg, Kronverksky prospekt, 49 litera a), e-mail: lemmaksim44@gmail.com

This article presents an experimental study of the effectiveness of honeypot solutions for protecting web applications from automated threats. The analysis of domestic and foreign scientific papers devoted to the use of honeypot systems, CAPTCHA and web bot detection methods is carried out. As part of the research, a testbed has been developed with various types of web forms, including classic honeypot traps, Google reCAPTCHA v2 and Cloudflare Turnstile. To test the effectiveness of the protection mechanisms, simple HTTP bots and more advanced headless bots that mimic the behavior of a real user were used. Based on a series of experiments, a comparative analysis of solutions was performed according to the criteria of detection speed, stability, false alarm rate, performance and quality of the data collected. The results showed that classic honeypot mechanisms are effective mainly against simple bots, but they lose effectiveness when using modern automation tools. The highest protection rates were demonstrated by CAPTCHA systems, especially Cloudflare Turnstile, which provides a high level of detection of automated queries with minimal impact on the user experience.

Keywords: Honeypot, web application, information security, CAPTCHA, web bots, headless bots, Cloudflare Turnstile, Google reCAPTCHA, web form protection, automated attacks.

В условиях цифровизации и стремительного развития информационных технологий наличие корпоративного веб-сайта стало обязательным элементом деятельности большинства современных компаний. Такие сайты могут выполнять как информационные функции в виде лендинг-страниц, так и представлять собой сложные веб-приложения, включая интернет-магазины, сервисные платформы и т.п. Компании активно стремятся перенести процессы и данные в онлайн-среду, обеспечивая удобный доступ к информации для клиентов и партнеров.

Однако открытость веб-ресурсов делает их уязвимыми. Для противодействия угрозам разработчики внедряют различные средства защиты. Одним из эффективных решений являются honeypot-системы – элементы пассивной защиты, встроенные в архитектуру веб-приложения. Они могут использоваться как для выявления несанкционированных попыток доступа к административным панелям и API, так и для защиты от атак, связанных с загрузкой вредоносных файлов, а также для предотвращения эксплуатации известных уязвимостей.

Для проведения исследования необходимо провести анализ отечественных научных источников, который охватывает большое количество тем связанных с системами Honeypot в веб-приложениях. Одним из важных пунктов является оценка актуальности и перспективности технологии. Работа «Анализ ханипотов с открытым исходным кодом» [1] за авторством А.А. Сергановского и В.Н. Романовича содержит анализ репозиториев платформы GitHub по запросу «honeypot». В работе представлены графики, отражающие результаты проведенного анализа, рассмотрим один из них. На рисунке 1 представлена гистограмма, демонстрирующая количество всех проанализированных, уникальных honeypot. Как видно из графика, веб-ловушки, направленные на имитацию веб-приложений и сайтов, занимают одну из лидирующих позиций. В исследовании авторы упоминают, что за последнее время увеличилась активность обновлений репозиториев. Таким образом еще раз доказывается актуальность исследований honeypot-систем в области информационной безопасности веб-приложений.

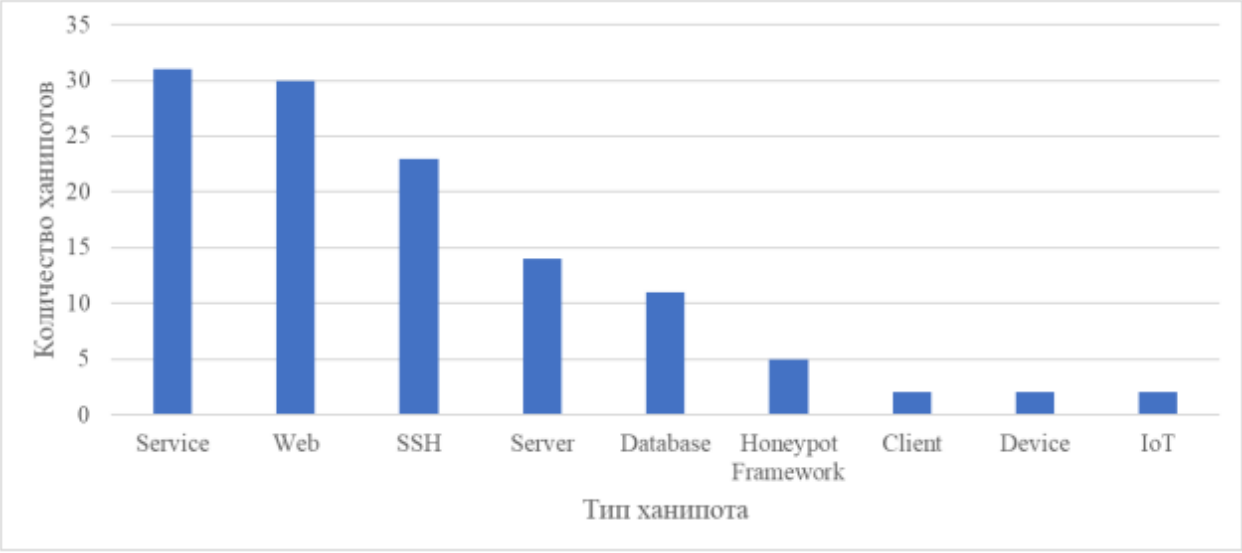

Рисунок 1 - Распределение honeypot по типам

При анализе статей был также рассмотрен потенциал использования honeypot-систем для обнаружения и изучения поведения веб-ботов. Работа «Защита веб-контента от несанкционированного автоматизированного сбора информации» [2] содержит исследование,

направленное на обзор методов защиты информации на веб-сайтах. Авторы Р.Ю. Демина, Д.Э. Шукралиева и Р.Р. Ференс объясняют разницу между легитимными и нелегитимными ботами и приводят статистику количества нежелательных ботов в сети.

Легитимные боты используются для сбора легальной информации согласно всем правилам сайтов и законов Российской Федерации, в то время как нелегитимные боты собирают информацию, которая содержит персональные данные пользователей, интеллектуальную собственность, финансово-значимый контент или информацию, предназначенную для нанесения ущерба веб-ресурсам. По данным этого исследования доля нежелательного роботизированного трафика в 2022 году составила 24% всего веб-трафика, что показывает значительный рост таких программ по сравнению с 2017 годом.

Авторы выделяют следующие методы защиты:

- CAPTCHA;
- динамическая смена идентификаторов значимых элементов;
- honeypot-системы;
- размещение информации картинками.

В работе «Применение технологии Honeypot для изучения поведения веб-роботов» [3] авторы А.А. Менщиков, Ю.А. Гатчин, А.В. Комарова более подробно объясняют значимость использования honeypot-систем для предотвращения угроз парсинга. Авторы вводят определение honeypot как ресурс-приманка, позволяющий не блокировать действия злоумышленников, и изучать стратегии их поведения. Конкретная реализация может кардинально отличаться в зависимости от целей и задач и представлять из себя как отдельный ресурс, так и динамическую систему, имитирующую реальный веб-сайт. Авторы работы провели исследование для выявления ботов и определения их поведения. Они пришли к выводу, что у большинства ботов отсутствует динамический анализ собранной информации. Также это исследование показывает, что honeypot-системы эффективны в обнаружении ботов и позволяют анализировать их поведение с целью дальнейшего противодействия им.

В анализе также рассмотрены работы, посвященные описанию технологии Honeypot. Работа «Использование Honeypot для привлечения внимания и поиска злоумышленников» [4] за авторством А.А. Сухова и В.В. Гармонова описывает все типы honeypot-систем, их достоинства и правила размещения. Авторы вводят еще одно определение honeypot. Honeypot – это системы безопасности, которые привлекают кибератаки и отслеживают их действия в защищенной, изолированной и контролируемой среде. Honeypot могут отвлекать потенциальные атаки от критически важных ресурсов и действовать как платформа для сбора информации об атакующих и их тактиках, техниках и процедурах. В исследовании авторы приводят типы honeypot и их определения:

1) Honeypot с низким уровнем взаимодействия. Он предназначен для имитации ограниченного числа сервисов, не сложен в реализации.

2) Honeypot с высоким уровнем взаимодействия. Он предлагает сложную, реалистичную среду, поскольку в состав ловушки могут входить дополнительные системы, базы данных и процессы.

3) Гибридный Honeypot. Данный вариант имитирует элементы прикладного уровня, но не имеет операционной системы.

В работе выделяю следующие достоинства honeypot-систем:

- раннее обнаружение атак;

- сбор разведывательной информации и анализ методов атак;
- обнаружение новых угроз;
- улучшение уровня безопасности и снижение рисков.

Авторы отмечают, что в мировой практике Honeypot используются для соблюдения специфических индустриальных требований. Их используют как часть комплексной и более широкой стратегии безопасности для обнаружения и реагирования на угрозы. Для того чтобы Honeypot был безопасен и эффективен в работе определяют следующие правила:

1) Он должен быть размещен в пределах сети, чтобы предложить потенциальным атакующим цель, при этом выглядя как обычная служба.

2) Он должен быть изолирован от остальной сети и не содержать важной информации.

3) Его журналы должны регулярно проверяться.

4) Honeypot должен управляться опытным персоналом и иметь регулярное обслуживание.

5) Он должен быть интегрируемым с другими мерами безопасности.

6) Юридические последствия использование Honeypot должны быть учтены, так как некоторые страны имеют строгие законы относительно мониторинга и перехвата связи.

Кроме отечественных научных работ были рассмотрены и зарубежные работы. Они показали, что интерес к технологии honeypot остается стабильно высоким. Это подтверждают значительное число публикаций, появившихся по данной теме за последние годы. В работе авторов A. Javadpour, F. Ja’fari, T. Taleb, M. Shojafar и C. Benzaïd «A comprehensive survey on cyber deception techniques to improve honeypot performance» [5] представлен масштабный обзор исследований в области honeypot, охватывающий последние два десятилетия. Целью работы стало систематизировать накопленные знания и обратить внимание на перспективные подходы и направления, которые пока остаются недостаточно изученными. Авторы детально описали существующие классификации honeypot-систем, а также различные методы обмана злоумышленников. К таким методам относятся расширенная имитация, манипуляции с базами данных и перенаправление трафика. Помимо этого, значительное внимание было уделено вопросам оптимизации приманок, моделированию атак и способам оценки эффективности honeypot-систем.

Данное исследование представляет ценность прежде всего как теоретическая база, на которую можно опираться при дальнейшей работе в этой области. Авторы предлагают ряд подходов к совершенствованию существующих систем защиты, что особенно актуально с точки зрения повышения точности и оперативности обнаружения вредоносной активности. Описанные методики могут послужить основой для создания более совершенных решений, способных выявлять угрозы на ранних стадиях и обеспечивать более надежную защиту.

В работе «A survey of contemporary open-source honeypots, frameworks, and tools» [6] авторы N. Ilg, P. Duplys, D. Sisejkovic и M. Menth обращаются к проблеме современных киберугроз и роли honeypot-технологий в противодействии им. На основе анализа большого числа исследований и решений с открытым исходным кодом авторы предлагают следующую классификацию honeypot-систем:

- как средство раннего обнаружения атак;
- как инструмент для сбора образцов вредоносного ПО;
- как способ изучения методов атак и уязвимостей.

Отдельно авторы подчеркивают роль технологий контейнеризации, которые существенно упрощают развертывание и автоматизацию honeypot-систем. Вместе с тем они указывают на сопутствующие риски, в частности на необходимость обеспечения безопасности самих контейнерных сред.

В рамках эксперимента, направленного на сравнительный анализ honeypot-решений, отдельное внимание было уделено подбору критериев оценки. Они выбирались исходя из того, насколько полно каждый из них способен отразить функциональную и техническую сторону системы в ситуации, когда существует реальная угроза безопасности. Для проведения анализа были определены следующие критерии:

1) Скорость обнаружения вторжений.
2) Уровень ложных срабатываний.
3) Объем и качество собираемых данных.
4) Устойчивость honeypot-системы.
5) Простота внедрения.
6) Производительность.

Данные критерии в совокупности охватывают наиболее значимые характеристики honeypot-решений, на которые следует обращать внимание при сравнительной оценке.

В ходе эксперимента использовалось несколько вариантов веб-форм, отличающихся механизмами защиты. Одна форма содержала скрытые honeypot-поля, предназначенные для обнаружения автоматического заполнения. Другие формы включали механизмы CAPTCHA, реализованные с использованием сервисов Google и Cloudflare. Схема тестового стенда изображена на Рисунке 2. Такое разделение позволяет сравнить эффективность различных подходов к защите форм от автоматизированных запросов в одинаковых условиях тестирования.

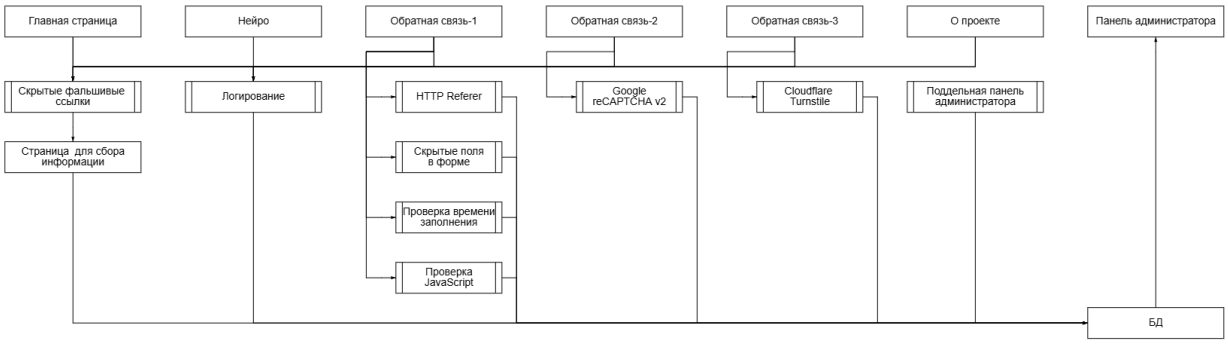

Рисунок 2 - Схема тестового стенда

Для проведения эксперимента были разработаны боты, имитирующие различные типы поведения. В тестовый набор вошли простые скриптовые боты, выполняющие базовые HTTP-запросы без эмуляции браузера, схемы которых изображены на Рисунке 3.

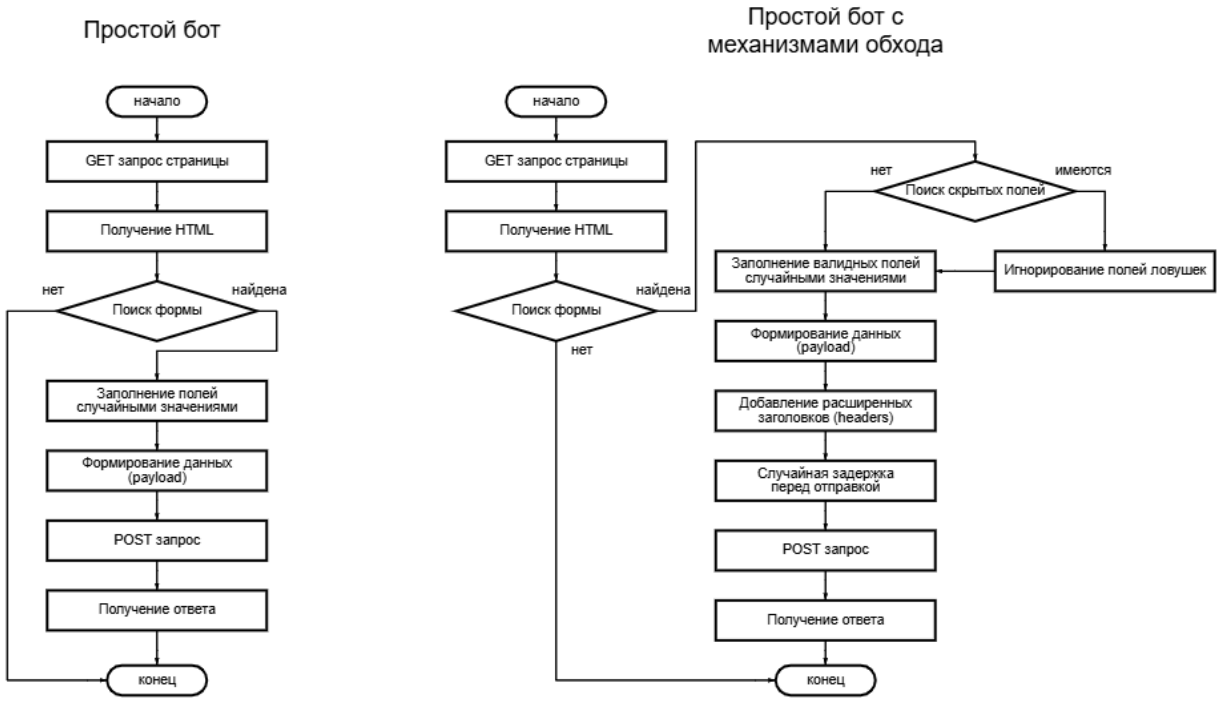

Рисунок 3 - Схемы простых ботов

И более сложные headless-боты, способные имитировать действия реального пользователя в браузерной среде, схемы которых изображены на Рисунке 4. Использование различных моделей поведения позволяет оценить устойчивость механизмов защиты веб-форм к автоматизированным атакам разного уровня сложности.

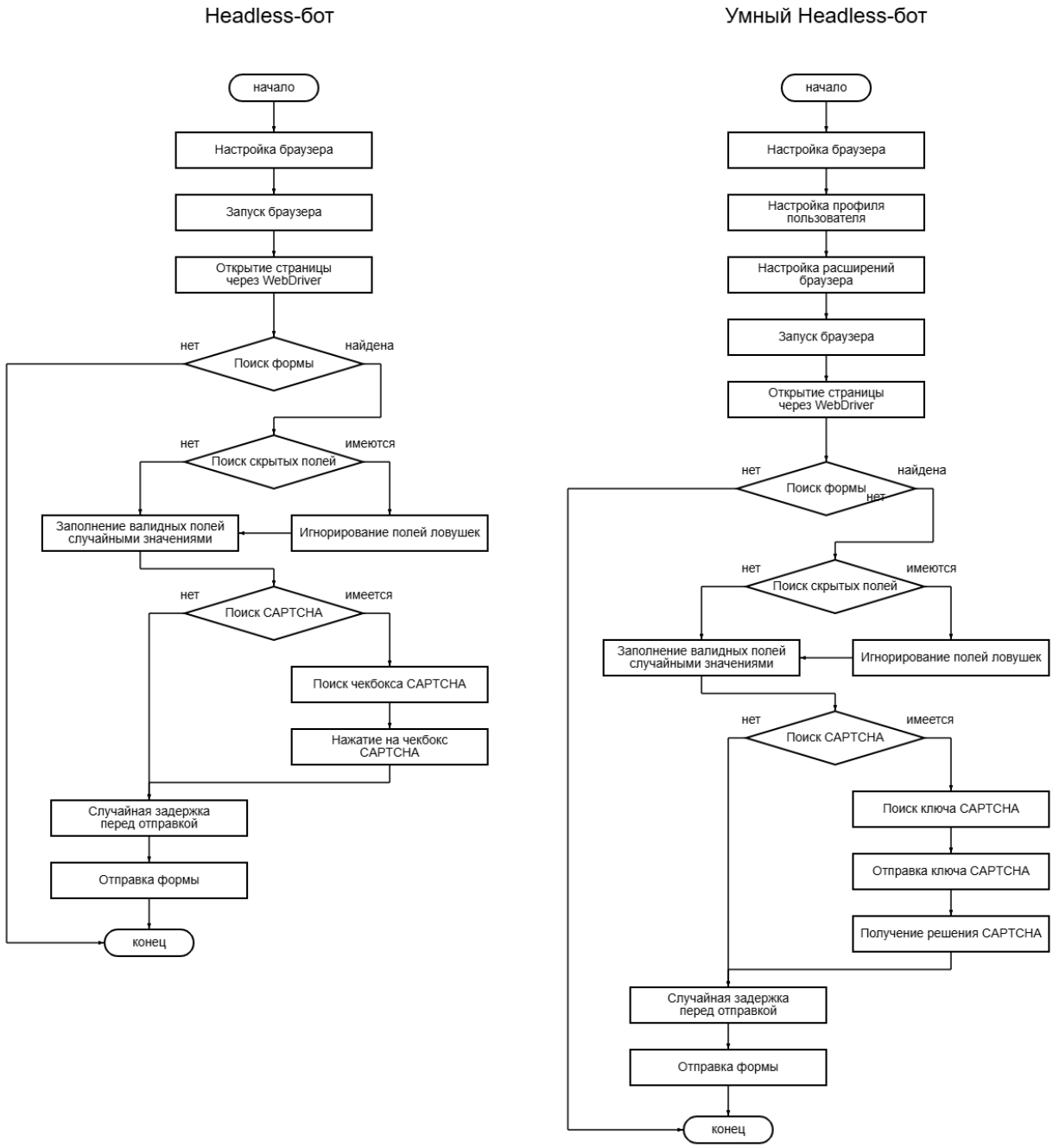

Рисунок 4 - Схемы headless-ботов

Для проведения тестирования был написан небольшой скрипт на языке программирования Python, который в автоматическом режиме запускал ботов для тестирования разных форм. В рамках эксперимента каждый бот запускался одну тысячу раз на каждую из разработанных форм. Такой подход позволил получить достаточный объем данных для последующего анализа эффективности honeypot-решений. Повторение большого количества попыток отправки форм также помогло снизить влияние случайных факторов и повысить достоверность результатов эксперимента. Таблица 1 содержит сравнительные результаты по 1000 запросам для пяти категорий ботов.

Таблица 1 - Сравнение частоты срабатывания ловушек на 1000 запросов

Тип ловушки	Простой бот	Простой бот с механизмами обхода	Headless-бот	Headless-бот с реальным профилем пользователя	Умный Headless-бот
HTTP Referer	1000	0	0	0	0
Скрытое поле (input)	1000	0	0	0	0
Скрытое поле (textarea)	1000	0	0	0	0
Проверка на выполнение JavaScript	1000	1000	0	0	0
Проверка времени заполнения формы	1000	98	21	19	0
Google reCAPTCHA v2	1000	1000	986	921	263
Cloudflare Turnstile	1000	1000	998	987	84

По результатам проведенного анализа можно утверждать, что простые honeypot-решения, показывают высокую эффективность только против самых простых ботов, а при переходе к более продвинутым ботам их эффективность падает до нуля. Из ловушек можно выделить проверку выполнения JavaScript, которая полностью блокирует простых ботов, но оказывается неэффективной против ботов, способных выполнять клиентские скрипты. А также выделить проверку времени заполнения формы. Видно, что её эффективность постепенно снижается по мере усложнения ботов. Таким образом, можно сделать вывод, что классические honeypot-решения демонстрируют слабую эффективность против современных ботов. Поэтому целесообразно рассмотреть более продвинутые механизмы защиты, такие как CAPTCHA.

Системы CAPTCHA продемонстрировали значительно более высокие результаты по сравнению с honeypot-решениями. В эксперименте были рассмотрены решения Google reCAPTCHA v2 и Cloudflare Turnstile. Оба механизма показали высокую эффективность против простых HTTP-ботов и ботов с базовыми механизмами обхода, обеспечивая 100% обнаружение автоматизированных запросов. Для обычного headless-бота Google reCAPTCHA v2 показал 98,6% блокировок, тогда как Cloudflare Turnstile показал 99,8%. Как показано на Рисунке 5, CAPTCHA пропустила лишь первые несколько запросов, после чего начала стабильно блокировать последующие попытки отправки формы.

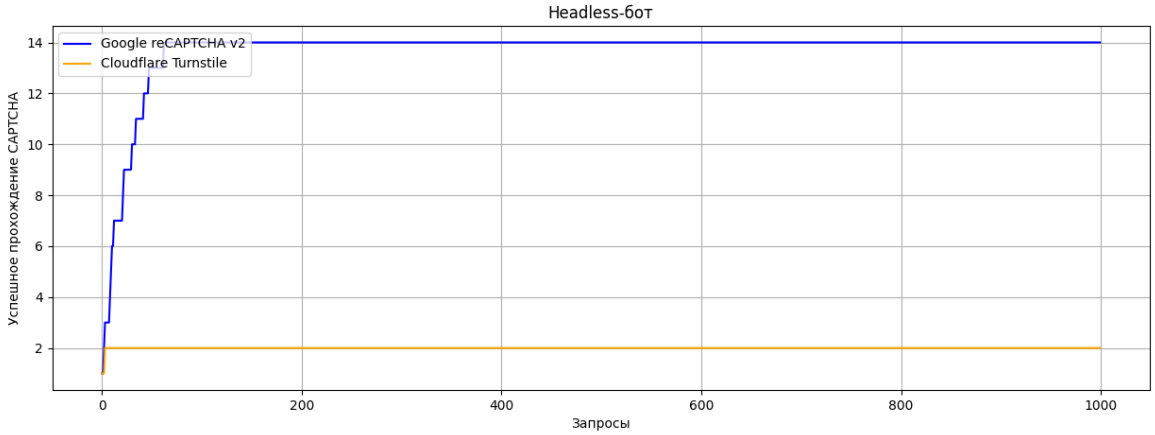

Рисунок 5 - Статистика прохождения CAPTCHA Headless-ботом

При использовании headless-бота с реальным браузерным профилем эффективность механизмов CAPTCHA несколько снижается. В данном случае Google reCAPTCHA v2 продемонстрировала уровень обнаружения 92,1%, тогда как Cloudflare Turnstile 98,7%. Это может быть связано с тем, что использование реального профиля браузера позволяет боту применять сохраненные cookies, параметры окружения и другие характеристики, приближающие его поведение к поведению реального пользователя. Как показано на Рисунке 6, данный тип бота успешно проходит проверку CAPTCHA на протяжении большего числа запросов, после чего эффективность прохождения постепенно снижается.

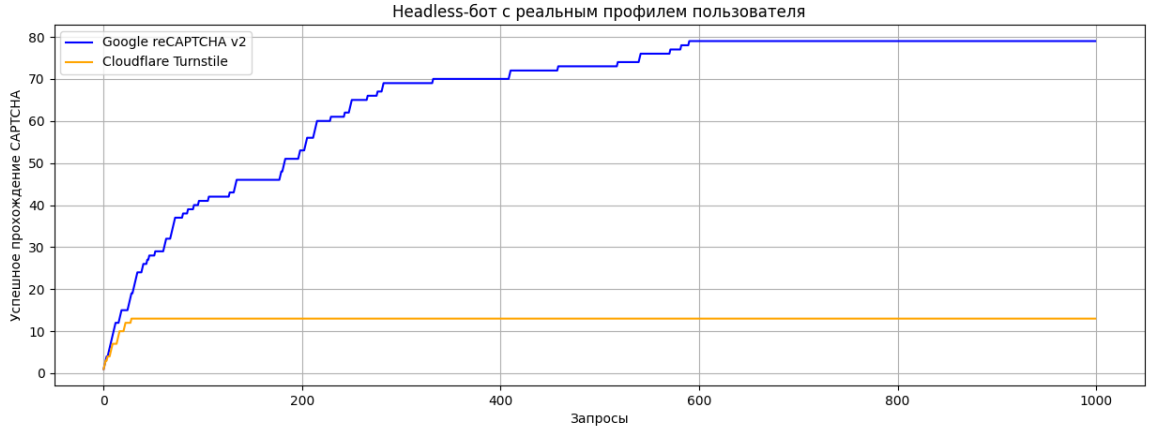

Рисунок 6 - Статистика прохождения CAPTCHA Headless-ботом с реальным профилем пользователя

Наиболее заметное снижение эффективности наблюдается при использовании умного headless-бота. В данном сценарии Google reCAPTCHA v2 зафиксировала 26,3% срабатываний, тогда как Cloudflare Turnstile зафиксировал только 8,4%. Это свидетельствует о том, что при реализации более сложных алгоритмов автоматизации, включающих имитацию пользовательского поведения, использование реального браузерного профиля, а также механизмы обхода CAPTCHA, современные защитные механизмы можно частично или полностью обойти. На Рисунке 7 представлены случаи успешного прохождения CAPTCHA данным типом бота.

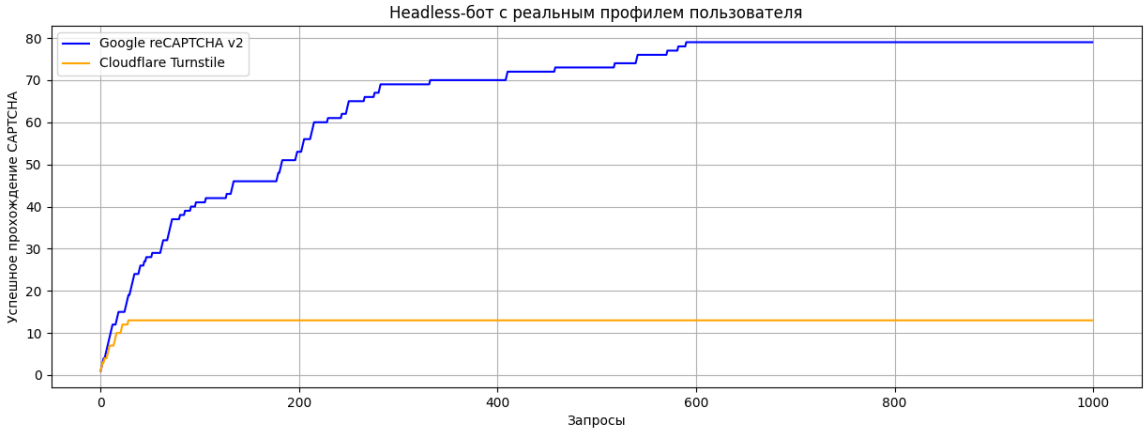

Рисунок 7 - Статистика прохождения CAPTCHA умным headless-ботом

Для оценки эффективности различных защитных механизмов веб-форм представлена Таблица 2, в которой обобщены результаты проведенного анализа.

Таблица 2 - Оценка эффективности ловушек для веб-формы

Тип ловушки	Скорость обнаружения	Ложные срабатывания	Объем и качество данных	Устойчивость	Простота внедрения	Производительность
HTTP Referer	Высокая	Низкие	Низкий	Низкая	Простое	Высокая
Скрытое поле (input)	Высокая	Низкие	Низкий	Низкая	Простое	Высокая
Скрытое поле (textarea)	Высокая	Низкие	Низкий	Низкая	Простое	Высокая
Проверка на выполнение JavaScript	Высокая	Низкие	Средний	Средняя	Простое	Высокая
Проверка времени заполнения формы	Средняя	Средние	Средний	Низкая	Простое	Высокая
Google reCAPTCHA v2	Средняя	Средние	Высокий	Высокая	Простое	Средняя
Cloudflare Turnstile	Средняя	Низкие	Высокий	Высокая	Простое	Средняя

Результаты показывают, что системы CAPTCHA обеспечивают значительно более высокий уровень обнаружения автоматизированных запросов по сравнению с простыми

honeypot-механизмами. При этом следует отметить, что поведенческие CAPTCHA, такие как Cloudflare Turnstile, способны эффективно перекрывать задачи, решаемые honeypot-ловушками, при этом не создавая дополнительных неудобств для пользователей, как это наблюдается при использовании Google reCAPTCHA v2. Это делает подобные решения наиболее предпочтительными для защиты веб-форм.

Список литературы

1. Сергановский А. А., Романович В. Н. Анализ ханипотов с открытым исходным кодом // Успехи в науке и образовании 2023 : сб. ст. II Междунар. науч.-исслед. конкурса. Пенза, 10 дек. 2023 г. Пенза : Наука и Просвещение, 2023. С. 76–80. EDN QPPZFM.
2. Демина Р.Ю., Шукралиева Д.Э., Ференс Р.Р. Защита веб-контента от несанкционированного автоматизированного сбора информации // Проблемы проектирования, применения и безопасности информационных систем в условиях цифровой экономики : материалы XXII Междунар. науч.-практ. конф. Ростов-на-Дону, 21–22 нояб. 2022 г. Ростов н/Д: Ростовский государственный экономический университет «РИНХ», 2022. С. 70–75. EDN AJLODZ.
3. Менщиков А.А., Гатчин Ю.А., Комарова А.В. Применение технологии honeypot для изучения поведения веб-роботов // Проблемы информационной безопасности : материалы VII Всерос. заоч. интернет-конф. Ростов-на-Дону, 20–21 февр. 2018 г. Ростов н/Д : АзовПринт, 2018. С. 45–48. EDN POPDGV.
4. Сухов А.А., Гармонов В.В. Использование Honeypot для привлечения внимания и поиска злоумышленников // Высокотехнологичное право: точка бифуркации: материалы V Междунар. науч.-практ. конф. : в 3 ч. Москва – Красноярск, 15–16 февр. 2024 г. Москва: Национальный исследовательский университет «МИЭТ», 2024. С. 32–37. EDN NGXWRZ.
5. Javadpour A., Ja'fari F., Taleb T., Shojafar M., Benzaid C. A comprehensive survey on cyber deception techniques to improve honeypot performance // Computers & Security. 2024. Vol. 140, no. 4. P. 103792. DOI: 10.1016/j.cose.2024.103792.
6. Ilg N., Duplys P., Sisejkovic D., Menth M. A survey of contemporary open-source honeypots, frameworks, and tools // Journal of Network and Computer Applications. 2023. Vol. 220. P. 103737. DOI: 10.1016/j.jnca.2023.103737. EDN KMKMPA.

References

1. Serganovsky A. A., Romanovich V. N. Analysis of open-source honeypots // Advances in Science and Education 2023: Coll. Art. of the II Int. Research Competition. Penza, December 10, 2023. Penza: Science and Education, 2023. pp. 76–80. EDN QPPZFM.
2. Demina R. Yu., Shukralieva D. E., Ferens R. R. Protecting web content from unauthorized automated collection of information // Problems of design, application, and security of information systems in the digital economy: Proc. of the XXII Int. Research and Practical Conf. Rostov-on-Don, November 21–22. 2022 Rostov n / D: Rostov State University of Economics "RINH", 2022. pp. 70-75. EDN AJLODZ.
3. Menshchikov A. A., Gatchin Yu. A., Komarova A. V. Using honeypot technology to study the behavior of web robots // Problems of information security: Proc. 7th All-Russian

correspondence internet conf. Rostov-on-Don, February 20-21, 2018 Rostov n / D: AzovPrint, 2018. pp. 45-48. EDN POPDGV.

4. Sukhov A. A., Garmonov V. V. Using honeypot to attract attention and find intruders // High-tech law: bifurcation point: Proc. V Int. scientific-practical. conf. : at 3 o'clock. Moscow - Krasnoyarsk, February 15-16, 2024. Moscow: National Research University of Electronic Technology, 2024. pp. 32-37. EDN NGXWRZ.
5. Javadpour A., Ja'fari F., Taleb T., Shojafar M., Benzaid C. A comprehensive survey on cyber deception techniques to improve honeypot performance // Computers & Security. 2024. Vol. 140, no. 4. p. 103792. DOI: 10.1016/j.cose.2024.103792.
6. Ilg N., Duplys P., Sisejkovic D., Menth M. A survey of contemporary open-source honeypots, frameworks, and tools // Journal of Network and Computer Applications. 2023. Vol. 220. P. 103737. DOI: 10.1016/j.jnca.2023.103737. EDN KMKMPA.

Хамитов И.А. Многокритериальная модель выбора структуры GPU-вычислительной системы для задач искусственного интеллекта // Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч.1 с. 57–61

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.272.2:004.8

МНОГОКРИТЕРИАЛЬНАЯ МОДЕЛЬ ВЫБОРА СТРУКТУРЫ GPU-ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ЗАДАЧ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Хамитов И.А.
ФГБОУ ВО «МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ», Москва, Россия (119454, г. Москва, пр-т Вернадского, д. 78, стр. 4), e-mail: ilnur_khamitov02@mail.ru

В статье рассматривается задача выбора структуры вычислительной системы на базе графических процессоров для прикладных задач искусственного интеллекта. Обосновывается, что экспертное принятие архитектурных решений и сравнение оборудования только по пиковой производительности не позволяют надежно оценить будущую эффективность системы в условиях крупной AI-инфраструктуры. Предлагается многокритериальная модель ранжирования GPU-конфигураций, учитывающая производительность, стоимость, энергопотребление, объем памяти, масштабируемость, задержку обработки и прикладные метрики LLM-нагрузок.

Ключевые слова**:** Kubernetes, GPU, вычислительная система, искусственный интеллект, LLM, энергоэффективность, многокритериальная оценка, дата-центр, стоимость владения.

MULTI-CRITERIA MODEL FOR SELECTING THE STRUCTURE OF A GPU-BASED COMPUTING SYSTEM FOR ARTIFICIAL INTELLIGENCE TASKS

Khamitov I.A.
MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY, Moscow, Russia (119454, Moscow, avenue Vernadsky, 78, b. 4), e-mail:

The article addresses the problem of selecting the structure of GPU-based computing systems for artificial intelligence workloads. It shows that expert-based decisions and equipment comparison by peak performance alone are insufficient for estimating practical efficiency of large AI infrastructures. A multi-criteria model is proposed for ranking GPU configurations using performance, cost, power consumption, memory capacity, scalability, latency and LLM-oriented workload metrics.

Keywords: GPU, computing system, artificial intelligence, LLM, energy efficiency, multi-criteria evaluation, data center, total cost of ownership.

Введение

Рост практического применения искусственного интеллекта привел к тому, что вычислительная инфраструктура стала одним из ключевых факторов внедрения AI-сервисов. Обучение моделей, инференс больших языковых моделей, мультимодальная обработка данных и корпоративные интеллектуальные помощники требуют специализированных серверов с графическими ускорителями. При этом выбор такой инфраструктуры связан не только с достижением требуемой скорости вычислений, но и с ограничениями по бюджету, энергопотреблению, охлаждению и последующей эксплуатации.

Значимость энергетического фактора усиливается по мере роста дата-центров. Международное энергетическое агентство прогнозирует, что мировое потребление

электроэнергии дата-центрами может увеличиться примерно с 485 ТВт·ч в 2025 году до 950 ТВт·ч к 2030 году, а потребление центров, ориентированных на AI, будет расти еще быстрее [1]. В этих условиях ошибка при выборе структуры вычислительной системы может стать не только технической, но и экономической проблемой: проект получает завышенные расходы на питание и охлаждение, а расчетная окупаемость ухудшается.

В инженерной практике архитектура GPU-системы нередко определяется экспертно: по опыту специалистов, доступности оборудования, рекомендациям поставщика или сравнению отдельных паспортных характеристик. Такой подход удобен при первичном отборе решений, но он плохо формализует компромиссы между производительностью, стоимостью, энергией, памятью и качеством обслуживания запросов. Например, сервер с большей теоретической производительностью может уступить более сбалансированной конфигурации при реальной LLM-нагрузке, если обладает меньшим объемом памяти, высокой задержкой или слабым межпроцессорным обменом.

Целью работы является разработка подхода к выбору структуры GPU-вычислительной системы, при котором альтернативные конфигурации оцениваются не по одному показателю, а по совокупности взаимосвязанных критериев. Такой подход позволяет сделать выбор более прозрачным и снизить риск неэффективных капитальных и эксплуатационных затрат.

Исследование

Проблема экспертного выбора архитектуры

Проектирование вычислительной системы для задач искусственного интеллекта предполагает выбор типа GPU, количества ускорителей, процессорной платформы, объема оперативной и видеопамяти, межсоединения, сетевой инфраструктуры, системы охлаждения и параметров электропитания. Эти элементы образуют единую структуру, поэтому изменение одного параметра влияет на остальные. Увеличение числа GPU повышает потенциальную производительность, но одновременно увеличивает стоимость, энергопотребление и требования к межпроцессорному обмену.

Однокритериальный выбор не отражает этой взаимосвязи. Сравнение по FLOPS или Tensor TFLOPS показывает только верхнюю границу вычислительных возможностей и не описывает, насколько эффективно конкретная нагрузка сможет использовать оборудование. Разработчики TOP500 прямо отмечают, что результат LINPACK не является универсальной характеристикой системы, поскольку один числовой показатель не способен описать все стороны производительности [2]. Для задач машинного обучения более прикладную оценку дают MLPerf-бенчмарки, однако даже они не заменяют технико-экономического выбора конкретной инфраструктуры под заданные ограничения [3].

Следовательно, при выборе GPU-системы важно перейти от экспертного сравнения отдельных характеристик к модели, которая связывает требования задачи с параметрами архитектуры и позволяет ранжировать несколько альтернатив.

Набор критериев для оценки конфигурации

Каждая рассматриваемая конфигурация может быть описана набором технических и экономических параметров: достижимая производительность, среднее энергопотребление, стоимость закупки, объем GPU-памяти, коэффициент масштабируемости, задержка ответа и

скорость генерации токенов. Для сопоставления альтернатив предлагается рассчитывать нормированные частные критерии, представленные в Таблице 1.

Таблица 1 – Частные критерии оценки GPU-конфигурации

Критерий	**Смысл показателя**	**Роль при выборе**
Производительность	Насколько система покрывает требуемый объем вычислений	Позволяет исключить недостаточно мощные решения
Энергоэффективность	Отношение полезной производительности к потребляемой мощности	Снижает эксплуатационные расходы и нагрузку на инженерную инфраструктуру
Стоимость	Закупочная или расчетная стоимость конфигурации	Влияет на доступность проекта и срок окупаемости
Память	Достаточность GPU-памяти для модели, кэша и контекста	Критична для LLM, RAG и мультимодальных задач
Масштабируемость	Сохранение прироста производительности при добавлении GPU	Важна для кластеров и распределенной обработки
Задержка	Время получения ответа на пользовательский запрос	Определяет качество интерактивных AI-сервисов
LLM-эффективность	Скорость генерации токенов и интенсивность обслуживания запросов	Описывает практическую пригодность системы для генеративного ИИ

Нормирование необходимо потому, что исходные параметры измеряются в разных единицах: ваттах, рублях, миллисекундах, гигабайтах, токенах в секунду. После приведения критериев к сопоставимому виду можно рассчитать интегральный показатель пригодности конфигурации для заданной нагрузки.

$I = w_1K_p + w_2K_e + w_3Kc + w_4Km + w_5Ks + w_6Kl + w_7Kllm,$

где K_p — критерий производительности; K_e — критерий энергоэффективности; Kc — критерий стоимости; Km — критерий обеспеченности памятью; Ks — критерий масштабируемости; Kl — критерий задержки; Kllm — критерий эффективности LLM-нагрузки; $w_1 \ldots w_7$ — весовые коэффициенты, сумма которых равна единице.

Идея модели состоит в том, что лучшей считается не обязательно самая дорогая или самая производительная система, а та, которая наиболее полно соответствует прикладному сценарию. Для интерактивного инференса увеличивается значимость задержки и скорости генерации; для пакетной обработки — пропускной способности, энергоэффективности и стоимости; для длинного контекста — объема видеопамяти и устойчивости при росте входных данных.

Алгоритм применения модели

Применение модели включает пять этапов. На первом этапе задаются ограничения: минимальная производительность, максимальный бюджет, допустимая мощность, требуемый объем GPU-памяти, предельная задержка и тип нагрузки. На втором этапе формируется перечень альтернативных конфигураций, которые потенциально могут быть использованы в проекте.

На третьем этапе выполняется предварительная фильтрация. Если конфигурация не удовлетворяет жестким условиям, например не имеет достаточного объема видеопамяти для размещения модели, она исключается до расчета итогового показателя. Такой шаг предотвращает попадание в ранжирование формально дешевых, но непригодных решений.

На четвертом этапе для оставшихся вариантов рассчитываются частные критерии. Для вычислительных показателей могут использоваться паспортные характеристики, результаты бенчмарков, эмпирические коэффициенты использования, а также прикладные метрики: количество токенов в секунду, средняя задержка и расчетная интенсивность обслуживания запросов. На пятом этапе рассчитывается интегральная оценка, после чего конфигурации сортируются по убыванию итогового значения.

Преимущество такого алгоритма заключается в объяснимости результата. Итоговый выбор можно обосновать не общей фразой о большей мощности оборудования, а конкретными причинами: достаточной памятью, лучшим соотношением производительности и мощности, меньшей стоимостью обработки запроса или более высокой устойчивостью к масштабированию.

Практическая значимость

Предложенная модель может использоваться при подготовке технико-экономического обоснования закупки GPU-серверов, модернизации корпоративной AI-платформы, сравнении предложений поставщиков и проектировании дата-центра. Особенно важным является учет энергопотребления, поскольку эксплуатационные затраты становятся значимой частью жизненного цикла вычислительной системы.

Модель также снижает риск неверного масштабирования. Недостаточная конфигурация приводит к невыполнению требований по качеству сервиса и необходимости дополнительной закупки. Избыточная конфигурация создает лишнюю капитальную нагрузку, увеличивает потребление электроэнергии и может сделать проект экономически непривлекательным еще до запуска.

Заключение

В статье рассмотрена проблема выбора структуры GPU-вычислительной системы для задач искусственного интеллекта. Показано, что экспертный подход и сравнение по пиковой производительности не обеспечивают достаточной обоснованности при проектировании крупной AI-инфраструктуры.

Предложена многокритериальная модель, в которой альтернативные конфигурации оцениваются по совокупности производительных, энергетических, стоимостных и прикладных показателей. В отличие от однокритериального выбора модель позволяет учитывать специфику нагрузки: интерактивный инференс, пакетную обработку и задачи с увеличенным контекстом.

Практическая ценность подхода заключается в возможности заранее оценить компромиссы между стоимостью, энергопотреблением, памятью и качеством обслуживания. Это повышает прозрачность архитектурных решений, снижает риск избыточных расходов и позволяет выбирать вычислительную систему с учетом реальных условий эксплуатации.

Список литературы

1. Международное энергетическое агентство. Ключевые вопросы энергетики и искусственного интеллекта: Краткое содержание [Электронный ресурс]. — URL: https://www.iea.org/reports/key-questions-on-energy-and-ai/executive-summary (дата обращения: 10.05.2026).
2. ТОП500. Тест LINPACK [Электронный ресурс]. — URL: https://top500.org/project/linpack/ (дата обращения: 10.05.2026).
3. МЛКоммонс. MLPerf Inference: Датацентр [Электронный ресурс]. — URL: https://mlcommons.org/benchmarks/inference-datacenter/ (дата обращения: 10.05.2026).
4. Уильямс С., Уотерман А., Паттерсон Д. Roofline: содержательная визуальная модель производительности для многоядерных архитектур // Communications of the ACM. — 2009. — Том. 52. — № 4. — С. 65–76.
5. Хеннесси Дж. Л., Паттерсон Д. А. Компьютерная архитектура: количественный подход. - 6-е изд. - Кембридж: Морган Кауфманн, 2019.
6. Саати Т. Л. Процесс аналитической иерархии. - Нью-Йорк: МакГроу-Хилл, 1980.
7. Грин500. Списки Green500 [Электронный ресурс]. — URL: https://top500.org/lists/green500/ (дата обращения: 14.05.2026).
8. МЛКоммонс. MLPerf Training Benchmark [Электронный ресурс]. — URL: https://mlcommons.org/benchmarks/training/ (дата обращения: 15.05.2026).

References

1. International Energy Agency. Key Questions on Energy and AI: Executive summary [Электронный ресурс]. — URL: https://www.iea.org/reports/key-questions-on-energy-and-ai/executive-summary (дата обращения: 10.05.2026).
2. TOP500. The LINPACK Benchmark [Электронный ресурс]. — URL: https://top500.org/project/linpack/ (дата обращения: 10.05.2026).
3. MLCommons. MLPerf Inference: Datacenter [Электронный ресурс]. — URL: https://mlcommons.org/benchmarks/inference-datacenter/ (дата обращения: 10.05.2026).
4. Williams S., Waterman A., Patterson D. Roofline: an insightful visual performance model for multicore architectures // Communications of the ACM. — 2009. — Vol. 52. — № 4. — P. 65–76.
5. Hennessy J. L., Patterson D. A. Computer Architecture: A Quantitative Approach. — 6th ed. — Cambridge: Morgan Kaufmann, 2019.
6. Saaty T. L. The Analytic Hierarchy Process. — New York: McGraw-Hill, 1980.
7. Green500. Green500 Lists [Электронный ресурс]. — URL: https://top500.org/lists/green500/ (дата обращения: 14.05.2026).
8. MLCommons. MLPerf Training Benchmark [Электронный ресурс]. — URL: https://mlcommons.org/benchmarks/training/ (дата обращения: 15.05.2026).

Нестеров А.В. Оптимизация продуктовых решений на основе предиктивной аналитики и машинного обучения// Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч. 1 с. 62–69

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.8:658.5

ОПТИМИЗАЦИЯ ПРОДУКТОВЫХ РЕШЕНИЙ НА ОСНОВЕ ПРЕДИКТИВНОЙ АНАЛИТИКИ И МАШИННОГО ОБУЧЕНИЯ

Нестеров А.В., Дешко И.П. (научный руководитель)
ФГБОУ ВО «МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ», Москва, Россия (119454, г. Москва, пр-т Вернадского, д. 78, стр. 4), e-mail: sanchotar@gmail.com

В статье рассматриваются методы машинного обучения для оптимизации решений в продуктовом менеджменте. Проведён анализ ключевых задач продуктовой аналитики, где применение предиктивных моделей даёт наибольший эффект: прогнозирование оттока пользователей, оценка пожизненной ценности клиента, оптимизация воронки конверсии. Обоснованы требования к качеству данных и объяснимости моделей. Предложен подход к интеграции прогнозных сигналов в операционные процессы управления продуктом.

Ключевые слова**:** Продуктовый менеджмент, машинное обучение, предиктивная аналитика, прогнозирование оттока, пожизненная ценность клиента, оптимизация конверсии.

OPTIMIZATION OF PRODUCT DECISIONS BASED ON PREDICTIVE ANALYTICS AND MACHINE LEARNING

Nesterov A.V., Deshko I.P. (Academic Supervisor)
MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY, Moscow, Russia (119454, Moscow, avenue Vernadsky, 78, b. 4), e-mail: sanchotar@gmail.com

This paper examines machine learning methods for optimizing decisions in product management. An analysis is conducted of key product analytics tasks where predictive models yield the greatest effect: churn prediction, customer lifetime value estimation, and conversion funnel optimization. Requirements for data quality and model explainability are substantiated. An approach to integrating predictive signals into operational product management processes is proposed.

Keywords: Product management, machine learning, predictive analytics, churn prediction, customer lifetime value, conversion optimization.

Современный продуктовый менеджмент функционирует в условиях экспоненциального роста данных о поведении пользователей, но при этом большинство решений о развитии продукта по-прежнему принимается на основе запаздывающих метрик и качественных гипотез. Традиционные подходы, такие как анализ исторических данных, опросы пользователей и экспертные оценки, неизбежно страдают от субъективности и инерционности – они показывают, что уже произошло, но не позволяют оперативно реагировать на изменения в реальном времени. Согласно аналитическим обзорам, компании, внедряющие предиктивную аналитику в процессы управления продуктом, сокращают время реакции на изменения рыночной ситуации на 30–40% и повышают точность прогнозов ключевых показателей до 75–85% при правильно настроенных моделях машинного обучения [1, 2]. При этом важно понимать, что сам по себе прогноз – это лишь вероятностная оценка будущего, а ценность он приобретает только тогда, когда становится основой для конкретных управленческих

действий. Однако между получением аналитического инсайта и его операциональным применением в продуктовых решениях сохраняется значительный разрыв, который в данной работе предлагается сокращать за счёт интеграции прогнозных моделей непосредственно в контуры управления продуктом. Речь идёт не просто о добавлении «умных» дашбордов, а о создании замкнутого цикла, в котором модель не только предсказывает, но и инициирует действия, а затем оценивает их эффективность для последующего обучения.

Объектом исследования выступают процессы принятия решений в продуктовом менеджменте, связанные с управлением пользовательским опытом, оптимизацией воронок и планированием функциональности. Предметом исследования являются методы машинного обучения и предиктивной аналитики, применимые для повышения обоснованности и скорости продуктовых решений. Целью работы является выявление наиболее эффективных сценариев применения прогнозных моделей в продуктовой аналитике и формулирование требований к их интеграции в управленческий контур. Дополнительная задача – показать, какие организационные и технические барьеры возникают на пути внедрения таких моделей в реальных продуктовых командах, и предложить способы их преодоления. В отличие от исследований, фокусирующихся исключительно на точности алгоритмов, в данной работе акцент сделан на практической применимости и встраиваемости моделей в существующие процессы.

Продуктовый менеджмент как дисциплина традиционно опирается на цикл «гипотеза – эксперимент – анализ – решение». Этот цикл может быть существенно ускорен, если вместо ручного анализа исторических данных использовать модели машинного обучения, которые автоматически выявляют паттерны и формируют прогнозы. Например, вместо того чтобы раз в неделю смотреть отчёт о падении конверсии и начинать расследование причин, команда может получать автоматические сигналы об аномалиях в реальном времени с указанием наиболее вероятных факторов. Более того, модель может не только сигнализировать о проблеме, но и предлагать (или даже автоматически применять) корректирующие действия – изменить параметры A/B-теста, запустить коммуникацию с определённым сегментом пользователей или скорректировать приоритеты в бэклоге. В современной практике выделяются три ключевые задачи, где предиктивные модели показывают наибольшую эффективность: прогнозирование оттока пользователей (churn prediction), оценка пожизненной ценности клиента (customer lifetime value, LTV) и оптимизация воронки конверсии (conversion funnel optimization). Каждая из этих задач имеет свою специфику и требует различных подходов к построению моделей, различной частоты переобучения и разных стратегий валидации. Выбор именно этих трёх задач обусловлен тем, что они охватывают весь жизненный цикл взаимодействия пользователя с продуктом – от привлечения до удержания – и при этом имеют прямое влияние на ключевые бизнес-метрики.

Прогнозирование оттока является одной из наиболее изученных областей применения машинного обучения в управлении продуктом. Основная задача – по историческим данным о поведении пользователя предсказать вероятность того, что он прекратит использование продукта в заданный временной горизонт (обычно 7, 14 или 30 дней). Для построения таких моделей чаще всего используются логистическая регрессия, случайный лес и градиентный бустинг (XGBoost, LightGBM). Каждый из этих методов имеет свои преимущества: логистическая регрессия даёт высокую интерпретируемость, случайный лес устойчив к выбросам, а градиентный бустинг, как правило, обеспечивает наилучшую точность. Как

отмечается в исследованиях, градиентный бустинг обеспечивает высокую точность прогнозирования, позволяя выявлять сложные паттерны, которые однофакторный анализ может пропустить [3]. Качественный прогноз оттока требует не только учёта частоты и давности действий пользователя (recency, frequency, monetary – RFM-метрики), но и анализа поведенческих паттернов: снижение критических действий, изменение маршрутов в продукте, негативные сигналы в обратной связи. Например, если пользователь, который ранее ежедневно выполнял целевое действие, перестал заходить в приложение на протяжении пяти дней подряд, это более сильный сигнал, чем простое снижение частоты. Аналогично, если пользователь оставил негативный отзыв в поддержке или в социальных сетях, это также повышает вероятность оттока. При этом точность модели на тестовой выборке обычно составляет 70–85% в зависимости от качества исходных данных и выбранных признаков. Важно, что прогноз оттока сам по себе не является решением – он должен быть связан с набором удерживающих действий (таргетированная коммуникация, предложение скидки, изменение пользовательского опыта). В этом заключается ключевой принцип управляемой предиктивности: модель даёт сигнал, но действие должно быть определено отдельной политикой с учётом контекста и возможных побочных эффектов. Например, слишком агрессивные удерживающие кампании могут раздражать пользователей и приводить к обратному эффекту. Поэтому политика действий должна включать не только правила «если риск высок, то сделать X», но и механизмы A/B-тестирования самих удерживающих вмешательств.

Оценка пожизненной ценности клиента представляет собой более сложную прогнозную задачу, поскольку LTV зависит не только от поведения пользователя, но и от внешних факторов, включая сезонность, конкурентную среду и макроэкономические условия. Традиционные методы RFM-анализа дают лишь приблизительную оценку прошлой ценности, тогда как машинное обучение позволяет строить прогнозы будущей LTV на основе временных рядов поведения. Для этой цели применяются регрессионные модели, модели на основе скрытых марковских процессов и, в последнее время, нейросетевые подходы. Каждый метод имеет свою область применения: простые регрессии хороши для стабильных продуктов с предсказуемым поведением, нейросети требуют больших объёмов данных, но способны улавливать нелинейные зависимости. Исследования показывают, что градиентный бустинг и нейросетевые архитектуры (LSTM) существенно повышают точность прогнозов LTV по сравнению с линейной регрессией, достигая значений R^2 в диапазоне 0,94 при правильно подобранных гиперпараметрах и достаточном объёме данных [4]. Ключевая особенность прогнозирования LTV в контексте продуктового менеджмента заключается в том, что этот показатель является производным от множества продуктовых решений: изменение интерфейса, добавление или удаление функций, ценовая политика. Поэтому модель LTV должна переобучаться при каждом существенном изменении продукта, что накладывает требования к автоматизации циклов переобучения. В противном случае прогнозы быстро устаревают и вводят в заблуждение команду разработки. Например, если компания запускает новую платную подписку, то старая модель, обученная на данных до этого изменения, будет систематически занижать прогнозируемый LTV, поскольку не учитывает новый источник монетизации. Следовательно, необходимо настроить автоматическое обнаружение дрейфа данных и триггеры для переобучения – например, после каждого крупного релиза или еженедельно, если объём новых пользователей достаточно велик.

Оптимизация воронки конверсии – задача, в которой машинное обучение может применяться как для диагностики узких мест, так и для формирования рекомендаций по их устранению. Традиционные инструменты аналитики (например, построение воронок и анализ падения конверсии между этапами) показывают, где происходит потеря пользователей, но не объясняют почему. Модели машинного обучения способны заглянуть глубже: они могут анализировать сотни потенциальных факторов одновременно, выявляя те, которые статистически значимо коррелируют с отказом от перехода на следующий этап. Методы машинного обучения, в частности кластеризация и деревья решений, позволяют сегментировать пользователей по паттернам поведения и выявлять скрытые факторы, влияющие на переход к следующему этапу. Например, модель может обнаружить, что пользователи, пришедшие из определённого канала трафика, с высокой вероятностью покидают воронку на этапе регистрации из-за сложности формы, тогда как для других каналов этот фактор незначителен. Такой инсайт напрямую указывает на конкретное продуктовое решение: упростить форму регистрации именно для проблемного сегмента или изменить канальную стратегию. Более того, с помощью деревьев решений можно построить интерпретируемые правила – например, «если пользователь пришёл из канала X и время заполнения формы превышает 30 секунд, то вероятность конверсии на следующем шаге составляет менее 10%». Эти правила могут быть напрямую переданы команде разработки для реализации точечных улучшений. Дальнейшее развитие этого подхода – использование моделей для персонализации воронки в реальном времени, когда каждый следующий этап адаптируется под прогнозируемую вероятность успеха конкретного пользователя. Например, если модель предсказывает, что данный пользователь с высокой вероятностью бросит заполнение длинной анкеты, система может динамически сократить анкету или предложить альтернативный путь – войти через социальную сеть.

Однако применение машинного обучения в продуктовом менеджменте сталкивается с рядом системных ограничений, которые необходимо учитывать при проектировании соответствующих систем. Эти ограничения носят не только технический, но и организационный характер. Первое ограничение – качество и полнота данных. Модели машинного обучения критически зависят от того, насколько полно в данных отражены все значимые факторы. Если команда продукта не собирает события о неудачных попытках выполнения действия (например, ошибки валидации формы), то модель не сможет предсказывать их влияние на конверсию. Поэтому внедрение предиктивной аналитики должно начинаться с аудита событийной модели и обеспечения сбора всех релевантных сигналов. На практике это означает тесное взаимодействие продуктовых менеджеров, аналитиков и разработчиков – необходимо определить, какие именно пользовательские действия и их исходы имеют значение для прогнозирования. Второе ограничение – объяснимость решений. В отличие от рекламной оптимизации, где допустима «чёрная коробка» при условии положительной динамики метрик, в управлении продуктом менеджеры должны понимать, почему модель сделала тот или иной прогноз, чтобы принять обоснованное решение о действии. Это требует использования методов объяснимого искусственного интеллекта (XAI), таких как SHAP-значения или LIME, а также прозрачных, интерпретируемых моделей (например, деревья решений вместо глубоких нейросетей) там, где это возможно без существенной потери точности. Без объяснимости менеджеры не смогут доверять модели и, следовательно, не будут использовать её рекомендации в критически

важных решениях. Третье ограничение – динамическая нестабильность поведенческих паттернов. Пользовательское поведение меняется вместе с продуктом, и модель, обученная на данных до релиза новой функции, может давать систематически ошибочные прогнозы после него. Это требует организации непрерывного мониторинга качества моделей и автоматизированного переобучения при обнаружении дрейфа концепции (concept drift). Например, если после обновления дизайна главного экрана модель начинает ошибаться в прогнозах конверсии, необходимо либо переобучить её на новых данных, либо временно откатить использование модели до выяснения причин.

Сравнительный анализ существующих подходов к интеграции предиктивной аналитики в продуктовый менеджмент позволяет выделить три уровня зрелости таких систем. На первом уровне модели используются для пассивной отчётности: дашборд показывает прогнозы оттока или LTV, но решения принимаются менеджером вручную. Этот уровень характерен для большинства компаний, которые только начинают внедрять машинное обучение в продуктовую аналитику. На втором уровне модели встроены в процессы поддержки решений: система автоматически формирует рекомендации (например, «включить удерживающую кампанию для сегмента X») и предоставляет обоснование, но финальное действие утверждает человек. Такой подход позволяет сохранить контроль со стороны менеджера, но при этом существенно ускоряет принятие решений, так как аналитическая работа уже выполнена моделью. На третьем уровне реализуется замкнутый контур управления: прогнозная модель напрямую связана с исполнительными механизмами (автоматическая отправка пуш-уведомлений, изменение параметров A/B-теста, перераспределение ресурсов на разработку), при этом все действия фиксируются для последующего измерения их инкрементального эффекта. Третий уровень требует высокой степени доверия к модели и наличия предохранителей – лимитов на автоматические изменения, возможности быстрой отмены действий, а также механизмов для измерения эффекта от каждого автоматического решения. Большинство современных платформ продуктовой аналитики находятся на первом или переходном ко второму уровне, что создаёт значительный потенциал для развития в направлении автоматизированного управления продуктом.

Измерение эффективности продуктовых решений, принятых на основе прогнозов, является отдельной методологической проблемой. Классические методы атрибуции (last-click, first-click, линейная) неприменимы, поскольку решение может влиять на пользователя в долгосрочной перспективе, а не только на момент конверсии. Например, удерживающее сообщение, отправленное сегодня, может предотвратить отток, который произошёл бы через две недели. Простая атрибуция по последнему событию не сможет правильно приписать этот эффект. В этой связи предпочтительным подходом является инкрементальное измерение через рандомизированные контролируемые испытания (RCT): когда решение (например, запуск удерживающей коммуникации) применяется к случайно выбранной группе пользователей, а эффект оценивается как разница в ключевых метриках между этой группой и контрольной группой, не получившей воздействия. Такой подход позволяет отделить эффект именно продуктового решения от влияния внешних факторов и временных трендов. Важно, что модель машинного обучения, используемая для прогноза, не должна оценивать свой собственный эффект – для этого нужен независимый измерительный контур [5]. На практике это означает, что решения, принятые на основе прогнозов, должны рандомизироваться на уровне отдельных пользователей или сегментов, а результаты сравниваться с контрольной

группой, где решения принимались без использования модели (например, по стандартному протоколу). Только так можно получить несмещённую оценку того, насколько модель улучшает реальные бизнес-показатели по сравнению с существующим процессом.

Отдельного внимания заслуживают регуляторные ограничения, связанные с обработкой персональных данных при построении прогнозных моделей. В Российской Федерации действует Федеральный закон № 152-ФЗ «О персональных данных», который требует минимизации собираемых данных, получения согласия на обработку и обеспечения безопасности [6]. Применительно к прогнозным моделям это означает, что использование персональных идентификаторов (email, phone, device ID) должно быть ограничено, предпочтительна псевдонимизация, а сами модели не должны принимать решения, затрагивающие права пользователей, без возможности человеческого контроля. Например, если модель принимает решение о блокировке пользователя или о принудительном изменении тарифа, это должно происходить только с участием человека. В европейской практике аналогичные требования закреплены в GDPR, а также в Акте об искусственном интеллекте (EU AI Act), который классифицирует системы прогнозирования поведения пользователей как высокорисковые при определённых условиях. Это означает, что для таких систем требуется проведение оценки соответствия, ведение журналов событий и обеспечение прозрачности алгоритмов. Таким образом, архитектура любой системы, использующей машинное обучение для управления продуктом, должна с самого начала включать механизмы privacy-by-design: минимизацию данных, короткие сроки хранения, псевдонимизацию в аналитическом слое и полную аудитируемость всех автоматических решений. Без соблюдения этих требований внедрение предиктивной аналитики может привести к юридическим рискам и потере доверия со стороны пользователей. Более того, в некоторых юрисдикциях пользователи имеют право на объяснение решения, принятого алгоритмом, что накладывает дополнительные требования к интерпретируемости моделей.

Практическая реализация описанных подходов требует интеграции нескольких компонентов: системы сбора событий, хранилища данных, вычислительного слоя для обучения и инференса моделей и системы оркестрации действий. В качестве системы сбора событий могут выступать такие инструменты, как Segment, Snowplow или собственные разработки на основе Kafka. Хранилище данных обычно реализуется на базе Snowflake, BigQuery или Redshift. Для обучения моделей часто используются управляемые сервисы вроде AWS SageMaker, Google Vertex AI или Databricks. Оркестрация действий может быть построена на Apache Airflow, Temporal или специализированных решениях для автоматизации маркетинга и продуктовых экспериментов. Однако ключевым фактором успеха является не столько технологический стек, сколько выработка дисциплины: регулярное обновление моделей, валидация гипотез через эксперименты, прозрачная связь между прогнозом, действием и измеренным эффектом. Команды продуктового менеджмента, которые внедряют такую дисциплину, постепенно переходят от реактивного управления к проактивному, когда решения принимаются на основе прогнозов, а не постфактум анализа уже случившихся событий [7]. В долгосрочной перспективе это позволяет не только повысить ключевые метрики, но и сократить когнитивную нагрузку на менеджеров, освобождая их время для стратегического планирования и креативных задач, которые не могут быть автоматизированы. Именно в этом заключается конечная цель применения предиктивной аналитики в

продуктовом менеджменте: не заменить человека, а дать ему инструменты для принятия более качественных и своевременных решений.

При написании статьи были применены технологии искусственного интеллекта.

Список литературы

1. Ремизова, А. А. Поведенческая аналитика: анализ современного состояния и ее применение для решения задач бизнеса / А. А. Ремизова, Т. Р. Самигулин. – Текст: электронный // КиберЛенинка. – URL: https://cyberleninka.ru/article/n/povedencheskaya-analitika-analiz-sovremennogo-sostoyaniya-i-ee-primenenie-dlya-resheniya-zadach-biznesa (дата обращения: 02.04.2026).
2. Марчук, В. В. Оптимизация бизнес-процессов с использованием прогнозной аналитики, современных экономических решений и инноваций / В. В. Марчук, Н. А. Шатохин, Е. В. Сенотрусова, Н. С. Лапшин, В. О. Черненко. – Текст: электронный // КиберЛенинка. – URL: https://cyberleninka.ru/article/n/optimizatsiya-biznes-protsessov-s-ispolzovaniem-prognoznoy-analitiki-sovremennyh-ekonomicheskih-resheniy-i-innovatsiy (дата обращения: 02.04.2026).
3. 7 Advanced Machine Learning Models for Customer Insights. – 2025. – URL: https://madgicx.com/blog/advanced-machine-learning-models-for-customer-insights (дата обращения: 02.04.2026).
4. Top 5 Machine Learning Models for LTV Prediction. – 2025. – URL: https://www.artisangrowthstrategies.com/blog/top-5-machine-learning-models-for-ltv-prediction (дата обращения: 02.04.2026).
5. Влияние предиктивной аналитики на деятельность компаний. – Текст: электронный // КиберЛенинка. – URL: https://cyberleninka.ru/article/n/vliyanie-prediktivnoy-analitiki-na-deyatelnost-kompaniy (дата обращения: 02.04.2026).
6. Федеральный закон от 27.07.2006 № 152-ФЗ «О персональных данных» (последняя редакция). – Текст: электронный // КонсультантПлюс. – URL: https://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 02.04.2026).
7. Михайлов, Н. Д. Рыночные тенденции интернет-рекламы в России (2024–2025 гг.) / Н. Д. Михайлов. – Текст: электронный // КиберЛенинка. – 2025. – URL: https://cyberleninka.ru/article/n/rynochnye-tendentsii-internet-reklamy-v-rossii-2024-2025-gg (дата обращения: 02.04.2026).

References

1. Remizova, A. A. Behavioral analytics: analysis of the current state and its application to solving business problems / A. A. Remizova, T. R. Samigulin. – Text: electronic // CyberLeninka. – URL: https://cyberleninka.ru/article/n/povedencheskaya-analitika-analiz-sovremennogo-sostoyaniya-i-ee-primenenie-dlya-resheniya-zadach-biznesa (date of access: 02.04.2026).
2. Marchuk, V. V. Optimization of business processes using predictive analytics, modern economic solutions and innovations / V. V. Marchuk, N. A. Shatokhin, E. V. Senotrusova, N. S. Lapshin, V. O. Chernenko. – Text: electronic // CyberLeninka. – URL: https://cyberleninka.ru/article/n/optimizatsiya-biznes-protsessov-s-ispolzovaniem-

prognoznoy-analitiki-sovremennyh-ekonomicheskih-resheniy-i-innovatsiy (date of access: 02.04.2026).
3. 7 Advanced Machine Learning Models for Customer Insights. – 2025. – URL: https://madgicx.com/blog/advanced-machine-learning-models-for-customer-insights (date of access: 02.04.2026).
4. Top 5 Machine Learning Models for LTV Prediction. – 2025. – URL: https://www.artisangrowthstrategies.com/blog/top-5-machine-learning-models-for-ltv-prediction (date of access: 02.04.2026).
5. The Impact of Predictive Analytics on Company Performance. – Text: electronic // CyberLeninka. – URL: https://cyberleninka.ru/article/n/vliyanie-prediktivnoy-analitiki-na-deyatelnost-kompaniy (date of access: 02.04.2026).
6. Federal Law of 27.07.2006 No. 152-FZ "On Personal Data" (latest revision). – Text: electronic // ConsultantPlus. – URL: https://www.consultant.ru/document/cons_doc_LAW_61801/ (access date: 04/02/2026).
7. Mikhailov, N. D. Market trends in online advertising in Russia (2024–2025) / N. D. Mikhailov. – Text: electronic // CyberLeninka. – 2025. – URL: https://cyberleninka.ru/article/n/rynochnye-tendentsii-internet-reklamy-v-rossii-2024-2025-gg (access date: 04/02/2026).

Ефремов А.С., Потапов С.А. Основные механизмы оптимизации языка JAVASCRIPT и их влияние на производительность// Международный журнал информационных технологий и энергоэффективности.– 2026. –Т. 11 № 6(68) Ч. 1 с. 70– 78

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.4'41

ОСНОВНЫЕ МЕХАНИЗМЫ ОПТИМИЗАЦИИ ЯЗЫКА JAVASCRIPT И ИХ ВЛИЯНИЕ НА ПРОИЗВОДИТЕЛЬНОСТЬ

[1]Ефремов А.С., [2]Потапов С.А.
ФГАОУ ВО "САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ", Санкт-Петербург, Россия (190000, город Санкт-Петербург, Большая Морская ул., д.67 лит. а), e-mail: [1]andrey200ok@yandex.ru [2]potapo.potapov2014@yandex.ru

Данная работа посвящена анализу механизмов оптимизации JavaScript-кода, применяемых в системе выполнения V8. Рассматривается многоуровневая архитектура движка, включающая интерпретатор Ignition, базовый JIT-компилятор Sparkplug и оптимизирующие компиляторы Maglev и TurboFan. Особое внимание уделяется JIT-компиляции, On-Stack Replacement, сбору статистики выполнения, inline caching, hidden classes, elements kinds и внутреннему представлению малых целых чисел SMI. Показано, что эффективность оптимизации зависит от стабильности типов, форм объектов и структуры массивов. В практической части выполнено сравнение производительности JavaScript-кода при разных максимальных уровнях оптимизации V8 с использованием JetStream 3.

Ключевые слова: Javascript, v8, jit-компиляция, ignition, sparkplug, maglev, turbofan, osr, hidden classes, inline caching, динамическая оптимизация.

COMMON MECHANISMS OF JAVASCRIPT OPTIMIZATION AND THEIR IMPACT ON PERFORMANCE

[1]Efremov A.S., [2]Potapov S.A.
ST. PETERSBURG STATE UNIVERSITY OF AEROSPACE INSTRUMENTATION, St. Petersburg, Russia (190000, St. Petersburg, Bolshaya Morskaya str., 67 lit. a), e-mail: [1]andrey200ok@yandex.ru [2]potapo.potapov2014@yandex.ru

This paper analyzes JavaScript code optimization mechanisms used in the V8 execution engine. The study examines V8's multi-tier architecture, including the Ignition interpreter, the Sparkplug baseline JIT compiler, and the optimizing compilers Maglev and TurboFan. Special attention is paid to JIT compilation, On-Stack Replacement, runtime feedback collection, inline caching, hidden classes, elements kinds, and the internal representation of small integers known as SMI. The paper shows that optimization efficiency depends on type stability, object shapes, and array structure. The experimental part compares JavaScript performance at different maximum V8 optimization levels using the JetStream 3 benchmark suite.

Keywords: Javascript, v8, jit compilation, ignition, sparkplug, maglev, turbofan, osr, hidden clas-ses, inline caching, dynamic optimization.

Введение

JavaScript изначально создавался как язык сценариев для веб-страниц, однако со временем стал использоваться в сложных клиентских интерфейсах, серверных приложениях и кроссплатформенных системах. Это повысило требования к производительности программ. Простая интерпретация кода сопровождается накладными расходами на обработку инструкций, диспетчеризацию операций и работу с динамическими типами. Кроме того,

JavaScript допускает изменение структуры объектов во время выполнения и использует прототипную модель наследования, что затрудняет предварительную статическую оптимизацию [1].

Одна и та же операция в JavaScript может иметь разные варианты выполнения. Например, оператор сложения может использоваться как для арифметики, так и для конкатенации строк, а доступ к свойству объекта зависит от его формы, порядка добавления свойств и цепочки прототипов. Поэтому система выполнения не может заранее во всех случаях создать специализированный машинный код.

Одним из основных способов решения этой проблемы является динамическая оптимизация. Сначала движок выполняет код универсальным способом, затем собирает информацию о реальных типах значений, вызовах функций и структурах объектов, после чего использует эти сведения для генерации более эффективного машинного кода [1].

В данной работе рассматривается V8 — открытая система выполнения JavaScript и WebAssembly, разрабатываемая Google. V8 используется в Chrome, Node.js и может встраиваться в C++-приложения [2]. В Node.js он отвечает за выполнение JavaScript-кода вне браузера, что делает его значимым как для клиентской, так и для серверной разработки [3].

Постановка задачи

Проблема исследования заключается в том, что динамическая природа JavaScript не позволяет заранее надёжно определить типы данных, формы объектов и характер операций. Следовательно, для повышения производительности требуется использовать механизмы, способные адаптировать выполнение программы к её фактическому поведению.

Цель работы — проанализировать основные механизмы оптимизации JavaScript в движке V8 и оценить их влияние на производительность программ.

Для достижения цели необходимо решить следующие задачи: рассмотреть причины проблем производительности JavaScript; описать архитектурные уровни V8; проанализировать JIT-компиляцию и On-Stack Replacement; рассмотреть feedback vector, inline caching, SMI, elements kinds и hidden classes; оценить влияние уровней оптимизации V8 на производительность с использованием набора бенчмарков JetStream 3.

Методика исследования

Исследование включает теоретический и экспериментальный этапы. На теоретическом этапе анализируются архитектура V8 и механизмы оптимизации JavaScript-кода: Ignition, Sparkplug, Maglev, TurboFan, JIT-компиляция, OSR, feedback vector, inline caching, SMI, elements kinds и hidden classes [1, 2, 4–13].

На экспериментальном этапе сравнивается производительность JavaScript-кода при разных максимальных уровнях оптимизации V8. Для оценки используется набор бенчмарков JetStream 3, предназначенный для измерения производительности JavaScript и WebAssembly [14]. Основной метрикой является Score: чем выше его значение, тем выше производительность.

Методика эксперимента включает запуск одного и того же набора тестов при разных уровнях оптимизации, фиксацию общего рейтинга Score, сравнение результатов между

Ignition, Sparkplug, Maglev и TurboFan, а также расчёт прироста производительности между соседними уровнями и относительно базового режима Ignition.

Архитектура оптимизации V8

V8 использует многоуровневую систему выполнения, задача которой состоит в балансе между быстрым запуском программы, экономией памяти и высокой скоростью выполнения горячего кода. Вместо немедленной компиляции всего кода в максимально оптимизированный машинный код движок постепенно переводит функции между уровнями в зависимости от частоты их использования и стабильности поведения [4].

На первом уровне исходный JavaScript-код преобразуется в байткод и выполняется интерпретатором Ignition. Это обеспечивает быстрый запуск программы и позволяет не тратить ресурсы на компиляцию редко выполняемых функций [4].

Следующий уровень — Sparkplug, базовый JIT-компилятор. Он быстро преобразует байткод в машинный код без сложных оптимизаций. Такой подход снижает накладные расходы интерпретации, но сохраняет низкую стоимость компиляции [5].

Для более часто выполняемого кода применяется Maglev. Он использует данные профилирования и быстро создаёт достаточно эффективный машинный код для горячих участков программы [6].

Наиболее мощным уровнем является TurboFan. Он применяется к часто выполняемому и достаточно стабильному коду, используя данные профилирования для более сложных оптимизаций: специализации операций, встраивания функций, устранения избыточных вычислений и оптимизации циклов [4].

Таким образом, V8 реализует постепенную оптимизацию: Ignition обеспечивает запуск и сбор первичной информации, Sparkplug снижает накладные расходы интерпретации, Maglev быстро оптимизирует горячий код, а TurboFan обеспечивает максимальную производительность на наиболее стабильных участках программы.

JIT-компиляция и On-Stack Replacement

JIT-компиляция выполняется во время работы программы и позволяет учитывать её фактическое поведение: типы значений, формы объектов, целевые функции вызовов и характер операций [1]. На основе этой информации V8 создаёт специализированный машинный код.

Оптимизация в V8 является спекулятивной. Компилятор формирует предположения на основе ранее собранной статистики. Например, если операция сложения обычно выполняется над малыми целыми числами, для неё может быть создан быстрый специализированный путь. Если обращение к свойству происходит у объектов одной формы, общий поиск свойства может быть заменён доступом по известному смещению [7].

Если предположения перестают выполняться, V8 выполняет деоптимизацию: оптимизированный код заменяется более универсальным вариантом выполнения. Это позволяет сохранить корректность программы при изменении типов данных или структур объектов [7].

Для перехода между версиями кода используется On-Stack Replacement. Этот механизм позволяет заменить текущую версию исполняемого кода прямо во время работы функции. Он

особенно важен для длинных циклов, поскольку даёт возможность применить оптимизацию без ожидания повторного вызова функции.

Базовые механизмы оптимизации

Оптимизация V8 основана на совокупности механизмов, которые делают динамическое поведение JavaScript более предсказуемым для движка. К таким механизмам относятся SMI, feedback vector, elements kinds, hidden classes и inline caching.

Представление малых целых чисел (SMI)

Согласно спецификации ECMAScript, тип Number представляет числовые значения в формате double-precision 64-bit binary format IEEE 754 [8]. Однако внутри V8 для малых целых чисел используется специальное представление SMI. Оно позволяет хранить небольшие целые значения непосредственно в машинном слове без выделения отдельного объекта в куче. Это снижает накладные расходы и ускоряет арифметические операции [9].

Если значение выходит за пределы диапазона SMI или требует представления с плавающей точкой, V8 переходит к более общему способу хранения. Поэтому стабильное использование малых целых чисел в горячем коде повышает вероятность применения быстрых путей выполнения.

Feedback vector

Feedback vector используется для накопления информации о поведении функции во время выполнения. В нём сохраняются сведения об обращениях к свойствам, вызовах функций, арифметических операциях, сравнениях и доступе к элементам массивов [7].

Например, для операции сложения может быть зафиксировано, что она чаще выполняется над целыми числами, а для обращения к свойству — что объект имеет одну и ту же форму. Эти данные используются inline caching и оптимизирующими компиляторами. Если поведение функции стабильно, V8 может создать специализированный машинный код. Если поведение меняется, feedback обновляется, а оптимизированный код может быть деоптимизирован.

Elements kinds

Elements kinds — механизм классификации массивов в V8. Он нужен, поскольку массивы JavaScript могут содержать значения разных типов и изменять структуру во время выполнения. V8 различает массивы малых целых чисел, массивы чисел с плавающей точкой, массивы обычных элементов, а также плотные массивы и массивы с пропусками [10].

Плотные однотипные массивы обрабатываются быстрее, так как требуют меньше проверок. Переходы между elements kinds обычно направлены от более специализированных представлений к более общим. Например, добавление вещественного числа в массив SMI переводит его к double-представлению, а добавление строки или объекта — к ещё более универсальному виду [10]. Поэтому смешивание типов и появление пропусков снижает эффективность оптимизации.

Hidden classes

Hidden classes используются для ускорения доступа к свойствам объектов. JavaScript позволяет изменять структуру объектов во время выполнения, поэтому V8 внутренне описывает форму объекта с помощью структур Map. Hidden class хранит набор свойств, порядок их добавления и расположение значений в памяти [11].

Если несколько объектов создаются одинаково и получают свойства в одном порядке, они могут иметь одну hidden class. В этом случае доступ к свойству выполняется не как полный поиск по имени, а как обращение по известному смещению [11].

Для описания свойств используются DescriptorArray, а переходы между формами объектов фиксируются с помощью TransitionArray [12]. Важным фактором является порядок добавления свойств: объекты с одинаковым итоговым набором свойств могут иметь разные hidden classes, если свойства добавлялись в разной последовательности. Поэтому стабильное создание объектов одного типа улучшает условия для оптимизации.

Inline caching

Inline caching ускоряет доступ к свойствам и вызовы методов за счёт повторного использования информации, полученной при предыдущих выполнениях той же операции. При первом обращении к свойству V8 использует общий механизм поиска, затем сохраняет форму объекта и способ доступа к значению. При последующих обращениях движок проверяет форму объекта и, если она совпадает, использует быстрый путь [13].

Inline cache может быть мономорфным, полиморфным или мегаморфным. Мономорфный вариант наиболее эффективен, поскольку одно место программы работает с объектами одной формы. При большем числе форм эффективность специализации снижается [13].

Inline caching связан с feedback vector и hidden classes. Hidden classes описывают формы объектов, feedback vector сохраняет сведения о выполнении, а inline cache использует эти данные для ускорения конкретных операций доступа [7, 13].

Тестирование производительности при разных уровнях оптимизации

В целях эмпирической оценки влияния механизмов оптимизации на производительность была проведена серия тестов на бенчмарке JetStream 3. В качестве метрики для каждого отдельного теста использовался Score – безразмерная метрика, высчитываемся на основе среднего геометрического значения времени выполнения программы [15].

Большее значение Score соответствует лучшей производительности. Результаты данного тестирования представлены в Таблице 1.

Таблица 1 - Результаты тестирования (значение Score) с разными уровнями оптимизации на бенчмарке JetStream 3

Название теста	Ignition	Sparkplug	Maglev	Turbofan
WSL	2.01	2.66	5.45	5.61
web-ssr	151.72	185.2	236.97	234.06
validatorjs	344.09	391.16	461.6	459.65
UniPoker	105.09	115.83	692.34	815.47
typescript-lib	1.23	2.37	6.69	6.91
threejs	15.78	19.66	79.59	77.33
sync-fs	77.62	109.35	437.47	586.32

Sunspider	146.46	162.97	762.78	945.81
stanford-crypto-sha256	48.88	52.79	366.18	1030.3
stanford-crypto-pbkdf2	52.26	56.33	282.14	983.13
stanford-crypto-aes	41.44	47.82	295.91	570.62
splay	97.88	129.1	346.7	365.03
source-map-wtb	107.04	132.12	424.14	393.11
richards	63.69	75.37	1230.43	1381.64
Пregexp-octane	331.56	401.35	753.95	760.79
raytrace-public-class-fields	39.56	50.74	389.18	790.8
raytrace-private-class-fields	28.78	36.9	312.76	665.88
raytrace	50.38	59.53	542.78	755.95
proxy-vue	431.38	568.21	857.72	883.83
proxy-mobx	815.29	1043.29	1761.02	1740.01
prismjs-startup-es6	383.6	420.75	422.05	395.61
prettier-wtb	23.44	36.06	99.11	101.1
postcss-wtb	16.71	21.53	43.83	47.25
pdfjs	74.68	94.73	311.88	327.03
OfflineAssembler	99.87	126.54	298.88	284.85
octane-code-load	1345.12	1285.94	1342.87	1296.56
navier-stokes	100.24	100.53	505.92	594.61
multi-inspector-code-load	448.8	439.77	509.97	491.9
mobx-startup	65.42	131.06	233.74	232.97
ML	9.95	13.38	100.08	168.97
mandreel	5.47	6.21	94.73	157.56
lazy-collections	80.69	154.76	315.01	336.22
json-stringify-inspector	412.81	473.79	515.94	510.9
json-parse-inspector	286.73	337.6	331.9	334.44
jsdom-d3-startup	9.55	13.36	20.52	20.32
js-tokens	126.27	188.72	383.61	381.14
hash-map	44.25	55.66	623.84	896.38
gbemu	24.67	21.07	173.63	204.18
gaussian-blur	21.14	23.3	204.4	470.02
FlightPlanner	465.52	578.19	968.03	1025.15
first-inspector-code-load	287.8	284.92	269.8	272.23
esprima-next-wtb	13.7	17.29	117.81	129.82
espree-wtb	11.32	15.63	86.12	105.49

earley-boyer	163.82	208.09	1091.44	1201.03
doxbee-promise	397.46	488.61	814.36	772.23
doxbee-async	418.09	663.65	1110.83	1496.76
delta-blue	96.77	115.95	1670.57	2462.9
crypto	89.83	104.21	802.51	1773.85
chai-wtb	135.89	173.23	266.04	288.1
cdjs	41.37	49.19	307.57	341.59
Box2D	72.72	80.64	597.11	649.37
bigint-noble-ed25519	62.08	80.44	139.23	150.8
Basic	268.48	341.65	1212.89	1324.8
babylonjs-startup-es6	28.37	26.63	27.51	28.62
babylonjs-scene-es6	37.79	45.58	74.53	78.57
babylon-wtb	12.8	17.63	100.54	112.25
Babylon	204.5	259.34	1093.51	1259.21
babel-wtb	58.97	80.03	158.95	162.39
babel-minify-wtb	35.15	49.47	90.47	128.41
async-fs	130.61	190.6	388.17	506.76
Air	303.29	337.08	726.13	768.94
ai-astar	141.96	177.31	884.75	1026.06
acorn-wtb	8.31	11.04	69.72	76.73
Общий рейтинг	71.52	89.12	286.39	344.85

Таблица 2. - Прирост к производительности от каждого уровня оптимизации

Уровень оптимизации	Рейтинг	Прирост к предыдущему уровню (%)	Прирост относительно производительности без оптимизации (%)
Ignition	71.52		
Sparkplug	89.12	24.61%	24.61%
Maglev	286.39	221.35%	300.43%
TurboFan	344.85	20.41%	382.17%

Переход от Ignition к Sparkplug увеличил производительность на 24.61%. Наиболее значимый прирост был получен при переходе от Sparkplug к Maglev: рейтинг вырос с 89.12 до 286.39, что соответствует увеличению на 221.35%. Переход от Maglev к TurboFan дал дополнительный прирост 20.41%. Итоговое ускорение TurboFan относительно Ignition составило 382.17%.

Результаты показывают, что наибольший вклад в повышение производительности даёт оптимизирующая JIT-компиляция, использующая статистику выполнения. При этом максимальный уровень оптимизации не всегда является лучшим для каждого отдельного теста, поскольку результат зависит от структуры кода, стабильности типов и стоимости самой оптимизации.

Заключение

В ходе работы были рассмотрены основные механизмы оптимизации JavaScript, применяемые в движке V8. Показано, что производительность обеспечивается совокупностью решений: многоуровневой JIT-компиляцией, сбором статистики выполнения, inline caching, hidden classes, elements kinds, SMI, спекулятивной оптимизацией и деоптимизацией.

V8 сначала выполняет код универсальным способом, а затем постепенно создаёт специализированные версии машинного кода для часто исполняемых участков. Такой подход позволяет учитывать динамическую природу JavaScript и снижать накладные расходы, связанные с изменяемыми типами и структурами объектов.

Практическое тестирование подтвердило влияние уровней оптимизации на производительность. Общий рейтинг JetStream 3 вырос с 71.52 в режиме Ignition до 344.85 в режиме TurboFan, что соответствует приросту 382.17%. Наибольший вклад внёс Maglev, обеспечивший резкий рост производительности после базовой компиляции Sparkplug. Это подтверждает, что профилирующая JIT-компиляция является одним из ключевых механизмов ускорения современных JavaScript-приложений.

Список литературы

1. Жуйков Р., Мельник Д., Бучацкий Р., Варданян В., Иванишин В., Шарыгин Е. Методы динамической и предварительной оптимизации программ на языке JavaScript // Труды ИСП РАН. – Т. 26. – №1. – 2014. – С. 297-314. https://doi.org/10.15514/ISPRAS-2014-26(1)-10
2. Официальный сайт V8. What is V8? [Электронный ресурс]. – Режим доступа: https://v8.dev (дата обращения: 25.04.2026)
3. Node.js Documentation. The V8 JavaScript Engine [Электронный ресурс]. – Режим доступа: https://nodejs.org/en/learn/getting-started/the-v8-javascript-engine (дата обращения: 25.04.2026)
4. Официальный сайт V8. Launching Ignition and TurboFan [Электронный ресурс]. – Режим доступа: https://v8.dev/blog/launching-ignition-and-turbofan (дата обращения: 26.04.2026)
5. Официальный сайт V8. Sparkplug – a non-optimizing JavaScript compiler [Электронный ресурс]. – Режим доступа: https://v8.dev/blog/sparkplug (дата обращения: 26.04.2026)
6. Официальный сайт V8. Maglev – V8s Fastest Optimizing JIT [Электронный ресурс]. – Режим доступа: https://v8.dev/blog/maglev (дата обращения: 26.04.2026)
7. Варданян В. Г. Методы оптимизации программ на языке JavaScript, основанные на статистике выполнения программы // Труды ИСП РАН. – 2016. – Т. 28. – №1. – С. 520. https://doi.org/10.15514/ISPRAS-2016-28(1)-1
8. ECMA International. ECMAScript 2023 Language Specification. ECMA-262 [Электронный ресурс]. – Режим доступа: https://262.ecma-international.org/14.0/ (дата обращения: 27.04.2026)
9. Официальный сайт V8. Pointer Compression in V8 [Электронный ресурс]. – Режим доступа: https://v8.dev/blog/pointer-compression (дата обращения: 27.04.2026)
10. Официальный сайт V8. Elements kinds in V8 [Электронный ресурс]. – Режим доступа: https://v8.dev/blog/elements-kinds (дата обращения: 27.04.2026)
11. Официальный сайт V8. Maps (Hidden Classes) in V8 [Электронный ресурс]. – Режим доступа: https://v8.dev/docs/hidden-classes (дата обращения: 28.04.2026)

12. Официальный сайт V8. Fast properties in V8 [Электронный ресурс]. – Режим доступа: https://v8.dev/blog/fast-properties (дата обращения: 28.04.2026)
13. Bynens M. JavaScript engine fundamentals: Shapes and Inline Caches [Электронный ресурс]. – Режим доступа: https://mathiasbynens.be/notes/shapes-ics (дата обращения: 28.04.2026)
14. WebKit. Introducing the JetStream 3 Benchmark Suite [Электронный ресурс]. – Режим доступа: https://webkit.org/blog/17899/introducing-the-jetstream-3-benchmark-suite (дата обращения: 29.04.2026).

References

1. R. Zhuikov, D. Melnik, R. Buchatsky, V. Vardanyan, V. Ivanishin, E. Sharygin. Methods of dynamic and preliminary optimization of JavaScript programs. Proceedings of ISP RAS. - Vol. 26. - No. 1. - 2014. - pp. 297-314. https://doi.org/10.15514/ISPRAS-2014-26(1)-10
2. Official V8 website. What is V8? [Electronic resource]. - Access mode: https://v8.dev (date of access: 04/25/2026)
3. Node.js Documentation. The V8 JavaScript Engine [Electronic resource]. – Access mode: https://nodejs.org/en/learn/getting-started/the-v8-javascript-engine (date accessed: 25.04.2026)
4. Official V8 website. Launching Ignition and TurboFan [Electronic resource]. – Access mode: https://v8.dev/blog/launching-ignition-and-turbofan (date accessed: 26.04.2026)
5. Official V8 website. Sparkplug – a non-optimizing JavaScript compiler [Electronic resource]. – Access mode: https://v8.dev/blog/sparkplug (date accessed: 26.04.2026)
6. Official V8 website. Maglev – V8s Fastest Optimizing JIT [Electronic resource]. – Access mode: https://v8.dev/blog/maglev (date accessed: 26.04.2026)
7. Vardanyan V. G. Methods for optimizing JavaScript programs based on program execution statistics // Proceedings of ISP RAS. – 2016. – Vol. 28. – No. 1. – p. 520. https://doi.org/10.15514/ISPRAS-2016-28(1)-1
8. ECMA International. ECMAScript 2023 Language Specification. ECMA-262 [Electronic resource]. – Access mode: https://262.ecma-international.org/14.0/ (date accessed: 27.04.2026)
9. Official website of V8. Pointer Compression in V8 [Electronic resource]. – Access mode: https://v8.dev/blog/pointer-compression (accessed date: 27.04.2026)
10. Official V8 website. Elements kinds in V8 [Electronic resource]. – Access mode: https://v8.dev/blog/elements-kinds (accessed date: 27.04.2026)
11. Official V8 website. Maps (Hidden Classes) in V8 [Electronic resource]. – Access mode: https://v8.dev/docs/hidden-classes (accessed date: 28.04.2026)
12. Official V8 website. Fast properties in V8 [Electronic resource]. – Access mode: https://v8.dev/blog/fast-properties (accessed date: 28.04.2026)
13. Bynens M. JavaScript engine fundamentals: Shapes and Inline Caches [Electronic resource]. – Available at: https://mathiasbynens.be/notes/shapes-ics (Accessed: 28.04.2026)
14. WebKit. Introducing the JetStream 3 Benchmark Suite [Electronic resource]. – Available at: https://webkit.org/blog/17899/introducing-the-jetstream-3-benchmark-suite (Accessed: 29.04.2026).

Селезнев Д.А., Иващенков Н.А. Перспективы развития эндоваскулярных нейроинтерфейсов// Международный журнал информационных технологий и энергоэффективности.– 2026. –Т. 11 № 6(68) Ч. 1 с. 79–86

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.94:004.8

ПЕРСПЕКТИВЫ РАЗВИТИЯ ЭНДОВАСКУЛЯРНЫХ НЕЙРОИНТЕРФЕЙСОВ

[1]Селезнев Д.А., **[2]Иващенков Н.А.**
ФГАОУ ВО "САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ "ЛЭТИ" ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)", Санкт-Петербург, Россия (197022, город Санкт-Петербург, ул Профессора Попова, д. 5 литера Ф), e-mail: seleznev_d_a_2001@mail.ru
[2]ФГБОУ ВО "ТАМБОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Г.Р. ДЕРЖАВИНА", Тамбов, Россия (392000, Тамбовская область, город Тамбов, Интернациональная ул., д. 33)

Эндоваскулярные нейроинтерфейсы (EV-BCI) - перспективный класс БКИ, позволяющий записывать и передавать нейронные сигналы через венозную систему мозга без краниотомии. Технология основана на установке стент-мониторов с электродами (например, «Stentrode») в поверхностные мозговые сосуды (обычно верхний сагиттальный синус). Предварительные исследования на животных и первые клинические испытания в парализованных пациентах демонстрируют устойчивую запись и достаточное качество сигналов (сравнимое с субдуральной ЭЭГ) при отсутствии серьёзных осложнений[1][2]. Основные вызовы — риск тромбозов, долговременная стабильность электродов и анатомические вариации сосудов[3]. Современные разработки (например, микропроволочные электродные элементы ~300 мкм с лазерной сваркой[4][5]) направлены на миниатюризацию и повышение биосовместимости устройств. Этические аспекты включают обеспечение безопасности пациента и конфиденциальности мыслей. В целом, EV-BCI открывают новый путь к масштабному применению имплантируемых НИИ для пациентов с нейродегенерацией и травмами спинного мозга, однако требуют дальнейших исследований по оптимизации материалов, алгоритмов обработки сигналов и долгосрочной надежности[6][7].

Ключевые слова: Эндоваскулярный нейроинтерфейс, Stentrode, кровеносные сосуды мозга, биосовместимость, неврологический интерфейс, BCI.

PROSPECTS FOR THE DEVELOPMENT OF ENDOVASCULAR NEUROINTERFACES

[1]Seleznev D.A., **[2]Ivaschenkov N.A**.
[1]ST. PETERSBURG STATE ELECTROTECHNICAL UNIVERSITY "LETI". V.I. ULYANOVA (LENINA), St. Petersburg, Russia (197022, St. Petersburg, Professora Popova str., 5 letter F), e-mail: seleznev_d_a_2001@mail.ru
[2]DERZHAVIN TAMBOV STATE UNIVERSITY, Tambov, Russia (392000, Tambov, Internatsionalnaya str., 33)

Endovascular brain–computer interfaces (BCIs) are an emerging technology enabling neural signal recording and transmission via the cerebral venous system, avoiding craniotomy. A self-expanding stent-electrode array (e.g. “Stentrode”) is delivered through the jugular vein into the superior sagittal sinus adjacent to sensorimotor cortex. Preclinical and early human studies have demonstrated stable long-term recording of motor-related signals, with signal quality comparable to subdural electrocorticography and minimal serious adverse events[1][2]. Key challenges include thrombosis risk, device stability, and anatomical variability[3]. Innovations such as 300 µm laser-welded micro-wire electrodes improve miniaturization[4][5]. Ethical considerations involve patient safety and privacy of neural data. Endovascular BCI holds promise for assisting paralyzed patients, but further work is needed to optimize materials, signal processing, and long-term biocompatibility[6][7].

Keywords: Endovascular neural interface, Stentrode, brain blood vessels, biocompatibility, neurological interface, BCI.

Введение

БКИ (brain–computer interfaces) открывают путь к управлению устройствами силой мысли, однако традиционные инвазивные методики требуют краниотомии и вживления электродов непосредственно в мозг[1]. *Эндоваскулярный БКИ* (EV-BCI) реализует нейронную запись через стенку кровеносного сосуда, что делает процедуру подобной установке стента и снижает травматичность. В частности, устройство «Stentrode» представляет собой тонкоплёночную сетку с электродами (обычно 16 каналов), вводимую через шейную вену и расправляемую в верхнем сагиттальном синусе перед мотонейро-корой[8][9]. На рис.1 показана схема полностью имплантируемой системы EV-BCI, где регистрирующее устройство устанавливается в сосуд и соединено гибким проводом с подшитым под кожу передатчиком[2].

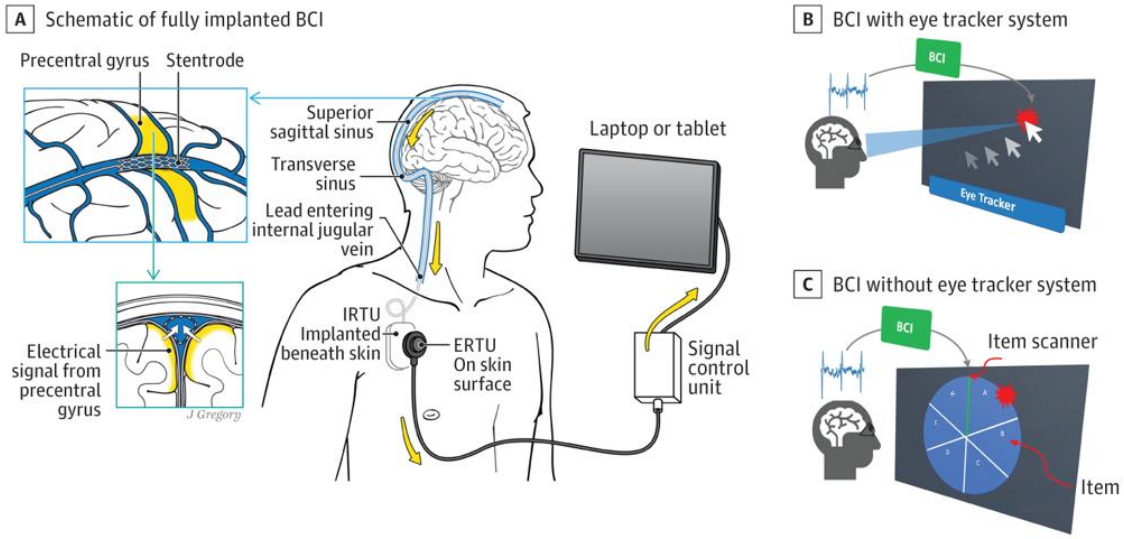

Рисунок 1 - Схематическое устройство эндоваскулярного БКИ: электроды на раскрывающемся стенте устанавливаются в полость кровеносного сосуда мозга, нейронный сигнал через катетер передаётся на подкожный блок, а далее — на внешнее устройство управления. (Иллюстрация по данным Mitchell et al. 2023[2].)

Исследования показывают, что эндоваскулярные БКИ позволяют фиксировать моторные потенциалы и другие корковые ритмы при снижении рисков, связанных с открытой операцией. Так, Ognard et al. (2025) провели систематический обзор 26 исследований EV-BCI и отметили, что «эндоваскулярные БКИ обеспечивают регистрацию через церебральные вены, избегая краниотомии»[1]. При этом в опытах на овцах и первых клинических испытаниях в качестве ALS-пациентов устройство Stentrode продемонстрировало устойчивую запись сигналов в течение месяцев без серьёзных осложнений[10][11]. Качество записи при этом оказалось соизмеримо с субдуральной ЭКоГ[11].

История и первые исследования

Идея эндоваскулярной регистрации нейросигналов восходит к 1970-м годам. Впервые Penn et al. (1973) продемонстрировали запись ЭЭГ кролика с помощью эндоваскулярного электрода, введенного в сосуд[12]. В последующие годы Nakase и др. применили внутриартериальные электроды для мониторинга корковых сигналов у человека[12]. Развитие

методики сдерживала примитивность электродов и риск тромбоза. Ситуация изменилась с появлением самораскрывающихся стентов как основы для электродной решетки. В 2020–2021 гг. группа Окслея (Synchron) сообщила о первых успешных имплантациях Stentrode у людей с БАС (публикация в Lancet Neurol 2021), продемонстрировав работоспособность системы для управления компьютерным интерфейсом.

Технологии и конструирование

Ключевым элементом EV-BCI является *стент-электрод* - тонкий металлический каркас (обычно из нитинола) с прикреплёнными на нём микроэлектродами. При распространении стент фиксируется в просвете сосуда, прижимая электроды к стенке. Аппаратная часть включает пульпостью управления (блок импульсного питания) и беспроводной передатчик. Система Synchron (Stentrode) предусматривает имплантацию блока с радиопередатчиком в подкожную область над грудью, соединенного проводом со стентом. После включения сигнал с электродов по оптике или радиоканалу поступает к внешнему устройству.

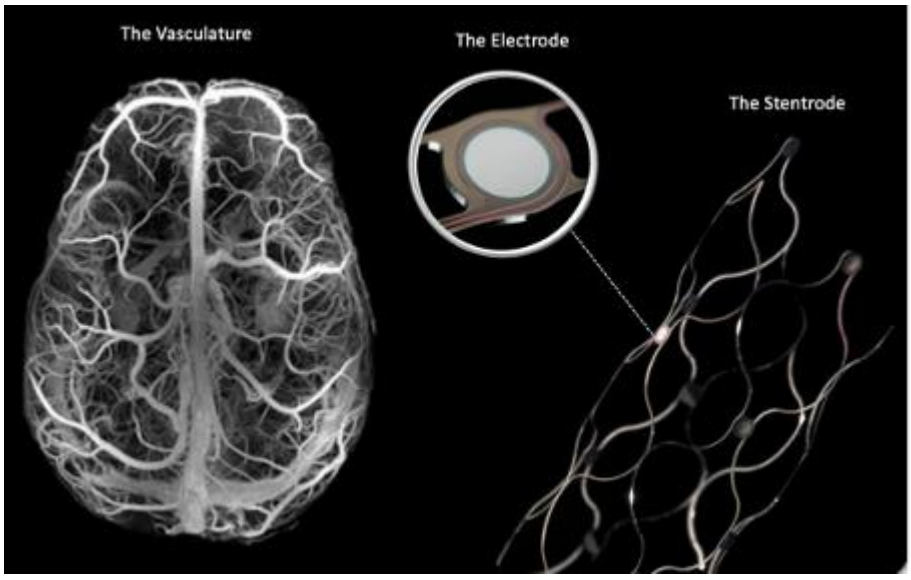

Рисунок 2 - Фотография стент-монитора «Stentrode» с прилегающими электродами (модель от компании Synchron)[8]. Устройство имеет тонкостенный самораскрывающийся каркас с микролитиевыми электродами для эндоваскулярной регистрации биоэлектрических сигналов.

Для повышения компактности разрабатываются новые конструкции. Wen et al. (2023) применили **лазерную сварку** для изготовления микропроводов диаметром ~300 мкм с дисковыми электродами, что уменьшило общий диаметр узла до возможности введения через катетер 0.6–0.8 мм[4][5]. Это обеспечивает низкое импедансное сопротивление и стабильное ионный ток на электродах. Другой подход — гибкие полимерные электродные нити (EP-01) для внутрисосудистой ЭЭГ. Masuda et al. (2025) разработали тонкий (платиновый провод в полимерной оболочке) электрод, способный входить через стандартный микрокатетер. В первом же клиническом опыте EP-01 успешно регистрировал внутрисосудистую ЭЭГ в нескольких синусах мозга у пациента с эпилепсией; амплитуда эндоваскулярной ЭЭГ в 3–4 раза превышала таковую скальповой ЭЭГ[13][14].

Клинические испытания и результаты

Первоначальные клинические испытания EV-BCI проходили в Австралии и США. Ключевым этапом стало исследование SWITCH (Stentrode With Thought-Controlled switch) под руководством Mitchell et al. (2023). В этой серии из 5 пациентов (4 с БАС, 1 с первичной боковой склерозой) стент-электрод был установлен в верхнем сагиттальном синусе и подключен к БКИ-системе. В ходе годового наблюдения *не было зафиксировано ни одного серьёзного осложнения, связанного с устройство*м (тромбозы, кровотечения или неврологические дефициты)[2]. Пациенты смогли эффективно управлять курсором и выполнять задачи набора текста силой мысли: средняя точность «кликов» составляла ~92–93 %, скорость набора до ~13–20 символов в минуту[9][2]. Это свидетельствует о высокой функциональной пригодности системы: ключевые параметры (амплитуда, спектральные характеристики β-ритма во время моторной активности) были стабильны в течение всего года[2], а качество сигнала сравнимо с субдуральными матрицами[15].

Подтверждая безопасность, исследование SWITCH показало, что запись через кровеносный сосуд возможна и эффективна: «Обнаружено, что возможно регистрировать нейронные сигналы из кровеносного сосуда мозга, и благоприятный профиль безопасности этого подхода может способствовать более широкому внедрению БКИ для парализованных пациентов»[7]. Эти результаты согласуются с выводами систематического обзора: EV-BCI *минимизирует инвазивность и имеет эффективность, сравнимую с традиционными методами,* но требует дальнейшего накопления клинических данных[3][6].

Перспективы и новые направления

Перспективы EV-нейроинтерфейсов связаны с расширением клинических приложений и развитием технологии. В качестве регистрационного БКИ они могут быть применены не только при боковом амиотрофическом склерозе, но и у пациентов со спинальной травмой, инсультом или другими двигательными нарушениями. В других исследованиях предлагается использовать эндоваскулярные электроды для *глубокой стимуляции мозга* при неврологических заболеваниях. Например, Opie et al. (2018) показали в овцах, что стимуляция моторной коры через стент-электрод вызывает управляемые мышечные ответы, сопоставимые с прямыми электродами[16]. Это открывает путь к малоинвазивному вмешательству при эпилепсии, болезни Паркинсона и депрессии, хотя прямых клинических испытаний ещё не проводилось.

Технологически важно уменьшать размер электродов, улучшать адгезию к стенке сосуда и разрабатывать биоразлагаемые элементы. Также ведутся работы над закрытыми *имплантируемыми системами* с беспроводной передачей, индуктивным питанием и встроенной обработкой сигналов. Анализ данных с EV-BCI может использовать современные методы машинного обучения, адаптивные алгоритмы декодирования и фильтрацию артефактов.

Проблемы и этические аспекты

Несмотря на успехи, EV-нейроинтерфейсы сталкиваются с серьёзными вызовами. Основной медицинский риск - образование тромбов вокруг импланта, приводящих к окклюзии сосуда. Действительно, один из ключевых выводов обзора: «Ключевые проблемы включают риск тромбоза, долгосрочную стабильность электродов и анатомическую вариабельность»[3].

Для снижения этих рисков требуется тщательный подбор антикоагулянтной терапии, а также оптимизация поверхности электродов (например, покрытие антиплазменным полимером). Долгосрочная надежность - ещё одна проблема: физическая миграция стента или дезактивация электродов могут ухудшить работу интерфейса со временем[3].

Этические вопросы включают информированное согласие пациентов (особенно тяжёлых), конфиденциальность нейроданных и потенциальные психические эффекты (например, изменение ощущений). Требуется сбалансированная оценка «польза/риск», особенно учитывая пожизненный характер имплантации (как «пейсмейкер для мозга»). Развитие регулирующих стандартов также отстаёт от темпа технологий, что вызывает дискуссии об ответственности за возможные сбои БКИ.

Заключение

Эндоваскулярные нейроинтерфейсы представляют собой многообещающее направление нейроинженерии. Их преимущество - минимальная инвазивность при сохранении высокого качества сигналов[1][11]. Полевые испытания показали возможность восстановления коммуникации для парализованных пациентов без серьёзных осложнений[2]. Тем не менее технология находится на ранних этапах: необходимы дополнительные клинические исследования, оптимизация материалов и тщательный мониторинг безопасности[6][7]. В ближайшие годы ожидается расширение числа испытаний (например, многоцентровые и длинносрочные исследования) и внедрение новых технических решений (еще более тонкие и гибкие электроды, автоматические системы контроля). Таким образом, эндоваскулярные нейроинтерфейсы имеют потенциал сделать БКИ более доступными и безопасными, открывая путь к их широкому применению в неврологической реабилитации и нейромедицины.

Список литературы

1. Огнард Дж. и др. Достижения в области эндоваскулярного интерфейса мозг-компьютер: систематический обзор и будущие перспективы // Журнал нейробиологических методов. – 2025. – Т. 420. – С. 110471.
2. Митчелл П. и др. Оценка безопасности полностью имплантированного эндоваскулярного интерфейса мозг-компьютер для лечения тяжелого паралича у 4 пациентов: исследование стентрода с цифровым переключателем, управляемым мыслью (SWITCH) // JAMA неврология. – 2023. – Т. 80. – №. 3. – С. 270-278.
3. Дубинин И. и др. Нейронно-компьютерные интерфейсы: теория, практика, перспективы // Прикладные науки. – 2025. – Т. 15. – №. 16. – С. 8900.
4. Вэнь Б., Шен Л., Кан Х. Лазерная сварка микропроволочного стент-электрода как минимально инвазивный эндоваскулярный нейронный интерфейс //Микромашины. – 2024. – Т. 16. – №. 1. – С. 21.
5. Араки К. и др. Выявление межприступных эпилептиформных разрядов с помощью множественных двусторонних вставок недавно разработанного микрокатетер-совместимого эндоваскулярного электрода электроэнцефалограммы: клиническое исследование осуществимости //Epilepsia Open. – 2025.
6. Солдози С. и др. Систематический обзор эндоваскулярных стент-электродных массивов, минимально инвазивный подход к интерфейсам мозг-машина //Нейрохирургический фокус. – 2020. – Т. 49. – №. 1. – С. E3.

7. Оксли Т. Дж. и др. Моторный нейропротез, имплантированный с помощью нейроинтервенционной хирургии, улучшает способность к выполнению повседневных задач при тяжелом параличе: первый опыт применения у человека //Журнал нейроинтервенционной хирургии. – 2021. – Т. 13. – №. 2. – С. 102-108.
8. Мусси М. Г., Адамс К. Д. Гибридные интерфейсы мозг-компьютер на основе ЭЭГ: обзор с применением существующей таксономии гибридных интерфейсов мозг-компьютер и соображения для педиатрических применений //Frontiers in human neuroscience. – 2022. – Т. 16. – С. 1007136.
9. Лебедев М. А., Николелис М. А. Л. Интерфейсы мозг-машина: прошлое, настоящее и будущее //TRENDS in Neurosciences. – 2006. – Т. 29. – №. 9. – С. 536-546.
10. Аджибойе А. Б. и др. Восстановление движений захвата и хватания посредством стимуляции мышц, управляемой мозгом, у человека с тетраплегией: демонстрация концепции //The Lancet. – 2017. – Т. 389. – №. 10081. – С. 1821-1830.
11. Пандаринатх С. и др. Высокоэффективная коммуникация людей с параличом с использованием внутрикортикального интерфейса мозг-компьютер //elife. – 2017. – Т. 6. – С. e18554.
12. Хохберг Л. Р. и др. Достижение и захват предметов людьми с тетраплегией с помощью роботизированной руки с нейронным управлением //Nature. – 2012. – Т. 485. – №. 7398. – С. 372-375.
13. Коллинджер Дж.Л. и др. Высокоэффективное управление нейропротезом человека с тетраплегией //The Lancet. – 2013. – Т. 381. – №. 9866. – С. 557-564.
14. Хохберг Л.Р. и др. Нейронный ансамблевый контрольпротезных устройств человека с тетраплегией //Природа. – 2006. – Т. 442. – №. 7099. – С. 164-171.
15. Лебедев М., Оприс И. Интерфейсы мозг-машина: от макро- до микросхем //Последние достижения в области модульной организации коры головного мозга [Интернет]. Дордрехт: Springer Нидерланды. – 2015. – С. 407-28.
16. Орбан М. и др. Обзор мозговой активности и ЭЭГ-основанных интерфейсов мозг-компьютера для реабилитационных приложений // Биоинженерия. – 2022. – Т. 9. – № 12. – С. 768.
17. Чари А. и др. Интерфейсы мозг-машина: роль нейрохирурга // Мировая нейрохирургия. – 2021. – Т. 146. – С. 140-147.
18. Филлипс В. От нейронов к сетям: этические аспекты нейронных интерфейсов с использованием ИИ // ИИ и этика. – 2025. – Т. 5. – № 4. – С. 3531-3536.
19. Чжу Х. Ю. и др. Человекоцентричная метавселенная, обеспечиваемая интерфейсом мозг-компьютер: обзор // IEEE Communications Surveys & Tutorials. – 2024. – Т. 26. – № 3. – С. 2120-2145.
20. Цзян Л., Фунг Ч. Ч. С., Ченг Ч. П. В. Достижения в исследованиях по нейронавигированной локализации цели для повторяющейся транскраниальной магнитной стимуляции при депрессии: от стандартизации к индивидуализированной нейромодуляции //Frontiers in Neuroimaging. – 2025. – Т. 4. – С. 1703198.

References

1. Ognard J. et al. Advances in endovascular brain computer interface: Systematic review and future implications //Journal of Neuroscience Methods. – 2025. – Т. 420. – pp. 110471.

2. Mitchell P. et al. Assessment of safety of a fully implanted endovascular brain-computer interface for severe paralysis in 4 patients: the stentrode with thought-controlled digital switch (SWITCH) study //JAMA neurology. – 2023. – Т. 80. – №. 3. – pp. 270-278.
3. Dubynin I. et al. Neural–Computer Interfaces: Theory, Practice, Perspectives //Applied Sciences. – 2025. – Т. 15. – №. 16. – pp. 8900.
4. Wen B., Shen L., Kang X. Laser Welding of Micro-Wire Stent Electrode as a Minimally Invasive Endovascular Neural Interface //Micromachines. – 2024. – Т. 16. – №. 1. – pp. 21.
5. Araki K. et al. Detection of interictal epileptiform discharges using multiple bilateral insertions of a newly developed microcatheter-compatible endovascular electroencephalogram electrode: A clinical feasibility trial //Epilepsia Open. – 2025.
6. Soldozy S. et al. A systematic review of endovascular stent-electrode arrays, a minimally invasive approach to brain-machine interfaces //Neurosurgical Focus. – 2020. – Т. 49. – №. 1. – С. E3.
7. Oxley T. J. et al. Motor neuroprosthesis implanted with neurointerventional surgery improves capacity for activities of daily living tasks in severe paralysis: first in-human experience //Journal of neurointerventional surgery. – 2021. – Т. 13. – №. 2. – pp. 102-108.
8. Mussi M. G., Adams K. D. EEG hybrid brain-computer interfaces: A scoping review applying an existing hybrid-BCI taxonomy and considerations for pediatric applications //Frontiers in human neuroscience. – 2022. – Т. 16. – pp. 1007136.
9. Lebedev M. A., Nicolelis M. A. L. Brain–machine interfaces: past, present and future //TRENDS in Neurosciences. – 2006. – Т. 29. – №. 9. – pp. 536-546.
10. Ajiboye A. B. et al. Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration //The Lancet. – 2017. – Т. 389. – №. 10081. – pp. 1821-1830.
11. Pandarinath C. et al. High performance communication by people with paralysis using an intracortical brain-computer interface //elife. – 2017. – Т. 6. – С. e18554.
12. Hochberg L. R. et al. Reach and grasp by people with tetraplegia using a neurally controlled robotic arm //Nature. – 2012. – Т. 485. – №. 7398. – pp. 372-375.
13. Collinger J. L. et al. High-performance neuroprosthetic control by an individual with tetraplegia //The Lancet. – 2013. – Т. 381. – №. 9866. – pp. 557-564.
14. Hochberg L. R. et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia //Nature. – 2006. – Т. 442. – №. 7099. – pp. 164-171.
15. Lebedev M., Opris I. Brain-machine interfaces: from macro-to microcircuits //Recent Advances on the Modular Organization of the Cortex [Internet]. Dordrecht: Springer Netherlands. – 2015. – pp. 407-28.
16. Orban M. et al. A review of brain activity and EEG-based brain–computer interfaces for rehabilitation application //Bioengineering. – 2022. – Т. 9. – №. 12. – pp. 768.
17. Chari A. et al. Brain–machine interfaces: the role of the neurosurgeon //World neurosurgery. – 2021. – Т. 146. – pp. 140-147.
18. Phillips V. From neurons to networks: ethical dimensions of AI-infused neural interfaces //AI and Ethics. – 2025. – Т. 5. – №. 4. – pp. 3531-3536.
19. Zhu H. Y. et al. A human-centric metaverse enabled by brain-computer interface: A survey //IEEE Communications Surveys & Tutorials. – 2024. – Т. 26. – №. 3. – pp. 2120-2145.

20. Jiang L., Fung C. C. S., Cheng C. P. W. Research advances in neuronavigated target localization for repetitive transcranial magnetic stimulation in depression: from standardization to individualized neuromodulation //Frontiers in Neuroimaging. – 2025. – Т. 4. – pp. 1703198.

Илалова О.Р. Современные технологии искусственного интеллекта в обеспечении доступной среды для лиц с ограниченными возможностями здоровья // Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч. 1 с. 87–90

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.89

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ОБЕСПЕЧЕНИИ ДОСТУПНОЙ СРЕДЫ ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

Илалова О.Р.
ФГАОУ ВО "РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ ИМЕНИ ПАТРИСА ЛУМУМБЫ", Москва, Россия (117198, город Москва, Миклухо-Маклая ул., д. 6), e-mail: ilalova2001@gmail.com

В работе рассматриваются возможности применения технологий искусственного интеллекта для создания доступной среды и повышения качества жизни людей с ограниченными возможностями здоровья. Проводится анализ практических решений в области компьютерного зрения, обработки естественного языка и машинного обучения, которые помогают лицам с ограниченными возможностями здоровья преодолевать барьеры, связанные с нарушениями зрения, слуха, опорно-двигательного аппарата и когнитивными расстройствами. Особое внимание уделяется мобильным приложениям и интеллектуальным системам, обеспечивающим независимость и социальную интеграцию лиц с ограниченными возможностями здоровья.

Ключевые слова**:** Искусственный интеллект, доступная среда, инклюзия, компьютерное зрение, распознавание речи, машинное обучение, люди с ограниченными возможностями здоровья.

CONTEMPORARY ARTIFICIAL INTELLIGENCE TECHNOLOGIES IN PROVIDING ACCESSIBLE ENVIRONMENTS FOR INDIVIDUALS WITH DISABILITIES

Ilalova O.R.
PATRICE LUMUMBA PEOPLES' FRIENDSHIP UNIVERSITY OF RUSSIA, Moscow, Russia (117198, Moscow, Miklukho-Maklaya str., 6), e-mail: ilalova2001@gmail.com

The article examines the possibilities of using artificial intelligence technologies to create an accessible environment and improve the quality of life of people with disabilities. Practical solutions in the field of computer vision, natural language processing and machine learning that help people with disabilities overcome barriers related to visual, hearing, musculoskeletal and cognitive disorders are analyzed. Special attention is paid to mobile applications and intelligent systems that ensure the independence and social integration of people with disabilities.

Keywords: Artificial intelligence, accessible environment, inclusion, computer vision, speech recognition, machine learning, people with disabilities.

В современном обществе вопросы инклюзивности и создания равных возможностей для всех категорий граждан приобретают особую значимость. Обеспечение доступной среды для лиц с ограниченными возможностями здоровья является не только этическим императивом, но и практической необходимостью, требующей внедрения инновационных технологических решений [1, с. 670].

Искусственный интеллект (ИИ) выступает ключевым инструментом в решении задач, с которыми сталкиваются люди с ограниченными возможностями здоровья (ОВЗ). Технологии на основе ИИ открывают для таких людей новые способы взаимодействия с окружающим

миром, способствуют развитию бытовых навыков, социальной интеграции и повышению уровня независимости. Применение технологий ИИ обусловило беспрецедентное расширение возможностей для людей с инвалидностью, позволив преодолевать барьеры, которые ранее казались непреодолимыми [2, с. 28].

Современные разработки в области ИИ позволяют создавать специализированные технологические решения для людей с разными типами нарушений. В частности, для людей с нарушениями слуха перспективным направлением является использование алгоритмов машинного обучения для улучшения качества звука и шумоподавления. Отметим в этой связи технологию Krisp, которая позволяет выделять важные звуки и эффективно фильтровать фоновый шум, что особенно актуально в шумной городской среде.

Большое значение для людей с нарушениями слуха имеют также приложения для преобразования речи в текст в реальном времени. В качестве такого приложения можно назвать приложение Live Transcribe от Google, которое позволяет людям с нарушениями слуха полноценно участвовать в разговорах. Такие приложения поддерживают множество языков.

Для людей с нарушениями зрения также разработано множество технологических решений на основе ИИ. Прорывными технологиями в данном направлении выступают приложения, позволяющие описывать окружающую обстановку в реальном времени с использованием компьютерного зрения [3, с. 113]. В качестве примера такого приложения можно назвать приложение Seeing AI от Microsoft. Данное приложение помогает пользователям распознавать текст, идентифицировать объекты и даже определять лица людей, обеспечивая более безопасную навигацию и понимание окружающей среды.

Также для людей с нарушениями зрения большую значимость имеют технологии оптического распознавания символов (OCR), которые позволяют озвучивать печатные и рукописные тексты, делая доступными для таких людей книги, документы. Однако, при использовании соответствующих технологий необходимо принимать во внимание то обстоятельство, что такие технологии не могут быть абсолютно точными, и при работе с важными документами целесообразно привлекать помощь зрячего человека.

Для лиц с когнитивными нарушениями также разработаны различные технологические решения на основе ИИ. В частности, применяются инструменты для тренировки бытовых навыков, технологии, позволяющие напоминать таким лицам о важных делах, а также инструменты, позволяющие использовать адаптивные методики обучения. Системы обучения, основанные на ИИ, могут формировать индивидуальные траектории освоения материала, что способствует более эффективному усвоению информации и развитию самостоятельности лиц с когнитивными нарушениями.

Для людей с двигательными нарушениями особое значение имеют технологии управления жестами и голосом. Существующие приложения для таких людей, использующие ИИ, могут распознавать движения головы, глаз, а также голосовые команды, позволяя пользователям управлять устройствами без использования рук [1, с. 671].

В основе большинства инклюзивных технологий лежат достижения в области машинного обучения. Благодаря способности обучаться на больших массивах данных, программные алгоритмы могут распознавать образы, адаптироваться к индивидуальным потребностям пользователя и предоставлять персонализированные рекомендации.

Основными технологическими компонентами современных технологий ИИ в обеспечении доступной среды для лиц с ОВЗ в настоящее время выступают: компьютерное зрение, позволяющее интерпретировать визуальную информацию (распознавание объектов, сцен, лиц и др.); обработка естественного языка (NLP), позволяющая преобразовывать речь в текст, создавать субтитры и озвучивать текстовую информацию; сенсорные технологии и анализ биометрических данных, способствующие созданию адаптивных интерфейсов.

В настоящее время происходит активное объединение интеллектуальных обучающих систем (ИОС) с ИИ. В общем виде, структура такой системы включает три взаимодействующие модели: модель предметной области, модель обучающегося и педагогическую модель. На основе ИИ в рамках ИОС осуществляется сбор данных о действиях пользователя, проводится анализ его ошибок и эмоционального состояния, что позволяет выстраивать наиболее эффективную стратегию обучения.

Внедрение технологий ИИ в различные технологические решения, направленные на обеспечение доступной среды для лиц с ОВЗ имеет как существенные и неоспоримые преимущества и достоинства, так и проблемы, которые необходимо учитывать. Рассмотрим данные проблемы.

1. Техническое несовершенство данных технологий. По-прежнему сохраняются проблемы при распознании речи, происходит неверная идентификация объектов, также возможны задержки в ответах, которые могут снижать эффективность созданных приложений и создавать дополнительные барьеры для лиц с ОВЗ.

2. Дороговизна данных технологических решений. Многие специализированные программные решения остаются достаточно дорогими, что ограничивает их доступность для широких слоев населения. Разработка более доступных альтернативных решений является важной задачей в обозримом будущем.

3. Создаваемые приложения должны быть интуитивно понятными, особенно для людей с когнитивными нарушениями. Проблемой остается создание простого интерфейса для таких технических решений.

4. Проблемой является также то, что многие модели машинного обучения требуют больших вычислительных ресурсов, что делает приложения, создаваемые на их основе, зависимыми от постоянного высокоскоростного интернета.

5. Определенные опасения вызывает то обстоятельство, что в образовательной и социальной сферах существует опасность снижения творческих способностей людей с ОВЗ при применении рассматриваемых технологий, подмены живого общения цифровыми алгоритмами. ИИ не способен воспринимать многообразные эмоциональные состояния (волнение, радость, удивление и др.) и не обладает интуицией или эмпатией, что делает невозможным замены им человеческого участия [4, с. 59].

Развитие технологий глубокого обучения, обработки естественного языка и создание на основе ИИ технологий «умного дома» открывают новые возможности для повышения качества жизни людей с ОВЗ [5, с. 248]. Государственная политика в области цифрового развития, включая программы по устранению цифрового неравенства и созданию «умных» регионов, играет ключевую роль в обеспечении равного доступа к этим технологиям.

Таким образом, современные технологии ИИ обладают огромным потенциалом для обеспечения доступной среды для лиц с ОВЗ. Такие технологии предоставляют людям с ОВЗ

инструменты для преодоления коммуникационных, пространственных и когнитивных барьеров. Мобильные приложения, интеллектуальные системы распознавания и обучения способствуют социальной интеграции лиц с ОВЗ.

Список литературы

1. Буянов М.П., Чернышева А.С., Рудяк Ю.В. Использование искусственного интеллекта в мобильных приложениях для людей с ограниченными возможностями // Вестник науки. – 2024. – № 5 (74). – Т. 2. – С. 669–677.
2. Видова Т.А., Романова И.Н. Возможности применения технологий искусственного интеллекта в образовательном процессе // Образовательные ресурсы и технологии. – 2023. – № 1 (42). – С. 27–35.
3. Клетте Р. Компьютерное зрение. Теория и алгоритмы. – М.: ДМК-Пресс, 2019. – 506 с.
4. Уваров А.Ю., Гейбл Э., Дворецкая И.В. и др. Трудности и перспективы цифровой трансформации образования / под ред. А.Ю. Уварова, И.Д. Фрумина; Нац. исслед. ун-т «Высшая школа экономики», Ин-т образования. – М.: Изд. дом Высшей школы экономики, 2019. – 342 с.
5. Юнина Е.С. Анализ возможностей внедрения искусственного интеллекта в организацию умного дома для людей с ограниченными возможностями // Экономика: вчера, сегодня, завтра. – 2025. – № 4А. – С. 246-254.

References

1. Buyanov M.P., Chernysheva A.S., Rudyak Yu.V. Using Artificial Intelligence in Mobile Applications for People with Disabilities // Science Bulletin. - 2024. - No. 5 (74). - Vol. 2. - pp. 669-677.
2. Vidova T.A., Romanova I.N. Possibilities of Using Artificial Intelligence Technologies in the Educational Process // Educational Resources and Technologies. - 2023. - No. 1 (42). - pp. 27-35.
3. Klette R. Computer Vision. Theory and Algorithms. - Moscow: DMK-Press, 2019. - p.506
4. Uvarov A.Yu., Gable E., Dvoretskaya I.V., et al. Difficulties and Prospects of Digital Transformation of Education / edited by A.Yu. Uvarova, I.D. Frumina; National Research University Higher School of Economics, Institute of Education. – Moscow: Publishing House of the Higher School of Economics, 2019. – p.342
5. Yunina E.S. Analysis of the Possibilities of Implementing Artificial Intelligence in the Organization of a Smart Home for People with Disabilities // Economy: Yesterday, Today, Tomorrow. – 2025. – No. 4A. – pp. 246-254.

Демич А.А., Митянина А.В. Разработка интеллектуального помощника для студентов и сотрудников ВУЗА// Международный журнал информационных технологий и энергоэффективности.– 2026. –Т. 11 № 6(68) Ч. 1 с. 91–100

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.89

РАЗРАБОТКА ИНТЕЛЛЕКТУАЛЬНОГО ПОМОЩНИКА ДЛЯ СТУДЕНТОВ И СОТРУДНИКОВ ВУЗА

[1] **Демич А.А., Митянина А.В.**
ФГБОУ ВО "ЧЕЛЯБИНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ", Челябинск, Россия (454001, Челябинская область, город Челябинск, ул Братьев Кашириных, д. 129), e-mail: [1]demich-2002@mail.ru

В условиях цифровизации высшего образования возрастает объём информации, с которой ежедневно взаимодействуют студенты и сотрудники вузов. В статье рассматривается проблема фрагментированности цифровых сервисов университета, затрудняющая доступ обучающихся к оперативной информации и создающая избыточную нагрузку на административный персонал. Для решения данной проблемы спроектирован и реализован интеллектуальный помощник с микросервисной архитектурой, объединяющий в себе модуль диалогового поиска по нормативной документации (на базе технологии RAG) и модуль предиктивной аналитики успеваемости.

Ключевые слова: Интеллектуальные системы, искусственный интеллект, RAG-архитектура, машинное обучение, предиктивная аналитика, данные об успеваемости, микросервисная архитектура.

DEVELOPMENT OF AN INTELLIGENT ASSISTANT FOR UNIVERSITY STUDENTS AND STAFF

[1] Demich A. A., Mityanina A. V.
CHELYABINSK STATE UNIVERSITY, Chelyabinsk, Russia (454001, Chelyabinsk region, Chelyabinsk city, Bratyev Kashirinykh st., 129,), e-mail: [1]demich-2002@mail.ru

As higher education undergoes digitalisation, the volume of information that students and university staff interact with on a daily basis is increasing. This article examines the problem of fragmentation in university digital services, which hinders students' access to up-to-date information and places an excessive burden on administrative staff. To address this issue, a microservice architecture for an intelligent assistant has been designed and implemented, combining a dialogue-based search module for regulatory documentation (based on RAG technology) and a predictive analytics module for academic performance.

Keywords: Intelligent systems, artificial intelligence, RAG architecture, machine learning, predictive analytics, educational data, microservices architecture.

Современные университеты активно внедряют цифровые технологии для сопровождения образовательного процесса. Тем не менее, на практике информационная инфраструктура вуза часто представляет собой совокупность слабо связанных между собой систем: электронных журналов, личных кабинетов, корпоративных информационных систем и хранилищ нормативных документов. [1] В контексте высшего медицинского университета (на примере Южно-Уральского государственного медицинского университета) эти процессы приобретают особенно высокую значимость. Учебный процесс студента-медика отличается высокой интенсивностью, строгой регламентацией и колоссальным объемом сложного материала по дисциплинам.

В этих условиях выявляются две критические проблемы, препятствующие эффективному управлению образовательным процессом:

1. Информационная перегрузка и фрагментация среды. Нормативная документация (правила ликвидации задолженностей, порядки прохождения практик) хранится в виде неструктурированных файлов на официальных порталах, тогда как оперативные данные (расписание, текущие оценки) находятся в корпоративной информационной системе (КИС) «1С: Университет». Из-за необходимости обращаться к различным источникам студенты тратят значительное время на поиск базовой информации вместо фокусировки на учебе. Не находя нужный ответ, студентам приходиться обращаться к административному персоналу (струдники кафедр, деканаты, учебный центр). Как показывает практика, большинство обращений в деканаты носит рутинный, справочный характер, что вынуждает сотрудников ежедневно работать в режиме справочника, приводя к перегрузкам и выгоранию. Кроме того, самим сотрудникам требуются данные из нормативных актов, которые невозможно удерживать в памяти, что также ведет к потере рабочего времени на ручной поиск.[2]

2. Отсутствие инструментов превентивного анализа академических трудностей. Традиционная модель контроля успеваемости носит констатирующий характер: проблема фиксируется только тогда, когда студент уже получил задолженность или не сдал экзамен. Отчисление студента несет для вуза прямые финансовые потери и снижает ключевые показатели эффективности. Раннее выявление потенциальных рисков позволило бы администрации применять превентивные меры: назначать дополнительные консультации, привлекать тьюторов или проводить профилактические беседы.

Задачи анализа успеваемости активно исследуются, однако попытка прямого внедрения готовых решений сталкивается с рядом ограничений.

Зарубежные системы: обладают встроенными предиктивными модулями, однако их математические модели опираются на Болонскую систему, что делает прогнозы нерелевантными для российского специалитета с жестко фиксированными учебными планами

Отечественные КИС: успешно справляются с транзакционными задачами (учет контингента, ведомости), однако констатируют факты и не применяют методы машинного обучения для предсказания результатов на основе исторических данных.[3]

Таким образом, необходимо создание собственного сервиса, адаптированного под специфику отечественного высшего медицинского учреждения. Целевая аудитория системы включает авторизованных обучающихся и сотрудников вуза.

Сформулированы следующие функциональные и нефункциональные требования к системе:

1. Обеспечение авторизации пользователей с привязкой к ролевой модели КИС.
2. Использование исключительно внутренних данных вуза для исключения генерации недостоверных данных.
3. Масштабируемость и отказоустойчивость при пиковых нагрузках.
4. Информационная безопасность: изоляция контуров обработки персональных данных студентов и деперсонализация запросов к языковым моделям.
5. Кроссплатформенность клиентского приложения с использованием единой кодовой базы.

6. Аналитический модуль должен в фоновом режиме анализировать цифровой след студентов и формировать вероятностный прогноз успешной сдачи промежуточной аттестации.

7. Модуль информационной поддержки должен уметь вести диалог, извлекая ответы строго из загруженной базы документов вуза для исключения галлюцинаций.

Исходя из сформулированных требований и преодоления ограничений монолитных систем, в работе осуществлено использование гибкой микросервисной архитектуры. Принцип разделения ответственности обусловлен тем, что задачи интеллектуального текстового поиска (RAG) и предиктивного моделирования (ML) имеют различную математическую природу и требуют дифференцированных вычислительных ресурсов. Модульный подход позволяет разрабатывать и тестировать функциональные блоки, что значительно упрощает их сопровождение и обеспечивает высокий потенциал для гибкого развития и масштабирования системы в будущем.[4]

Спроектированная архитектура разделена на четыре основных уровня:

1. Слой хранения и управления данными: фундамент системы, обеспечивающий хранение и интеграцию структурированных и неструктурированных данных.

2. Интеграционный слой и маршрутизация: связующее звено, отвечающее за асинхронное взаимодействие компонентов и распределение нагрузки.

3. Слой интеллектуальных сервисов: вычислительное ядро системы, представленное независимыми микросервисами поиска (RAG) и аналитики (ML).

4. Клиентский слой: уровень пользовательских интерфейсов, обеспечивающий доступ к функциям ассистента.

Источники данных системы. [5] Интеллектуальный помощник использует данные, которые можно разделить на два типа:

1. Структурированные данные: оперативная информация из КИС «1С:Университет» (успеваемость, расписание, учебные планы), доступ к которой осуществляется через специально разработанные HTTP-сервисы (REST API).

2. Неструктурированные данные: корпус локальных нормативных актов, приказов и регламентов университета, представленный в форматах PDF, DOCX и HTML.

Слой хранения и управления данными. На данном уровне архитектуры осуществляется интеграция и персистентное хранение данных из вышеуказанных источников:

1. Векторная база данных (ChromaDB): используется для хранения многомерных представлений (эмбеддингов) неструктурированных документов, что обеспечивает работу механизма семантического поиска в RAG-ассистенте.

2. Реляционная база данных (PostgreSQL): выполняет роль служебного хранилища для ведения истории диалогов (сохранение контекста), управления профилями пользователей и кэширования частых поисковых запросов.[6]

Интеграционный слой и маршрутизация. Связующее звено, обеспечивающее асинхронное взаимодействие компонентов. API-шлюз принимает запросы от клиентских приложений и ставит их в очередь отказоустойчивого брокера сообщений (RabbitMQ). Брокер распределяет тяжелые вычислительные задачи (генерация текста или расчет ML-прогноза) между свободными воркерами, защищая систему от падения при пиковых нагрузках. Разработка API-шлюза микросервиса выполнена на базе современного веб-фреймворка FastAPI, который нативно поддерживает асинхронный ввод-вывод. Это предотвращает

блокировку потоков сервера в моменты пиковых студенческих нагрузок, что критически важно, учитывая, что время ответа тяжелой нейросети может достигать нескольких секунд.

Слой интеллектуальных сервисов. Вычислительное ядро, реализованное на языке Python и состоящее из двух независимых микросервисов:

1. Модуль информационной поддержки (RAG-сервис): выполняет семантический поиск по векторной базе, формирует расширенный промпт с контекстом и отправляет его в локально развернутую большую языковую модель (LLM) для генерации финального ответа.
2. Модуль предиктивной аналитики (ML-сервис): осуществляет регламентный забор структурированных данных из 1С, проводит инженерию признаков и прогоняет данные через обученную модель градиентного бустинга для расчета вероятности академической задолженности.[7]

Клиентский слой. Единый пользовательский интерфейс, абстрагирующий пользователя от сложности внутренних процессов. Разработка ведется с применением кроссплатформенной технологии Kotlin Multiplatform и интеграции мини-приложения на языке Go (MAX Mini App) для снижения порога входа обучающихся и обеспечения безопасной OAuth-авторизации.

Данная архитектура обеспечивает гибкость разработки: она позволяет независимо обновлять языковую модель в RAG-модуле или алгоритм прогнозирования, не останавливая работу всего приложения и не вмешиваясь в монолитный код системы 1С.

Качество и достоверность работы диалоговых систем критически зависят от репрезентативности используемых данных. [8] В рамках исследования был принят принцип полного отказа от открытых датасетов, так как они не отражают специфику промежуточной аттестации и прохождения практик в университете. Эмпирической базой выступают исключительно внутренние неструктурированные нормативные акты (приказы, регламенты) и динамические структурированные данные из системы учета ЮУГМУ.

Для обработки неструктурированного контента (документы в форматах HTML, PDF и DOCX) спроектирован специализированный программный модуль парсинга, осуществляющий асинхронный обход веб-страниц порталов вуза. С целью повышения семантической плотности текста применен метод DOM-фильтрации, алгоритм предварительно находит и удаляет (через метод decompose) все неинформативные теги, что исключает попадание в векторную базу навигационного шума и предотвращает смещение внимания языковой модели на случайные контакты или элементы меню.[9]

Процесс подготовки извлеченных текстов включает фрагментацию и нормализацию. Предобработанные документы передаются в алгоритм рекурсивного разбиения. Ключевым архитектурным решением на этапе векторизации стало внедрение механизма дедупликации на основе криптографического хеширования (алгоритм SHA-256). Алгоритм вычисляет уникальный хеш каждого нормализованного фрагмента (чанка); векторизации подлежат только уникальные фрагменты. Это решает проблему дублирования данных, экономит память в векторной СУБД и предотвращает смещение фокуса LLM из-за многократного повторения одного факта.

Связующим звеном архитектуры генерации, дополненной поиском (RAG), выступает векторная база данных. Сравнительный анализ популярных решений показал, что использование облачного сервиса недопустимо политикой информационной безопасности вуза, а локальная база FAISS (Meta) обладает ограничениями в части персистентности и фильтрации метаданных. Для практической реализации выбрана встраиваемая СУБД

ChromaDB, оптимизированная для локальной работы, поддерживающая персистентное хранение и HNSW-поиск.[10]

Требования Федерального закона РФ № 152-ФЗ «О персональных данных» запрещают передачу контекста на серверы внешних провайдеров. В связи с этим вычислительное ядро развернуто локально на платформе Ollama с использованием 4-битного квантования весов, что позволило кардинально снизить требования к видеопамяти графических ускорителей. Как будет показано ниже в описании экспериментов, обусловивших выбор компонентов системы, будет выбрана модель Qwen3.5-4B в качестве генеративного ядра, обеспечивающая баланс между требованиями к аппаратному обеспечению (~3 ГБ VRAM) и высоким качеством генерации структурированных ответов на русском языке.

Качество финального ответа интеллектуального помощника зависит от гиперпараметров конвейера обработки данных. Была разработана серия экспериментальных стендов для эмпирической оценки и настройки каждого изолированного компонента системы.[11]

Исследовалось влияние размера текстового блока на семантическую целостность. На валидационном датасете были сгенерированы три конфигурации: мелкая (300 токенов, перекрытие 50), базовая (600 токенов, перекрытие 120) и крупная нарезка (1200 токенов, перекрытие 200). Эксперимент показал, что база мелкой нарезки (675 фрагментов) вырывает предложения из контекста, теряя логические связи регламентов, в то время как база крупной нарезки (159 фрагментов) приводит к размытию семантики и переполнению оперативной памяти. Базовая конфигурация (сгенерировавшая 331 фрагмент) была выбрана как лучшая, поскольку она вмещает целостный смысловой абзац документа, не перегружая контекстное окно языковой модели.[12]

Сравнивалась семантическая точность перевода текста в многомерный вектор на примере трех моделей: англоязычной nomic-embed-text (768 измерений), Multilingual-E5-large (1024 измерения) и qwen3-embedding:4b (1536 измерений). Оценка проводилась методом similarity_search_with_score (фреймворк LangChain). Выбор данного метода обусловлен необходимостью получения не только извлеченного текста, но и количественной метрики релевантности. Наличие числового значения позволило объективно сравнить семантическую точность различных моделей-энкодеров на едином тестовом наборе данных. В качестве данных использовались 30 страниц с сайта и 20 нормативных документов с различным наполнением. Англоязычная модель (nomic-embed-text) продемонстрировала высокую скорость, но потерпела полный семантический провал (поиск вернул документ о столовой вместо регламента пересдач), что доказывает не применимость моделей без предварительного обучения на русскоязычных корпусах. В свою очередь мультиязычная модель (Multilingual-E5-large) хоть и продемонстрировала высокую семантическую точность и высокую скорость поиска (0.0815 сек), но при этом не дошла до результатов qwen3, при более высоком потреблении ресурсов. Модель qwen3-embedding:4b показала высокую скорость поиска (0.0743 сек) и корректно извлекла релевантный нормативный акт, обеспечив баланс между качеством понимания контекста и скоростью работы в локальном контуре.

Важнейшим гиперпараметром является top-k, определяющий количество извлекаемых фрагментов, передаваемых в языковую модель. Замеры показали, что при минимальном значении (k=1, длина контекста 148 символов) модель фокусируется на частных случаях из-за недостатка данных. При избыточном значении (k=10, контекст более 5200 символов) зафиксирована галлюцинация: из-за попадания в промпт документов низкого ранга нейросеть

смешала понятия академического отпуска и задолженности. Параметр k=3 (длина контекста ~1338 символов) выбран как рабочий, гарантирующий исчерпывающий ответ без риска зашумления промпта.

Кроме того, плотные векторные представления часто уступают классическим лексическим алгоритмам при поиске точных совпадений (специфических аббревиатур или номеров справок, например «095/у»). Был спроектирован стенд гибридного ансамблевого поиска (Ensemble Retriever), объединяющий векторный поиск и статистический алгоритм BM25 (с весами 0.5/0.5). Гибридный поиск позволил нивелировать недостатки каждого метода в отдельности, максимизируя полноту извлекаемого контекста.

Также проводился эксперимент с алгоритмом HyDE для повышения точности поиска. Его суть в том, чтобы генерировать примерный ответ на вопрос пользователя и переводить его в более понятный язык для поиска в векторной базе. Предполагалось, что данный алгоритм повысит точность, однако он показал его избыточность при использовании мощной модели qwen3 и был исключен из базовой архитектуры для снижения общей задержки.

Для предотвращения заполнения контекста дублирующимися абзацами классический косинусный поиск (Similarity Search) был сравнен с алгоритмом максимальной маржинальной релевантности (MMR). Суть эксперимента заключалась в подаче запроса с комплексным вопросом, объединяющего в себе два вопроса (Как получить стипендию и каков ее размер?) к базе содержащей семантические дубликаты. Эксперимент доказал способность алгоритма MMR гарантировать разнообразие контекста без потери производительности (скорость поиска 0.0643 сек).

Процесс обработки запроса в реальном времени реализован в виде гибридного пайплайна: система извлекает K=3 релевантных фрагментов из векторной БД и в случае запроса студента параллельно выполняет REST API-запрос к КИС «1С: Университет» для извлечения оперативного цифрового профиля студента (оценки, задолженности). Данные программно объединяются в единый системный промпт, что обеспечивает абсолютную персонализацию: модель опирается на актуальную выгрузку из 1С, формируя точные административные рекомендации.

Модуль предиктивной аналитики предназначен для перехода от констатирующего контроля успеваемости к раннему выявлению студентов с высоким риском отчисления. Решение данной задачи требует перевода транзакционных данных КИС в структурированный вектор признаков и подбора оптимального алгоритма бинарной классификации.

Выгрузка исторических данных (около 500 000 уникальных записей) реализована через специализированный программный модуль на встроенном языке 1С. Динамический набор данных формируется путем запросов к регистрам: «Посещаемость» (количество присутствий и пропусков), «Результаты вступительных испытаний» (средний балл ЕГЭ) и «Аттестация» (результаты экзаменов, историческое количество пересдач). Каждая строка итогового датасета представляет собой вектор состояния студента по одной учебной дисциплине.

На этапе математической предобработки выполнена строгая деперсонализация и форматирование числовых переменных. Ключевым шагом инженерии стала бинаризация целевой переменной: исходная мультиклассовая задача (оценки от «Отлично» до «Неявка») была сведена к двум классам. В Класс 1 («Группа риска») объединены оценки «Неудовлетворительно», «Не зачтено» и «Неявка», а в Класс 0 («Успешная сдача») — все

положительные результаты. Это позволило сфокусировать алгоритмы на поиске паттернов академической задолженности.

Экспериментальный стенд сравнивал три класса алгоритмов: логистическую регрессию, random forest и градиентный бустинг (CatBoost). Логистическая регрессия была выбрана в качестве baseline, так как данный алгоритм позволяет оценить, являются ли классы линейно разделимыми. Random forest в свою очередь позволяет решить проблему нелинейности данных, но не способен нативно обрабатывать категориальные признаки, требуя их обязательного предварительного кодирования. Градиентный бустинг в частности Catboost способен использовать категориальные признаки. Данные разделены на Train/Test в пропорции 80/20 со стратификацией stratify=y для сохранения доли студентов группы риска.

В ходе экспериментов классические алгоритмы (логистическая регрессия и random forest) потребовали применения прямого кодирования для текстового признака «Дисциплина» и стандартизации StandardScaler. Это привело к избыточному разрастанию размерности матрицы и фрагментации данных. Алгоритм CatBoost принимал строковые названия, напрямую самостоятельно осуществляя целевое кодирование, что сэкономило ресурсы и предотвратило утечку данных.

Тестирование на отложенной выборке доказало превосходство градиентного бустинга. Линейная модель показала низкую способность к обобщению (F1-Score = 0.5647), так как влияние пропусков на итоговую оценку имеет сложную нелинейную зависимость от сложности предмета. Случайный лес показал приемлемую точность, но уступил по ключевой метрике полноты. CatBoost продемонстрировал наивысшие показатели: Accuracy = 0.9530, Precision = 0.8611 и Recall = 0.8155, успешно выявляя более 81% студентов на грани отчисления.

Для математического доказательства отсутствия переобучения (Overfitting) проведен эксперимент со стратифицированной K-Fold кросс-валидацией (5 фолдов). Результаты зафиксировали среднее значение F1-Score на уровне 0.8312 со стандартным отклонением 0.014. Низкая дисперсия ошибки (<1.5% между фолдами) статистически доказывает, что модель CatBoost находит устойчивые глобальные паттерны, а не заучивает данные наизусть.

Оптимизация выбранной модели CatBoost проводилась с учетом бизнес-логики деканата. Академические данные имеют структурный дисбаланс (доля отчисляемых составляет около 20%). Для устранения тривиального предсказания мажоритарного класса внедрен механизм алгоритмического взвешивания (auto_class_weights='Balanced'), штрафующий функцию потерь за ошибку на миноритарном классе, что исключило необходимость использования зашумляющих синтетических данных.

Методом поиска по сетке (Grid Search) выбраны гиперпараметры, предотвращающие переобучение глубоких деревьев: iterations=200, depth=6, learning_rate=0.05.

Учитывая, что ложноотрицательное срабатывание (пропуск потенциального должника) ведет к отчислению и потере финансирования вузом, был проведен сдвиг порога классификации (Threshold). Снижение порога со стандартного 0.50 до 0.30 алгоритмически перевело модель в «пессимистичный» режим, что снизило Precision до 0.62, но максимизировало выявляемость группы риска (Recall) до 0.94.

Для обеспечения прозрачности системы проведен анализ важности признаков встроенными методами CatBoost. Установлена следующая иерархия предикторов академической неуспеваемости:

1. Процент пропусков занятий (~45%). Сильнейший предиктор: риск экспоненциально возрастает при превышении 15-20% пропущенных часов.
2. Учебная дисциплина (~25%). Выявлены дисциплины-маркеры («Биохимия», «Анатомия»), задолженность по которым резко повышает вероятность отчисления по сравнению с факультативами.
3. Историческое количество пересдач (~15%). Кумулятивный индикатор системных проблем обучающегося.
4. Средний балл ЕГЭ (~10%). Высокая предсказательная сила на первых курсах с постепенным затуханием на старших.

В рамках исследования решена проблема фрагментированности цифровых сервисов медицинского университета и дефицита предиктивных аналитических инструментов. Разработанная микросервисная архитектура интеллектуального помощника обеспечивает бесшовную и безопасную работу с персональными данными в изолированном контуре вуза.

Экспериментально подтверждена эффективность гибридного RAG-конвейера (на базе ChromaDB и LLM Qwen3.5-4B), исключающего проблему галлюцинаций за счет строгой опоры на локальные нормативные акты. Реализованный модуль машинного обучения (CatBoost) доказал свою высокую предсказательную способность (F1-Score: 0.8312), а проведенная оптимизация порога классификации позволила системе выявлять до 94% студентов группы риска на ранних стадиях. Внедрение архитектуры обеспечит существенное снижение административной нагрузки на сотрудников деканатов и создаст технологическую основу для превентивной работы по сохранению контингента обучающихся.

Список литературы

1. Федеральный закон от 27 июля 2006 г. № 152-ФЗ «О персональных данных» (с изм. и доп.). — Текст: электронный— URL: https://www.consultant.ru/document/cons_doc_LAW_61801/ (дата обращения: 12.04.2026).
2. Мюллер, Андреас, Гвидо, Сара. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. : Пер. с англ. — СПб. : ООО "Альфа-книга", 2017. - 480 с.: ил. - Парал. тит. англ. ISBN 978-5-9908910-8-1 (рус.)
3. Радченко, М. Г. 1С:Предприятие 8.3. Практическое пособие разработчика / М. Г. Радченко, Е. Ю. Хрусталева. — Москва : 1С-Паблишинг, 2013. — 926 с. Текст : электронный— URL: https://online.1c.ru/books/book/17628399 (дата обращения: 12.04.2026).
4. Флах, П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / П. Флах. — Москва : ДМК Пресс, 2015. — 400 с.
5. CatBoost - open-source gradient boosting library : official documentation. — Текст : электронный // CatBoost : [сайт]. — URL: https://catboost.ai/en/docs/ (дата обращения: 10.04.2026).
6. Chroma: the AI-native open-source embedding database. — Текст : электронный // Chroma : [сайт]. — URL: https://docs.trychroma.com/ (дата обращения: 11.04.2026).
7. FastAPI framework, high performance, easy to learn, fast to code, ready for production. — Текст : электронный // FastAPI : [сайт]. — URL: https://fastapi.tiangolo.com/ (дата обращения: 12.04.2026).

8. LangChain Documentation: Building applications with LLMs through composability. — Текст : электронный // LangChain : [сайт]. — URL: https://python.langchain.com/docs/get_started/introduction (дата обращения: 14.04.2026).
9. Lewis, P. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis, E. Perez, A. Piktus // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — P. 9459-9474. — Текст : непосредственный.
10. Богатырев, С. Ю. Машинное обучение в финансах : учебник / С. Ю. Богатырев, А. А. Помулев, А. В. Затевахина ; под редакцией С. Ю. Богатырева. — Москва : Прометей, 2024. — 224 с. — ISBN 978-5-00172-572-5. — Текст : электронный
11. Митяков, Е. С. Искусственный интеллект и машинное обучение : учебное пособие для вузов / Е. С. Митяков, А. Г. Шмелева, А. И. Ладынин. — 2-е изд., стер. — Санкт-Петербург : Лань, 2026. — 252 с. — ISBN 978-5-507-51198-3. — Текст : электронный
12. Филиппова, А. С. Курс лекций по дисциплине «Цифровые технологии в научно-исследовательской и управленческой деятельности» : учебно-методическое пособие / А. С. Филиппова, Э. И. Дяминова, Ф. З. Забихуллин. — Уфа : БГПУ имени М. Акмуллы, 2025. — 114 с. — ISBN 978-5-00251-024-5. — Текст : электронный

References

1. Federal Law No. 152-FZ of 27 July 2006 ‘On Personal Data’ (as amended and supplemented). — Text: electronic— URL: https://www.consultant.ru/document/cons_doc_LAW_61801/ (accessed: 12 April 2026).
2. Müller, Andreas, Guido, Sarah. Introduction to Machine Learning with Python. A Guide for Data Professionals. Translated from English. — St Petersburg: Alfa-Kniga LLC, 2017. — p.480: ill. — Parallel title in English. ISBN 978-5-9908910-8-1 (Russian)
3. Radchenko, M. G. 1C:Enterprise 8.3. A Practical Guide for Developers / M. G. Radchenko, E. Yu. Khrustaleva. — Moscow: 1C-Publishing, 2013. — p. 926 Text: electronic— URL: https://online.1c.ru/books/book/17628399 (accessed: 12 April 2026).
4. Flah, P. Machine Learning. The Science and Art of Building Algorithms that Extract Knowledge from Data / P. Flah. — Moscow: DMK Press, 2015. — p. 400
5. CatBoost - open-source gradient boosting library: official documentation. — Text: electronic // CatBoost: [website]. — URL: https://catboost.ai/en/docs/ (accessed: 10 April 2026).
6. Chroma: the AI-native open-source embedding database. — Text : electronic // Chroma : [website]. — URL: https://docs.trychroma.com/ (accessed: 11 April 2026).
7. FastAPI framework, high performance, easy to learn, fast to code, ready for production. — Text : electronic // FastAPI : [website]. — URL: https://fastapi.tiangolo.com/ (accessed: 12 April 2026).
8. LangChain Documentation: Building applications with LLMs through composability. — Text: electronic // LangChain: [website]. — URL: https://python.langchain.com/docs/get_started/introduction (accessed: 14 April 2026).
9. Lewis, P. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks / P. Lewis, E. Perez, A. Piktus // Advances in Neural Information Processing Systems. — 2020. — Vol. 33. — pp. 9459–9474. — Text: direct.

10. Bogatyrev, S. Yu. Machine Learning in Finance: A Textbook / S. Yu. Bogatyrev, A. A. Pomulev, A. V. Zatevakina; edited by S. Yu. Bogatyrev. — Moscow: Prometheus, 2024. — p. 224— ISBN 978-5-00172-572-5. — Text: electronic
11. Mityakov, E. S. Artificial Intelligence and Machine Learning: A Textbook for Universities / E. S. Mityakov, A. G. Shmeleva, A. I. Ladynin. — 2nd ed., rev. — Saint Petersburg: Lan, 2026. — p.252 — ISBN 978-5-507-51198-3. — Text: electronic
12. Filippova, A. S. Lecture Course on the Subject ‘Digital Technologies in Research and Management Activities’: Teaching Guide / A. S. Filippova, E. I. Dyaminova, F. Z. Zabikhullin. — Ufa: M. Akmulla Ufa State Pedagogical University, 2025. — p. 114— ISBN 978-5-00251-024-5. — Text: electronic

Филатов Р. А., Мордвинова А.Ю. Архитектурные принципы защиты операционных систем LINUX: рекомендации федерального регулятора// Международный журнал информационных технологий и энергоэффективности.– 2026. –Т. 11 № 6(68) Ч.1 с. 101–104

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056:004.7.056

АРХИТЕКТУРНЫЕ ПРИНЦИПЫ ЗАЩИТЫ ОПЕРАЦИОННЫХ СИСТЕМ LINUX: РЕКОМЕНДАЦИИ ФЕДЕРАЛЬНОГО РЕГУЛЯТОРА

Филатов Р. А., [1]**Мордвинова А.Ю**.
ФГАОУ ВО "СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ", Севастополь, Россия (299053, город Севастополь, Университетская ул., зд. 33), e-mail: [1] aumordvinova@mail.sevsu.ru

В статье рассматривается комплексный подход к конфигурированию операционных систем семейства Linux в соответствии с методическими рекомендациями Федеральной службы по техническому и экспортному контролю (ФСТЭК) России.

Ключевые слова: безопасность Linux, критическая информационная инфраструктура, методические рекомендации ФСТЭК, управление доступом, безопасная конфигурация ОС.

ARCHITECTURAL PRINCIPLES FOR LINUX OPERATING SYSTEM SECURITY: FEDERAL REGULATOR RECOMMENDATIONS

Filatov R. A., [1]**Mordvinova A. Yu.**
SEVASTOPOL STATE UNIVERSITY, Sevastopol, Russia (299053, Sevastopol, Universitetskaya st., 33), e-mail: [1] aumordvinova@mail.sevsu.ru

This article examines a comprehensive approach to configuring Linux operating systems in accordance with the guidelines of the Federal Service for Technical and Export Control (FSTEC) of Russia.

Keywords: Linux security, critical information infrastructure, FSTEC guidelines, access control, secure OS configuration

Введение

В условиях цифровизации государственной инфраструктуры и объектов критической информационной защиты перед IT-специалистами стоит задача обеспечения надёжной безопасности операционных систем семейства Linux. Федеральная служба по техническому и экспортному контролю разработала методические рекомендации, определяющие комплексный подход к конфигурированию ОС на основе ядра Linux в государственных информационных системах и объектах критической инфраструктуры РФ до момента полного перехода на сертифицированные отечественные решения. Переход к открытым программным решениям в государственном секторе обусловлен не только требованиями технологического суверенитета, но и необходимостью обеспечения прозрачности кода и гибкости настройки защитных механизмов. В связи с этим администраторам информационных систем необходимо внедрять принципы защиты по умолчанию и минимально необходимых привилегий, что напрямую коррелирует с базовыми требованиями регуляторов и формирует основу для построения эшелонированной обороны.

Основная часть

Фундамент безопасности любой операционной системы закладывается на уровне управления учётными записями. Первоочередное внимание должно уделяться исключению использования учётных записей с отсутствующими парольными защитами. Каждому активному пользователю системы необходимо установить уникальный пароль или заблокировать учётную запись через редактирование параметров файла /etc/shadow. Помимо базового блокирования пустых паролей, рекомендуется настроить политики сложности, минимальной длины и срока действия учётных данных. [1] Для этого используются модули PAM, такие как pam_pwquality, позволяющие задавать требования к наличию символов разных регистров и цифр, а также ограничивать повторное использование предыдущих паролей. Дополнительно целесообразно внедрить автоматическую временную блокировку учётных записей после заданного числа неудачных попыток входа, что эффективно противодействует автоматизированным атакам подбора учётных данных.

Критически важным является отключение возможности удалённого входа суперпользователя через SSH-протокол посредством установки параметра PermitRootLogin no в конфигурационном файле /etc/ssh/sshd_config. [2] Такой подход минимизирует риск несанкционированного доступа на уровне наивысших привилегий. В дополнение к запрету прямого входа от имени root рекомендуется перевести аутентификацию в SSH на использование криптографических ключей, отключив вход по паролю директивой PasswordAuthentication no. Изменение стандартного порта службы и применение средств динамической фильтрации, таких как Fail2Ban или встроенные механизмы iptables/nftables, создают дополнительный рубеж обороны, затрудняя автоматизированное сканирование портов и подбор учётных записей злоумышленниками.

Механизмы получения повышенных привилегий требуют строгого контроля. Команда su должна быть ограничена только для определённого круга пользователей посредством добавления соответствующей строки в файл PAM-конфигурации (/etc/pam.d/su). Список авторизованных пользователей указывается в группе wheel в файле /etc/group. Аналогично, доступ к команде sudo должен быть пересмотрен и ограничен через редактирование файла /etc/sudoers, где администратор может явно указать, какие пользователи могут выполнять какие команды с повышенными привилегиями. Особое значение приобретает обязательное журналирование всех действий, выполняемых с повышенными привилегиями. Настройка подсистемы auditd позволяет фиксировать не только успешные и неудачные попытки использования sudo или su, но и отслеживать изменения в критических конфигурационных файлах, запуск нестандартных процессов и обращения к защищённым системным ресурсам. Интеграция журналов с централизованными SIEM-платформами обеспечивает возможность оперативного реагирования на инциденты и формирования отчётности для регуляторных проверок.[3]

Правильная настройка прав доступа представляет собой второй уровень защиты. Файлы, содержащие информацию об учётных записях (/etc/passwd, /etc/group), должны иметь права 644, что позволяет всем пользователям их читать, но модифицировать могут только владельцы. В то же время, файлы хранилищ хешей паролей (/etc/shadow в системах GNU/Linux, Solaris, HP-UX или /etc/security/passwd в AIX) должны быть доступны только администраторам системы с правами 700 или эквивалентными (go-rwx).

Все исполняемые файлы, запущенные в текущий момент, и соответствующие им системные библиотеки не должны быть доступны для записи непривилегированным пользователям. Это достигается через команду вида chmod go-w /путь/к/файлу с последующей проверкой прав доступа на родительские директории. Особое внимание требуется к стартовым скриптам системы в директориях /etc/rc#.d и systemd-модулям (.service-файлы), права доступа к которым должны быть установлены посредством chmod o-w.

Файлы заданий планировщика cron (как системные в /etc/crontab, /etc/cron.d, /etc/cron.hourly, так и пользовательские) должны быть защищены от записи с помощью chmod go-wx. Критически важным является защита SUID/SGID-приложений: все такие приложения должны быть подвергнуты аудиту, и с тех из них, которые не являются необходимыми, должны быть сняты соответствующие биты. [4] Домашние директории пользователей требуют установки прав 700, а файлы истории команд и конфигурации оболочки (.bash_history, .bashrc, .profile) должны быть недоступны для просмотра и модификации другими пользователями. На уровне файловой системы целесообразно применять дополнительные параметры монтирования, такие как noexec, nosuid и nodev для разделов, содержащих пользовательские данные или временные файлы. Это предотвращает запуск произвольного кода и эксплуатацию уязвимостей через специальные биты в непредназначенных для этого директориях. Регулярное обновление ядра и пакетного состава, а также автоматизированный аудит конфигураций с использованием специализированных скриптов или инструментов соответствия стандартам CIS Benchmarks, позволяют своевременно выявлять отклонения от безопасного базового образа системы и закрывать уязвимости до их использования злоумышленниками.

Заключение

Безопасность операционных систем Linux требует комплексного подхода, объединяющего правильную конфигурацию подсистемы управления доступом, защиту критических файловых ресурсов, оптимизацию параметров ядра для минимизации вектора атак, изоляцию компонентов и активный мониторинг. Важно подчеркнуть, что единовременная настройка защитных механизмов не гарантирует долгосрочной устойчивости инфраструктуры. Безопасность Linux-систем в критической среде требует непрерывного цикла оценки, обновления политик и адаптации к изменяющемуся ландшафту угроз. Проведение периодических внутренних аудитов, тестирование на проникновение и обучение административного персонала актуальным практикам безопасной эксплуатации становятся неотъемлемыми элементами жизненного цикла защищённой системы. Методические рекомендации ФСТЭК, разработанные для государственных информационных систем, представляют собой проверенную архитектурную схему, обеспечивающую надёжную защиту от разнообразных классов атак при условии их полного и правильного внедрения. Последовательное внедрение описанных мер позволяет значительно повысить устойчивость Linux-систем к современным угрозам информационной безопасности.

Список литературы

1. Методический документ от 25 декабря 2022 г. // ФСТЭК России : сайт. – URL: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-25-dekabrya-2022-g (дата обращения: 17.04.2026)

2. Карцан И.Н., Мордвинова А.Ю. Система управления информационными рисками. В сборнике: Вопросы контроля хозяйственной деятельности и финансового аудита, национальной безопасности, системного анализа и управления. Материалы VII Всероссийской научно-практической конференции. Москва, 2022. С. 141-146.
3. Мордвинова А.Ю., Нуриев С.А. Исследование уязвимостей и угроз безопасности стандарта IEEE 802.11. Современные инновации, системы и технологии. 2023. Т. 3. № 3. С. 117-131.
4. Мордвинова А.Ю., Жуков А.О., Аверьянов В.С., Карцан И.Н. Нейросетевая обработка информации в социокиберфизичеких системах. Информационные и телекоммуникационные технологии. 2021. № 52. С. 8-12.

References

1. Methodological document of December 25, 2022 // FSTEC of Russia: website. – URL: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/metodicheskij-dokument-ot-25-dekabrya-2022-g (accessed: April 17, 2026)
2. Kartsan I.N., Mordvinova A.Yu. Information Risk Management System. In the collection: Issues of Control over Economic Activities and Financial Audit, National Security, Systems Analysis and Management. Proceedings of the VII All-Russian Scientific and Practical Conference. Moscow, 2022. pp. 141-146.
3. Mordvinova A.Yu., Nuriyev S.A. Study of Vulnerabilities and Security Threats of the IEEE 802.11 Standard. Modern Innovations, Systems, and Technologies. 2023. Vol. 3. No. 3. pp. 117-131.
4. Mordvinova A.Yu., Zhukov A.O., Averyanov V.S., Kartsan I.N. Neural Network Information Processing in Sociocyber-Physical Systems. Information and Telecommunication Technologies. 2021. No. 52. pp. 8-12.

Харченко К.А., Батенков К.А. Разработка системы автоматической классификации видов разрешённого использования земельных участков// Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч.1 с. 105–111

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.852

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ВИДОВ РАЗРЕШЁННОГО ИСПОЛЬЗОВАНИЯ ЗЕМЕЛЬНЫХ УЧАСТКОВ

[1] Харченко К.А., Батенков К.А.
ФГБОУ ВО «МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ», Москва, Россия (119454, г. Москва, пр-т Вернадского, д. 78, стр. 4), e-mail: [1]kirill.kharchenko.02@yandex.ru

В данной работе разработан алгоритм автоматической классификации видов разрешённого использования земельных участков на основе методов обработки естественного языка и машинного обучения. Алгоритм включает предобработку текстовых данных, приведение их к векторным представлениям и применение гибридной иерархической модели для точного определения вида разрешённого использования. Приведено строгое математическое описание метода, включающее формализацию TF-IDF векторизации, логистической регрессии, косинусного сходства с прототипами классов и итоговой решающей функции. Результаты показали accuracy = 0,82 и Weighted F1 = 0,85.

Ключевые слова: ВРИ земельных участков, иерархическая классификация, гибридная математическая модель, TF-IDF, логистическая регрессия, косинусное сходство.

DEVELOPMENT OF AN AUTOMATIC CLASSIFICATION SYSTEM FOR PERMITTED USES OF LAND PLOTS

[1] Kharchenko K.A., Batenkov K.A.
MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY, Moscow, Russia (119454, Moscow, avenue Vernadsky, 78, b. 4), e-mail: [1]kirill.kharchenko.02@yandex.ru

This paper presents an algorithm for automatic classification of permitted uses of land plots based on NLP and machine learning. A rigorous mathematical framework is provided, including TF-IDF vectorization, logistic regression, cosine prototype similarity, and the hybrid decision function. Experimental results demonstrated accuracy = 0.82 and Weighted F1 = 0.85.

Keywords: Permitted land uses, hierarchical classification, hybrid model, TF-IDF, logistic regression, cosine similarity.

1. Формальная постановка задачи.

В работе решается задача автоматической классификации текстовых описаний видов разрешённого использования земельных участков. Пусть задано множество текстовых описаний:

$$X = \{x_1, x_2, \dots, x_n\}, \qquad (1)$$

где

X — множество текстовых описаний ВРИ;

n — число записей в датасете.

Требуется построить отображение:

$$f: X \to Y, \qquad (2)$$

где

Y — множество целевых классов ВРИ.

Задача рассматривается в иерархической постановке. Классификация выполняется в два этапа: сначала определяется укрупнённый (родительский) класс, затем уточняется подкласс внутри выбранной группы:

$$f(x) = f_2\big(x \mid f_1(x)\big), \qquad (3)$$

где

$f_1(x)$ — функция определения родительского класса;

$f_2(x \mid \hat{c})$ — функция уточнения подкласса при заданном родительском классе $\hat{c}$.

Родительский класс определяется как:

$$\hat{c}_p = argmax\_\{c \in C_p\} P(c \mid x), \qquad (4)$$

где

C_p — множество родительских классов (укрупнённых групп ВРИ);

$P(c \mid x)$ — вероятность принадлежности текста x к классу c.

Подкласс определяется аналогично, но в пределах выбранного родительского класса:

$$\hat{c}_s = argmax\_\{c \in C(\hat{c}_p)\} P(c \mid x, \hat{c}_p), \qquad (5)$$

где

$C(\hat{c}_p)$ — множество подклассов, принадлежащих родительскому классу $\hat{c}_p$.

2. Векторное представление текста (TF-IDF).

Для преобразования текстовых описаний в числовые векторы используется метод TF-IDF (Term Frequency — Inverse Document Frequency). Вес термина t в документе d определяется как:

$$w(t,d) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t), \qquad (6)$$

где частота термина с подлинейной нормализацией:

$$\mathrm{tf}(t,d) = 1 + \log_2 \mathrm{freq}(t,d), \text{ если } \mathrm{freq}(t,d) > 0, \qquad (7)$$

и обратная документная частота:

$$\mathrm{idf}(t) = \log\left(\frac{N}{\mathrm{df}(t)}\right), \qquad (8)$$

где

$freq(t, d)$ — число вхождений термина t в документ d;

N — общее число документов в коллекции;

$df(t)$ — число документов, содержащих термин t.

Каждый текст x_i представляется в виде вектора:

$$v_i = \big(w(t_1,x_i), w(t_2,x_i), \ldots, w(t_m,x_i)\big) \in \mathbb{R}^m, \qquad (9)$$

где m — размерность словаря (число уникальных термов, отобранных по критериям min_df и max_df). В работе используются словесные n-граммы порядка (1, 2), что позволяет учитывать как отдельные слова, так и устойчивые биграммы.

3. Классификатор на основе логистической регрессии.

В качестве базового классификатора на каждом уровне иерархии используется логистическая регрессия. Для задачи с K классами вероятность принадлежности текста x к классу c вычисляется с помощью функции softmax:

$$P(y = c \mid x) = \frac{\exp\left(w_c^T v + b_c\right)}{\sum_j \exp\left(w_j^T v + b_j\right)}, \qquad (10)$$

где

v — TF-IDF вектор текста x;

w^c — вектор весов для класса c;

b^c — смещение (bias) для класса c;

K — число классов на данном уровне иерархии.

Параметры модели оптимизируются минимизацией функции кросс-энтропийных потерь:

$$L = -\frac{1}{n}\sum_i \sum_c y_i^c \cdot \log P(y = c \mid x_i), \qquad (11)$$

где

y_i^c — индикатор принадлежности примера i к классу c (1 или 0);

n — число обучающих примеров.

4. Косинусное сходство с прототипом класса.

Для дополнительного уточнения классификации вычисляется сходство TF-IDF вектора текста с прототипом каждого класса. [1] Прототип класса c определяется как центроид TF-IDF векторов обучающих примеров данного класса:

$$p^c = \frac{1}{|D^c|}\sum_{x \in D^c} v(x), \qquad (12)$$

где

D^c — множество обучающих примеров класса c;

$v(x)$ — TF-IDF вектор текста x.

Сходство текста с прототипом определяется косинусной мерой:

$$S_{\text{proto}}(x, c) = \cos(v, p^c) = \frac{v \cdot p^c}{\|v\| \cdot \|p^c\|}, \qquad (13)$$

где

v — TF-IDF вектор классифицируемого текста;

p^c — прототип (центроид) класса c;

$\|\cdot\|$ — евклидова норма вектора.

Значение S_{proto} находится в диапазоне [0, 1] для неотрицательных TF-IDF векторов и характеризует семантическую близость текста к типичному представителю класса.[2]

5. Коэффициент пересечения ключевых слов.

Для каждого класса c формируется множество ключевых слов keywords(c), характерных для данной категории ВРИ. Коэффициент пересечения определяется как доля ключевых слов класса, присутствующих в тексте:

$$K_{\text{over}}(x, c) = \frac{|\text{keywords}(x) \cap \text{keywords}(c)|}{|\text{keywords}(c)|}, \qquad (14)$$

где

keywords(x) — множество слов текста x (после очистки и нормализации);

keywords(c) — заранее определённое множество ключевых слов класса c.

Данный признак позволяет учитывать предметно-ориентированную лексику, которая может быть недостаточно выражена в статистическом TF-IDF представлении.[3]

6. Предметно-ориентированный признак.

Предметно-ориентированный признак D(x) формируется на основе наличия кодов классификатора ВРИ в тексте. Если текст содержит числовой код вида «N.M» или «N.M.K», соответствующий записи в классификаторе, то:

$$D(x) = \begin{cases} 1, & \text{если } \mathrm{code}(x) \in \mathrm{Classifier} \\ 0, & \text{иначе} \end{cases}, \qquad (15)$$

где

code(x) — извлечённый из текста код классификатора;

Classifier — множество валидных кодов ВРИ.

При D(x) = 1 классификация может быть выполнена детерминированно по справочнику, минуя вероятностную модель.

7. Гибридная решающая функция.

Итоговое решение формируется гибридной функцией, комбинирующей вероятностные оценки классификаторов и дополнительные семантические признаки:

$$F(x) = \alpha \cdot P_p(c \mid x) \cdot P_s(c' \mid x, c) + \beta \cdot S_{\text{proto}}(x, c') + \gamma \cdot K_{\text{over}}(x, c') + \delta \cdot D(x), \qquad (16)$$

где

$P_p(c \mid x)$ — вероятность родительского класса c, определённая логистической регрессией (10);

$P_s(c' \mid x, c)$ — вероятность подкласса c' внутри родительского класса;

$S_{proto}(x, c')$ — косинусное сходство с прототипом подкласса (13);

$K_{over}(x, c')$ — коэффициент пересечения ключевых слов (14);

$D(x)$ — предметно-ориентированный признак (15);

$\alpha, \beta, \gamma, \delta$ — весовые коэффициенты, определяемые на валидационной выборке.

Итоговый класс определяется как:

$$\hat{c} = \mathrm{argmax}_{c'} F(x, c'), \qquad (17)$$

Если max F(x) < θ, где θ — порог уверенности, запись направляется на ручную проверку. Это обеспечивает контроль качества при промышленном использовании модели.

8. Метрики оценки качества.

Качество классификации оценивается по метрикам precision, recall и F1-score. Для класса c:

$$\mathrm{Precision}(c) = \frac{\mathrm{TP}^c}{\mathrm{TP}^c + \mathrm{FP}^c}, \qquad (18)$$

$$\mathrm{Recall}(c) = \frac{\mathrm{TP}^c}{\mathrm{TP}^c + \mathrm{FN}^c}, \qquad (19)$$

$$F1(c) = \frac{2 \cdot \mathrm{Precision}(c) \cdot \mathrm{Recall}(c)}{\mathrm{Precision}(c) + \mathrm{Recall}(c)}, \qquad (20)$$

где

TP^c — число верно предсказанных примеров класса c;

FP^c — число примеров, ошибочно отнесённых к классу c;

FN^c — число примеров класса c, пропущенных моделью.

Агрегированные метрики:

$$\text{Macro } F1 = (1/K) \Sigma^c F1(c), \qquad (21)$$

$$\text{Weighted } F1 = \Sigma^c (n^c / n) \cdot F1(c), \qquad (22)$$

где n^c — число примеров класса c, n — общее число примеров. Macro F1 придаёт одинаковый вес всем классам, Weighted F1 учитывает размер каждого класса.[4]

9. Результаты экспериментов.

Оценка качества проводилась на тестовой выборке из 1 188 записей. Полученные значения метрик по основным подклассам представлены в Таблице 1.

Таблица 1 - Метрики качества классификации по основным подклассам

Подкласс	**Precision**	**Recall**	**F1-score**	**Support**
2.1	0,92	0,89	0,91	66
2.3	0,92	0,85	0,88	26
2.6	0,64	0,60	0,62	45
3.1	0,99	0,91	0,95	185
3.2	0,82	0,82	0,82	34
3.3	0,62	0,69	0,66	36
3.4	0,94	0,85	0,89	72
3.5	0,94	0,85	0,89	132
3.6	0,78	0,80	0,79	45
3.8	0,77	0,55	0,64	42
3.9	0,76	0,90	0,83	29
4.1	0,74	0,73	0,74	67
4.4	0,95	0,86	0,90	251
4.6	0,82	0,80	0,81	50
4.9	0,98	0,75	0,85	65
5.1	0,77	0,74	0,76	23
6.9	0,64	0,90	0,75	20
Accuracy			0,82	1188
Macro avg	0,82	0,79	0,80	1188
Weighted avg	0,89	0,82	0,85	1188

Матрица ошибок классификации по основным подклассам представлена на Рисунке 1:

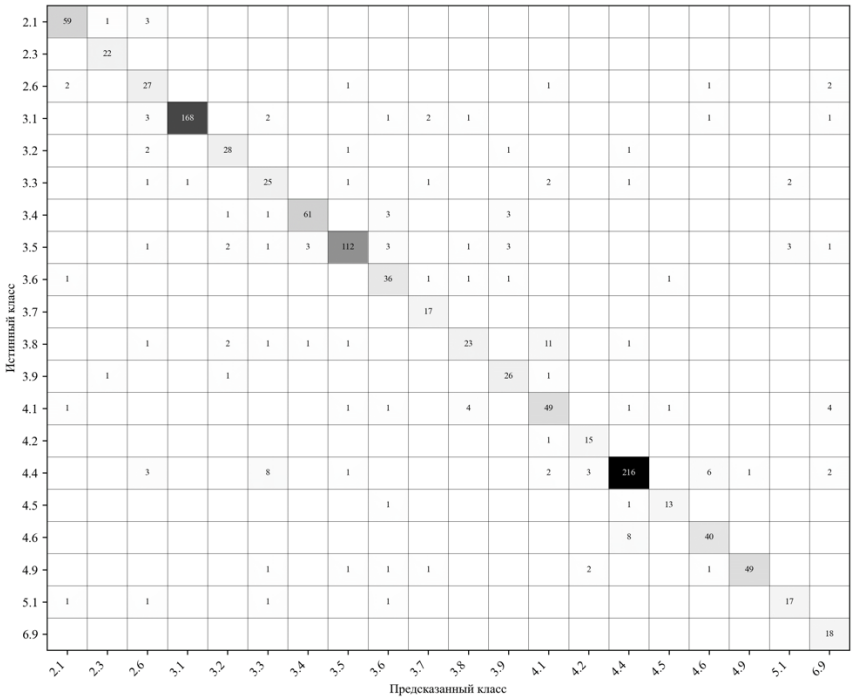

Рисунок 1 - Матрица ошибок классификации по основным подклассам

Анализируя полученные результаты, можно сказать, что разработанная модель обеспечивает достаточно устойчивое качество классификации текстовых описаний ВРИ. Средние значения accuracy = 0.82, Macro F1 = 0.80 и Weighted F1= 0.85, что говорит о приемлемой точности как в среднем по выборке, так и по отдельным подклассам. Самые высокие результаты показали подклассы 2.1, 3.1, 3.4, 3.5, 4.4 и 4.9, где значение F1-score находится в диапазоне от 0.85 до 0.95, что говорит об их достаточном объёме обучающих данных и более выраженными лексико-семантическими признаками классов. Более низкие значения получились у подклассов 2.6, 3.3, 3.8, 5.1 и 6.9, это связано с меньшим числом наблюдений и высокой семантической близостью к соседним классам. Анализ матрицы ошибок (Рисунок 1) показывает, что основная часть предсказаний сконцентрирована на главной диагонали, а возникающие ошибки носят объяснимый характер и возникают в основном между близкими по смыслу категориями.

Список литературы

1. Бурков А. Машинное обучение без лишних слов / А. Бурков. — СПб.: Питер, 2020. — 192 с.
2. Вьюгин В.В. Математические основы машинного обучения и прогнозирования / В.В. Вьюгин. — М.: Изд-во МЦНМО, 2021. — 399 с.
3. Марков С.С. Охота на электроовец. Большая книга искусственного интеллекта / С.С. Марков. — М.: ДМК Пресс, 2024. — 568 с.
4. Manning C.D., Raghavan P., Schütze H. Introduction to Information Retrieval. — Cambridge University Press, 2008. — 482 p.
5. Bishop C.M. Pattern Recognition and Machine Learning. — Springer, 2006. — 738 p.

References

1. Burkov, A. Machine Learning Without the Whispers / A. Burkov. St. Petersburg: Piter, 2020. p.192
2. V. V. Vyugin. Mathematical Foundations of Machine Learning and Forecasting / V. V. Vyugin. Moscow: MCNO Publishing House, 2021. p.399
3. Markov S. S. Electric Sheep Hunting. The Big Book of Artificial Intelligence / S. S. Markov. Moscow: DMK Press, 2024. p.568
4. Manning, C. D., Raghavan, P., Schütze, H. Introduction to Information Retrieval. Cambridge University Press, 2008. p.482
5. Bishop C. M. Pattern Recognition and Machine Learning. - Springer, 2006. - p. 738

Бетин В.В., Мордвинова А.Ю. Исследование категорий нарушителей и спектра их потенциальных возможностей// Международный журнал информационных технологий и энергоэффективности.– 2026. –Т. 11 № 6(68) Ч.1 с. 112–116

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.056

ИССЛЕДОВАНИЕ КАТЕГОРИЙ НАРУШИТЕЛЕЙ И СПЕКТРА ИХ ПОТЕНЦИАЛЬНЫХ ВОЗМОЖНОСТЕЙ

Бетин В.В., [1]**Мордвинова А.Ю**.
ФГАОУ ВО "СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ", Севастополь, Россия (299053, город Севастополь, Университетская ул., зд. 33), e-mail: [1] aumordvinova@mail.sevsu.ru

Статья анализирует актуальные угрозы кибербезопасности в условиях цифровизации, опираясь на статистику Роскомнадзора и классификацию ФСТЭК России. Рассматриваются типы нарушителей — внешние и внутренние, — а также их градация по уровням возможностей (от В.1 до В.4). Обсуждается степень опасности каждого класса с примерами атак и предлагаются targeted меры защиты: от базовых обновлений ПО для новичков до многоуровневых стратегий против государственных спецслужб. Материал подчеркивает важность адаптированных подходов для надежной охраны информационных систем.

Ключевые слова: Кибератаки, нарушители ИБ, классификация ФСТЭК, внешние угрозы, внутренние угрозы, уровни возможностей, защита информации, zero trust; цепочка поставок, национальная безопасность.

A STUDY OF INVULNER CATEGORIES AND THEIR RANGE OF POTENTIAL CAPABILITIES

Betin V.V., [1]Mordvinova A. Yu.
SEVASTOPOL STATE UNIVERSITY, Sevastopol, Russia (299053, Sevastopol, Universitetskaya st., 33), e-mail: [1] aumordvinova@mail.sevsu.ru

This article analyzes current cybersecurity threats in the context of digitalization, based on Roskomnadzor statistics and the classification of the Federal Service for Technical and Export Control of Russia. The article examines the types of intruders—external and internal—as well as their gradation by capability level (from B.1 to B.4). The danger level of each class is discussed, with attack examples and targeted security measures proposed, ranging from basic software updates for beginners to multi-layered strategies against government intelligence agencies. The material emphasizes the importance of tailored approaches for reliably protecting information systems.

Keywords: Cyberattacks, information security violators, FSTEC classification, external threats, internal threats, capability levels, information security, zero trust; supply chain, national security.

В эпоху цифровизации, нарушение информационной безопасности угрожает не только государственным учреждениям и частным компаниям, но и каждому пользователю глобальной сети. По данным Роскомнадзора в третьем квартале 2025 года произошло около 42 000 кибератак – 40% от общего числа компьютерных инцидентов за год. [1] Нарушители информационной безопасности – это основные исполнители большинства инцидентов в информационных инфраструктурах. Понимание их видов и способностей помогает выстраивать эффективную защиту информационной системы.

На данный момент нарушителей классифицирует ФСТЭК России, а именно в банке данных угроз ФСТЭК. Согласно данному документу, выделяют следующие виды нарушителей:

1. Внешние нарушители: специальные службы иностранных государств; террористические, экстремистские группировки; преступные группы (криминальные структуры); отдельные физические лица (хакеры); конкурирующие организации; лица, обеспечивающие поставку программных, программно-аппаратных средств, обеспечивающих систем; бывшие работники (пользователи).

2. Внутренние нарушители: разработчики программных, программно-аппаратных средств; поставщики вычислительных услуг, услуг связи; лица, привлекаемые для установки, настройки, испытаний, пусконаладочных и иных видов работ; лица, обеспечивающие функционирование систем и сетей или обеспечивающие системы оператора (администрация, охрана, уборщики и т. д.); авторизованные пользователи систем и сетей; системные администраторы и администраторы безопасности.

Также, согласно данному нормативному акту, нарушители классифицируются по категориям возможности, В.1 – В.4. Рассмотрим подробнее каждый из них.

Категория В.1: Нарушитель с базовыми возможностями.

Характеристики: использует только известные уязвимости, скрипты и общедоступные инструменты; обладает минимальными знаниями механизмов функционирования вредоносного ПО; имеет базовые компьютерные навыки на уровне пользователя; может осуществлять физические воздействия при наличии доступа к техническим средствам.

Типы нарушителей: физическое лицо (хакер); лица, обеспечивающие поставку программных, программно-аппаратных средств, обеспечивающих систем; лица, обеспечивающие функционирование систем и сетей или обеспечивающие системы оператора (администрация, охрана, уборщики и т.д.); авторизованные пользователи систем и сетей; бывшие работники (пользователи).

Особенности реализации угроз: способны реализовать только известные угрозы, направленные на задокументированные уязвимости; используют общедоступные инструменты и данные из сети «Интернет»; ограничены в возможности реализации атак ; не разрабатывают собственные вредоносные программы.

Категория В.2: Нарушитель с базовыми повышенными возможностями.

Характеристики: хорошо владеет общедоступными средствами реализации угроз; понимает принципы работы инструментов и может вносить изменения для повышения эффективности; обладает фреймворками и наборами специализированных инструментов; имеет навыки самостоятельного планирования сценариев угроз; знаком с защитными механизмами программного обеспечения; обладает практическими знаниями о функционировании систем и сетей.[2]

Типы нарушителей: преступные группы (два лица и более, действующие по единому плану); конкурирующие организации; поставщики вычислительных услуг, услуг связи; лица, привлекаемые для установки, настройки, испытаний, пусконаладочных и иных видов работ; системные администраторы и администраторы безопасности.

Особенности реализации угроз: способны реализовывать угрозы, направленные на недокументированные уязвимости; используют специально созданные инструменты, свободно распространяемые в Интернете; не имеют возможности атаковать физически изолированные сегменты систем; могут координировать действия в рамках группы.

Категория В.3: Нарушитель со средними возможностями.

Характеристики: имеет доступ к специализированным платным ресурсам (биржам уязвимостей); может приобретать дорогостоящие средства и инструменты для реализации угроз; способен самостоятельно разрабатывать инструменты для атак; имеет доступ к анализу встраиваемого ПО, системного и прикладного ПО; обладает навыками анализа программного кода для выявления уязвимостей; имеет глубокое понимание защитных механизмов; может действовать в составе группы специалистов.

Типы нарушителей: террористические, экстремистские группировки; разработчики программных, программно-аппаратных средств.[3]

Особенности реализации угроз: способны реализовывать угрозы на выявленные ими неизвестные уязвимости; используют самостоятельно разработанные инструменты; не имеют возможностей для атак на физически изолированные сегменты систем; обладают организационной структурой для координации атак.

Категория В.4: Нарушитель с высокими возможностями.

Характеристики: имеет доступ к исходному коду программного обеспечения для выявления уязвимостей «нулевого дня»; способен внедрять программные и программно-аппаратные закладки на этапах поставки; может создавать методы и средства реализации угроз с привлечением научных организаций; имеет возможность привлекать специалистов разных категорий возможностей; создает и применяет специальные технические средства для добывания информации через физические поля; способен долговременно и незаметно реализовывать угрозы; обладает исключительными знаниями о конкретных защитных механизмах атакуемых систем.

Типы нарушителей: специальные службы иностранных государств.

Особенности реализации угроз: имеют практически неограниченные возможности реализации угроз; используют недекларированные возможности и закладки, встроенные в компоненты систем; способны атаковать физически изолированные сегменты; обладают значительными ресурсами и государственной поддержкой; могут осуществлять долгосрочные операции по сбору информации.

Далее проведем анализ опасности данных классов нарушителей. Анализ представлен в порядке от самого опасного класса до менее опасного.

Класс нарушителя В.4 Тип нарушителя: Внешний Вид нарушителя: Специальные службы иностранных государств. Способности нарушителя: предоставляет угрозу национальной безопасности; обладает неограниченными возможностями; использует самые современные методы и средства. Пример атаки: атаки на критическую инфраструктуру России, шпионаж в военной и промышленной сферах.

Класс нарушителя В.3 Тип нарушителя: Внешний Вид нарушителя: Террористические и экстремистские группировки. [4] Способности нарушителя: дестабилизация общественной безопасности; использование информационных технологий для пропаганды и координации действий; атака на системы жизнеобеспечения. Пример атаки: атаки на системы управления транспортом, энергетическими объектами.

Класс нарушителя В.2 Тип нарушителя: Внутренний Вид нарушителя: Системные администраторы и администраторы безопасности. Способности нарушителя: максимальный уровень легального доступа; знание архитектуры и защитных механизмов сети; маскируют

свои действия под легитимные операции. Пример атаки: кража конфиденциальной информации, установка бэкдоров.

Класс нарушителя В.3 Тип нарушителя: Внутренний Вид нарушителя: Разработчики программных средств. Способности нарушителя: имеют доступ к исходным кодам приложений; могут внедрять скрытые функции и закладки; знакомы с внутренней архитектурой защищаемых систем. Пример атаки: внедрение троянов в банковское ПО, закладки в системное ПО.[5]

Класс нарушителя В.1 Тип нарушителя: Внешний Вид нарушителя: Хакеры-одиночки. Способности нарушителя: часто действуют из любопытства или для демонстрации своих возможностей; используют готовые инструменты и скрипты; ориентированы на массовые атаки на уязвимые системы. Пример атаки: взлом сайтов, DDoS-атаки на доступные ресурсы.

Класс нарушителя В.1 Тип нарушителя: Внутренний Вид нарушителя: Авторизованные пользователи и персонал. Способности нарушителя: могут случайно или намеренно нарушать политики безопасности; часто являются источником утечек из-за неосторожности; могут передавать учетные данные третьим лицам. Пример атаки: передача паролей, подключение несанкционированных устройств.

Теперь, когда были определены возможности и опасность нарушителей, рассмотрим рекомендации по защите от данных категорий нарушителей. Первоначально рассмотрим защиту от нарушителей категории В.1: Регулярное обновление ПО для закрытия известных уязвимостей. Базовое обучение пользователей основам ИБ. Контроль физического доступа к техническим помещениям. Простые, но эффективные парольные политики. Антивирусная защита и базовые средства обнаружения вторжений.

Защита от нарушителей категории В.2: Внедрение систем обнаружения аномального поведения. Разграничение прав доступа по принципу минимальных привилегий. Регулярный аудит действий администраторов и привилегированных пользователей. Контроль подключаемых устройств и внешних носителей. Внедрение систем предотвращения утечек данных (DLP)

Защита от нарушителей категории В.3: Глубокий анализ трафика и поведения в сети. Использование средств защиты от неизвестных угроз (песочницы, машинное обучение). Контроль цепочки поставок программного и аппаратного обеспечения. Внедрение принципа нулевого доверия (Zero Trust). Регулярное тестирование на проникновение силами независимых экспертов.

Защита от нарушителей категории В.4: Физическая изоляция критически важных сегментов. Использование отечественного оборудования и ПО с открытым кодом. Внедрение многоуровневых систем защиты с разными вендорами. Постоянный мониторинг на наличие скрытых закладок и аномальной активности. Взаимодействие со спецслужбами и участие в государственных программах защиты. Проведение регулярных проверок на электромагнитные утечки.

Таким образом, понимание классификации нарушителей информационной безопасности является ключевым элементом в построении эффективной системы защиты. Каждая категория нарушителей требует разработки специфических мер противодействия, соответствующих их возможностям и целям.

Список литературы

1. Россия под атакой. Под ударом хакеров телекоммуникации, ИТ и финансовый сектор // Cnews : сайт. – URL: https://www.cnews.ru/news/top/2025-11-05_s_iyulya_po_sentyabr_2025_goda.
2. БДУ ФСТЭК России - Нарушители // ФСТЭК России : сайт. – URL: https://bdu.fstec.ru/threat-section/potential
3. Карцан И.Н., Мордвинова А.Ю. Система управления информационными рисками. В сборнике: Вопросы контроля хозяйственной деятельности и финансового аудита, национальной безопасности, системного анализа и управления. Материалы VII Всероссийской научно-практической конференции. Москва, 2022. С. 141-146.
4. Мордвинова А.Ю., Нуриев С.А. Исследование уязвимостей и угроз безопасности стандарта IEEE 802.11. Современные инновации, системы и технологии. 2023. Т. 3. № 3. С. 117-131.
5. Мордвинова А.Ю., Жуков А.О., Аверьянов В.С., Карцан И.Н. Нейросетевая обработка информации в социокиберфизичеких системах. Информационные и телекоммуникационные технологии. 2021. № 52. С. 8-12.

References

1. Russia Under Attack. Telecommunications, IT, and the Financial Sector Under Hacker Attack // Cnews: website. – URL: https://www.cnews.ru/news/top/2025-11-05_s__iyulya_po_sentyabr_2025_goda.
2. BDU FSTEC of Russia - Violators // FSTEC of Russia: website. – URL: https://bdu.fstec.ru/threat-section/potential
3. Kartsan I.N., Mordvinova A.Yu. Information Risk Management System. In the collection: Issues of Control over Economic Activity and Financial Audit, National Security, Systems Analysis and Management. Proceedings of the VII All-Russian Scientific and Practical Conference. Moscow, 2022. pp. 141-146.
4. Mordvinova A.Yu., Nuriyev S.A. Study of Vulnerabilities and Security Threats of the IEEE 802.11 Standard. Modern Innovations, Systems, and Technologies. 2023. Vol. 3. No. 3. pp. 117-131.
5. Mordvinova A.Yu., Zhukov A.O., Averyanov V.S., Kartsan I.N. Neural Network Information Processing in Sociocyber-Physical Systems. Information and Telecommunication Technologies. 2021. No. 52. pp. 8-12.

Чуксин Б.Г., Труфанов А.И. Жанровая специфика в семантических связях сетевых моделей текстов // Международный журнал информационных технологий и энергоэффективности.– 2026. –Т. 11 № 6(68) Ч. 1 с. 117–121

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.912

ЖАНРОВАЯ СПЕЦИФИКА В СЕМАНТИЧЕСКИХ СВЯЗЯХ СЕТЕВЫХ МОДЕЛЕЙ ТЕКСТОВ

[1] **Чуксин Б.Г., Труфанов А.И.**
ФГБОУ ВО "ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ", Иркутск, Россия (664074, Иркутская область, город Иркутск, ул. Лермонтова, д. 83), e-mail: [1] bagisolo@mail.ru

В работе описывается методология построения графов совместной встречаемости на материале русскоязычного корпуса Taiga и представлены результаты пилотного эксперимента на подвыборке из 19 текстов трёх жанров — новостей, художественной прозы и нон-фикшн. По каждому тексту строился взвешенный граф с окном w=5; на полученных графах считались пять сетевых метрик. Проверка критерием Краскела-Уоллиса показала: три метрики из пяти — средняя длина кратчайшего пути (p=0.019), плотность (p=0.011) и модульность (p=0.012) — различаются между жанрами на статистически значимом уровне. Это даёт основания рассматривать графовый подход как рабочий инструмент жанровой классификации для русскоязычных текстов.

Ключевые слова: Графовые модели текста, граф совместной встречаемости, сетевые метрики, модульность, жанровая классификация, Taiga Corpus, NetworkX.

GENRE SPECIFICITY IN SEMANTIC CONNECTIONS OF NETWORK-BASED TEXT MODELS

[1]Chuksin B.G., Trufanov A.I.
IRKUTSK NATIONAL RESEARCH TECHNICAL UNIVERSITY, Irkutsk, Russia (664074, Irkutsk city Lermontova st., 83), e-mail: [1] bagisolo@mail.ru

This paper presents a methodology for constructing co-occurrence graphs from Russian-language texts of the Taiga corpus and reports the results of a pilot experiment on a sub-sample of 19 texts across three genres — news, fiction, and non-fiction. A weighted co-occurrence graph (window w=5) was built for each text, and five network metrics were computed. The Kruskal-Wallis test revealed statistically significant genre differences in three metrics: average shortest path length (p=0.019), graph density (p=0.011), and modularity (p=0.012). The findings support the viability of the graph-based approach for genre classification of Russian texts.

Keywords: graph-based text models, co-occurrence graph, network metrics, modularity, genre classification, Taiga Corpus, NetworkX.

Введение

Семантические связи в тексте — штука парадоксальная: любой читатель их чувствует, но формально описать непросто. Векторные подходы вроде TF-IDF, Word2Vec или BERT решают задачу в лоб — превращают текст в точку пространства. Удобно, но связи между конкретными словами при этом исчезают. Графовые модели устроены иначе: ребро между двумя узлами и есть тот самый объект изучения, а значит, к нему применим весь аппарат теории сетей [1].

В данной работе ставится задача разработать и проверить методологию построения co-occurrence графов на материале корпуса Taiga — одного из крупнейших русскоязычных

корпусов — и выяснить, меняется ли структура таких графов от жанра к жанру. Вопрос не абстрактный: графовые нейронные сети всё активнее применяются в задачах обработки текста [8], и то, каким именно окажется входной граф, напрямую влияет на качество модели. Пока выбор типа графового представления нередко делается на глаз — настоящая работа пытается дать для этого выбора хоть какое-то эмпирическое основание. На русскоязычном материале с полноценной статистической проверкой такой анализ проводится впервые.

Обзор литературы

Графовые модели текста как исследовательское направление сформировались в начале 2000-х годов. Одной из первых значимых работ, продемонстрировавших практическую применимость таких моделей, стала система TextRank [1], которая использовала граф совместной встречаемости для автоматического реферирования. Параллельно Ferrer i Cancho и Solé показали, что лексические сети обладают свойствами малого мира — низким диаметром и высоким коэффициентом кластеризации [2], что открыло новый угол зрения на организацию языка как системы.

Амансио и соавторы систематически исследовали связь между сложностью языка и топологическими свойствами текстовых графов [3], а Quispe и соавторы показали, что метрики co-occurrence сетей обладают высокой дискриминирующей способностью при классификации текстов [4]. Для семантических графов на основе дистрибутивного сходства принципиально важна работа Biemann и Riedl [5], где обосновывается идея связывания слов по близости их контекстуальных векторов.

Применительно к русскому языку наиболее близкими к данной работе являются исследования Зарубина Кирилла Александровича и Труфанова Андрея Ивановича по сетевым метрикам русской классической литературы [6] и собственная предыдущая работа автора, где аналогичный инструментарий применялся к интернет-текстам [7]. Настоящая статья продолжает эту линию исследований, распространяя её на разнородный жанровый корпус с применением строгой статистической проверки.

Методология

Корпус и выборка. Исследование проводится на материале корпуса Taiga [9] — многожанрового собрания русскоязычных текстов. В пилот вошли тексты трёх жанров — новости, художественная проза и нон-фикшн. Социальные сети изначально тоже планировались, однако из-за особенностей формата хранения этого подкорпуса удалось извлечь лишь один текст — этого очевидно недостаточно для какого-либо сравнения, поэтому жанр из анализа исключён. В итоге выборка получилась такой: 6 новостных текстов, 6 прозаических и 7 нон-фикшн, всего 19. Каждый текст обрезался до 500 токенов.

Предобработка. Предобработка намеренно сделана минималистичной. Регулярное выражение вытаскивает кириллические последовательности, всё приводится к нижнему регистру, затем вычищаются стоп-слова — их набралось 127 единиц — пунктуация и числа. Лемматизация в пилотном эксперименте не применялась — это ограничение будет снято в полной версии исследования с использованием библиотеки Natasha [10].

Построение графа. Граф строится следующим образом: каждая уникальная словоформа становится вершиной, а ребро между двумя вершинами появляется когда соответствующие

слова оказываются в пределах одного скользящего окна шириной пять токенов. Граф неориентированный, взвешенный — вес ребра отражает, сколько раз эта пара слов встретилась рядом в тексте. Граф построен средствами библиотеки NetworkX.

Извлечение метрик. На каждом графе считались пять метрик. Средняя степень вершины и коэффициент кластеризации характеризуют локальную связность; средняя длина кратчайшего пути считалась на наибольшей связной компоненте — для несвязных графов это стандартный обходной манёвр. Плотность даёт представление о насыщенности рёбрами в целом. Модульность вычислялась алгоритмом Лувена через пакет python-louvain и отражает, насколько чётко граф распадается на тематические кластеры.

Статистический анализ. Для проверки гипотезы о жанровых различиях применялся критерий Краскела-Уоллиса — непараметрический аналог однофакторного дисперсионного анализа, не требующий нормальности распределений. Уровень значимости $\alpha = 0.05$.

Результаты

В Таблице 1 представлены усреднённые значения пяти сетевых метрик для каждого жанра пилотной выборки.

Таблица 1 - Средние значения сетевых метрик co-occurrence графов по жанрам (w=5).

Жанр	Avg. Degree	Clustering	Avg. Path	Density	Modularity
Новости (n=6)	11.270	0.646	2.987	0.081	0.601
Проза (n=6)	11.410	0.646	3.317	0.049	0.644
Нон-фикшн (n=7)	11.098	0.647	4.227	0.028	0.718

В таблице 2 представлены результаты критерия Краскела-Уоллиса.

Таблица 2 - Результаты критерия Краскела-Уоллиса. * $p < 0.05$; н.з. — различия незначимы.

Метрика	H	p	Значимость
avg_degree	1.064	0.5873	н.з.
clustering	0.012	0.9942	н.з.
avg_path	7.906	0.0192	*
density	9.064	0.0108	*
modularity	8.854	0.0120	*

Три метрики из пяти оказались жанрово чувствительными — плотность (H=9.064, p=0.011), модульность (H=8.854, p=0.012) и средняя длина пути (H=7.906, p=0.019). Степень вершин и кластеризация практически не различаются между жанрами. Последнее само по себе любопытно: при окне w=5 локальная структура графа оказывается одинаковой вне зависимости от того, новость это, роман или научпоп.

Обсуждение

Нон-фикшн выделяется сильнее всего: модульность 0.718, плотность всего 0.028, средний путь — 4.227. За этими числами стоит вполне понятная вещь: такие тексты тематически монолитны, каждый следующий абзац вводит новые термины и редко возвращается к старым. Результат — разреженная сеть с хорошо отделёнными кластерами.

Новости — полная противоположность. Плотность 0.081, пути короткие (2.987), модульность низкая (0.601). Одни и те же слова — имена, топонимы, глаголы — мигрируют через весь текст, и граф получается густым и однородным без явных тематических зон.

Проза занимает промежуточное положение по всем метрикам. Художественный текст тематически разнообразен, но нарратив формирует связность через повторяющихся персонажей и сюжетные мотивы — отсюда умеренная модульность (0.644) и плотность (0.049).

Необходимо обозначить ограничения пилотного эксперимента. У пилота есть два очевидных слабых места. Первое — объём: 19 текстов дают статистику, но не дают уверенности в её устойчивости. Второе — отсутствие лемматизации: «идёт», «шёл» и «ходит» в нынешней версии — три разные вершины графа, что раздувает его и, скорее всего, занижает метрики связности. В полном эксперименте на 300 текстах оба момента будут закрыты: выборка вырастет, а Natasha обеспечит нормальную морфологическую нормализацию.

Заключение

В работе предложена и апробирована методология построения co-occurrence графов на материале русскоязычного корпуса Taiga. Пилотный эксперимент на 19 текстах трёх жанров дал конкретный результат: из пяти исследованных метрик три — плотность, модульность и средняя длина пути — жанрово специфичны, тогда как степень вершин и коэффициент кластеризации при окне w=5 оказались жанрово нейтральными. Последнее тоже результат: эти две метрики можно исключить из признакового пространства при последующей классификации.

Практический вывод прямой: плотность и модульность co-occurrence графа работают как жанровые маркеры для русскоязычных текстов. Дальнейшее исследование предполагает расширение выборки до 300 текстов, добавление лемматизации через библиотеку Natasha и сравнение co-occurrence графов с графами синтаксических зависимостей и семантического сходства. Переход от статического графа к динамическому — отслеживание того, как структура семантической сети складывается по мере последовательной обработки текста.

Список литературы

1. Mihalcea R., Tarau P. TextRank: Bringing Order into Texts // Proceedings of EMNLP. 2004. pp. 404–411.
2. Ferrer i Cancho R., Solé R.V. The small world of human language // Proceedings of the Royal Society of London. Series B. 2001. Vol. 268. pp. 2261–2265.
3. Amancio D.R., Aluisio S.M., Oliveira O.N., Costa L.F. Complex networks analysis of language complexity // EPL (Europhysics Letters). 2012. Vol. 100, No. 5. p. 58002.

4. Quispe L.V., Tohalino J.A., Amancio D.R. Using virtual edges to improve the discriminability of co-occurrence text networks // Physica A: Statistical Mechanics and its Applications. 2021. Vol. 562. pp. 125344.
5. Biemann C., Riedl M. Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity // Journal of Language Modelling. 2013. Vol. 1, No. 1. pp. 55–95.
6. Зарубин К.А., Труфанов А.И. Комплексные сети в русской классической литературе: атрибуция текста и сетевая модель // Обществознание и социальная психология. 2023. № 10 (58). С. 12–18.
7. Чуксин Б.Г., Труфанов А.И. Сетевая стилометрия русскоязычных интернет-текстов // Научные высказывания. 2024. № 12 (59). С. 24–28.
8. Yao L., Mao C., Luo Y. Graph Convolutional Networks for Text Classification // Proceedings of the AAAI Conference on Artificial Intelligence. 2019. Vol. 33, No. 01. P. 7370–7377.
9. Shavrina T., Shapovalova O. To the methodology of corpus construction for machine learning: «Taiga» syntax tree corpus and parser // Trudy mezhdunarodnoj konferencii «Korpusnaja lingvistika — 2017». Saint-Petersburg: Izd-vo SPbGU, 2017.
10. Korobov Y. Natasha: NLP library for Russian [Elektronnyj resurs]. — URL: https://github.com/natasha/natasha (data obrashhenija: 15.04.2026).

References

1. Mihalcea R., Tarau P. TextRank: Bringing Order into Texts, Proceedings of EMNLP, 2004, pp. 404–411.
2. Ferrer i Cancho R., Solé R.V. The small world of human language, Proceedings of the Royal Society of London. Series B, 2001, vol. 268, pp. 2261–2265.
3. Amancio D.R., Aluisio S.M., Oliveira O.N., Costa L.F. Complex networks analysis of language complexity, EPL (Europhysics Letters), 2012, vol. 100, no. 5, p. 58002.
4. Quispe L.V., Tohalino J.A., Amancio D.R. Using virtual edges to improve the discriminability of co-occurrence text networks, Physica A: Statistical Mechanics and its Applications, 2021, vol. 562, p. 125344.
5. Biemann C., Riedl M. Text: Now in 2D! A Framework for Lexical Expansion with Contextual Similarity, Journal of Language Modelling, 2013, vol. 1, no. 1, pp. 55–95.
6. Zarubin K.A., Trufanov A.I. Kompleksnye seti v russkoj klassicheskoj literature: atributsiya teksta i setevaya model [Complex networks in Russian classical literature: text attribution and network model], Obshchestvoznanie i sotsial'naya psikhologiya, 2023, no. 10 (58), pp. 12–18.
7. Chuksin B.G., Trufanov A.I. Setevaya stilometriya russkoyazychnykh internet-tekstov [Network stylometry of Russian-language internet texts], Nauchnye vyskazyvaniya, 2024, no. 12 (59), pp. 24–28.
8. Yao L., Mao C., Luo Y. Graph Convolutional Networks for Text Classification, Proceedings of the AAAI Conference on Artificial Intelligence, 2019, vol. 33, no. 01, pp. 7370–7377.
9. Shavrina T., Shapovalova O. To the methodology of corpus construction for machine learning: «Taiga» syntax tree corpus and parser, Proc. of International Conference «Corpus Linguistics — 2017», Saint-Petersburg, 2017.
10. Korobov Y. Natasha: NLP library for Russian. Available at: https://github.com/natasha/natasha (accessed 15.04.2026).

Соломатин К.В. Контурная модель размещения компонентов распределённых информационных систем в мультикластерной среде KUBERNETES// Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч.1 с. 122–131

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.75

КОНТУРНАЯ МОДЕЛЬ РАЗМЕЩЕНИЯ КОМПОНЕНТОВ РАСПРЕДЕЛЁННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ В МУЛЬТИКЛАСТЕРНОЙ СРЕДЕ KUBERNETES

Соломатин К.В.
ФГБОУ ВО «МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ», Москва, Россия (119454, г. Москва, пр-т Вернадского, д. 78, стр. 4), e-mail: kbc.off@list.ru

Предложена контурная модель размещения компонентов распределённых информационных систем в мультикластерной среде Kubernetes. Модель задаёт функциональное разделение на клиентский, серверный, хранилищный и управляющий контуры, фиксирует правила межконтурного взаимодействия и ориентирована на гетерогенные вычислительные среды. Подход апробирован на архитектуре рекомендательной системы подбора рекламных площадок.

Ключевые слова: Контурная модель, Kubernetes, мультикластерная архитектура, распределённые системы, рекомендательная система, WireGuard, Istio.

CONTOUR MODEL FOR COMPONENT PLACEMENT IN DISTRIBUTED INFORMATION SYSTEMS WITHIN A MULTI-CLUSTER KUBERNETES ENVIRONMENT

Solomatin K.V.
MIREA - RUSSIAN TECHNOLOGICAL UNIVERSITY, Moscow, Russia (119454, Moscow, avenue Vernadsky, 78, b. 4), e-mail: kbc.off@list.ru

A contour model for component placement in distributed information systems within a multi-cluster Kubernetes environment is proposed. The model introduces functional segregation into client, server, storage, and management contours, defines inter-contour interaction rules, and targets heterogeneous computing environments. The approach is illustrated using a recommendation system for advertising platform selection.

Keywords: Contour model, Kubernetes, multi-cluster architecture, distributed systems, recommendation system, WireGuard, Istio.

Введение

Развитие контейнерных технологий и микросервисной архитектуры сделало мультикластерные конфигурации одним из практических способов развёртывания распределённых информационных систем. Это особенно характерно для прикладных решений, сочетающих пользовательские интерфейсы, вычислительные сервисы и ресурсоёмкие подсистемы хранения и обработки данных [1, 2, 3].

Согласно отчётам Cloud Native Computing Foundation, свыше 60 % организаций, применяющих Kubernetes, используют более одного кластера в производственной среде. Основными причинами являются территориальное распределение инфраструктуры, изоляция зон ответственности, соблюдение регуляторных требований к локализации данных, а также

стремление ограничить область поражения при отказе отдельного управляющего компонента [4, 5].

При размещении всех компонентов в одном Kubernetes-кластере усложняется изоляция нагрузок, возрастает зависимость от общего состояния управляющей плоскости и снижается гибкость использования гетерогенной инфраструктуры, включающей VPS и выделенные физические узлы [4, 5]. Инструменты KubeFed, Rancher и Cluster API решают задачи оркестрации и сопровождения парка кластеров, однако не формализуют принципы функционального распределения компонентов между ними [6, 7, 8].

В исследованиях по микросервисной архитектуре основное внимание обычно уделяется декомпозиции системы, гранулярности сервисов и управлению API-взаимодействием [9]. Вопрос группировки компонентов по функциональным ролям на уровне нескольких кластеров остаётся менее формализованным. Существующие рекомендации ограничиваются практиками «один кластер на среду» или «один кластер на команду», не предлагая системного подхода к распределению компонентов на основании их функциональных характеристик.

Цель исследования

Цель статьи заключается в формализации контурной модели размещения компонентов распределённой системы в мультикластерной среде Kubernetes и демонстрации её применения к рекомендательному конвейеру обработки данных. Для достижения поставленной цели решаются следующие задачи: анализ существующих средств мультикластерного управления и выявление архитектурного пробела; формализация модели контурного размещения с определением структуры, принципов и метрики качества; проверка применимости модели на примере распределённой рекомендательной системы подбора рекламных площадок.

Материал и методы исследования

Среди распространённых средств работы с мультикластерной средой можно выделить KubeFed, Cluster API и Rancher. Их назначение связано с синхронизацией ресурсов, централизованным администрированием и управлением жизненным циклом кластеров, но не с описанием архитектурных принципов размещения прикладных компонентов [6, 7, 8].
KubeFed (Federation v2) обеспечивает репликацию объектов Kubernetes между кластерами на декларативной основе. Инструмент позволяет задавать политики размещения ресурсов, однако оперирует инфраструктурными абстракциями (Deployment, ConfigMap, Service) и не содержит механизма функциональной классификации компонентов. Cluster API формализует жизненный цикл самих кластеров: их создание, обновление и удаление. Rancher предоставляет единую консоль управления произвольным набором кластеров и поддерживает импорт существующих сред, включая K3s и RKE2, но не предлагает формальной модели размещения компонентов [6, 7, 8].

Тем самым перечисленные решения следует рассматривать как инструментальный слой реализации. Предлагаемая контурная модель относится к архитектурному уровню и задаёт критерии распределения компонентов по кластерам. Таблица 1 показывает степень покрытия соответствующих требований существующими средствами.

Таблица 1 - Покрытие требований контурной модели существующими инструментами

Требование	**Federation v2**	**Cluster API**	**Rancher**	**Контурная модель**
Поддержка гетерогенных сред	Ограниченная: ручная адаптация [6]	Частичная: зависит от провайдера	Да: импорт произвольных кластеров	K3s на VPS и физических серверах [10]
Наложенная сетевая связность	Не предусмотрена	Не предусмотрена	Частичная: через сторонние плагины	WireGuard full-mesh [11]
Межкластерное обнаружение сервисов	Федеративная DNS	Отсутствует	Через сторонний service mesh	Istio multicluster east-west [12]
Архитектурное разделение по ролям	Не определено	Не определено	Возможно через проекты и labels	Формализовано: сегрегация контуров
Стратегия деградации при отказе	Не определена	Не определена	Не определена	Постепенная деградация

Из Таблицы следует, что существующие средства покрывают отдельные аспекты мультикластерной эксплуатации, но не задают формальной схемы функционального разделения кластеров и сценариев деградации. Именно этот уровень абстракции и фиксируется контурной моделью.

Контурная модель размещения компонентов — архитектурный паттерн, при котором каждый Kubernetes-кластер мультикластерной среды выделяется для выполнения определённой функциональной роли (контура). Компоненты информационной системы распределяются по контурам на основании однородности требований к вычислительным ресурсам, безопасности, доступности и характеру нагрузки. Количество контуров определяется особенностями конкретной системы; ниже приведена обобщённая формализация модели.

Модель определяется кортежем $\langle S, E, C, R, I \rangle$, где: $S = \{s_1, s_2, \ldots, s_m\}$ — множество компонентов (сервисов) системы; $E \subseteq S \times S$ — множество зависимостей между компонентами (вызовы API, потоки данных), определяющее граф зависимостей $G = (S, E)$; $C = \{c_1, c_2, \ldots, c_n\}$ — множество функциональных контуров; $R: S \to C$ — функция размещения, отображающая каждый компонент на контур; $I \subseteq C \times C$ — множество разрешённых межконтурных взаимодействий.

Для оценки качества размещения вводятся следующие характеристики. Пусть $\text{intra}(c) = |\{(s_1, s_2) \in E : R(s_1) = R(s_2) = c\}|$ — число рёбер внутри контура c, а $\text{inter} = |\{(s_1, s_2) \in E : R(s_1) \neq R(s_2)\}|$ — суммарное число межконтурных рёбер. Принцип сегрегации контуров формулируется как задача максимизации отношения суммарной внутриконтурной связности к межконтурной: $Q(R) = \Sigma\, \text{intra}(c_i) / (\text{inter} + 1) \to \max$ при ограничении на однородность

ресурсного профиля внутри каждого контура. Данный критерий сочетает два требования: минимизацию межкластерного трафика (латентность наложенной сети [11]) и группировку компонентов с близкими ресурсными характеристиками.

Метрика Q(R) позволяет сравнивать различные варианты размещения при проектировании системы. Более высокое значение Q указывает на то, что большинство зависимостей обслуживается внутрикластерным трафиком с минимальной латентностью, тогда как межконтурные взаимодействия сведены к явно определённым точкам интеграции. На практике абсолютное значение Q менее информативно, чем относительное сравнение альтернативных вариантов размещения для одной и той же системы.

В рамках модели предполагается, что каждый компонент принадлежит ровно одному контуру, а компоненты одного контура обладают близким ресурсным профилем. Если пара контуров не входит в множество разрешённых взаимодействий, соответствующие зависимости между компонентами исключаются на уровне архитектурного графа. Допустимые межконтурные обращения реализуются через сервисную сетку с обязательным mTLS [12, 15]. При этом для каждого разрешённого взаимодействия должна быть определена стратегия деградации, исключающая каскадный отказ системы [4].

Для систем с выраженным разделением на интерфейсный, вычислительный, хранилищный и эксплуатационный слои принцип сегрегации естественным образом приводит к выделению четырёх функциональных контуров [3]. Каждый контур характеризуется преобладающим типом нагрузки: stateless-обработка HTTP-запросов для клиентского контура, CPU-интенсивные вычисления для серверного, IO-интенсивные операции для хранилищного и низкоприоритетная сборка метрик для управляющего.

В прикладном сценарии выделяются четыре функциональных контура. Клиентский контур обслуживает веб-интерфейсы, административную панель и контроллер входящего трафика; для него характерны stateless-нагрузка и повышенные требования к доступности. Серверный контур включает прикладные сервисы, в том числе парсеры внешних источников, ETL-узлы, API рекомендаций и сервис аутентификации; этот контур ориентирован на вычислительно-интенсивную нагрузку. Контур хранения объединяет персистентные компоненты, включая реляционную СУБД, кэш, брокер сообщений, аналитический и поисковый слои, и характеризуется интенсивной дисковой нагрузкой и stateful-природой компонентов [3, 4]. Контур управления содержит средства наблюдаемости, журналирования, трассировки и управления секретами, обеспечивая сквозной мониторинг остальных частей системы [13].

Множество разрешённых взаимодействий: $I = \{(c_1, c_2), (c_2, c_1), (c_2, c_3), (c_3, c_2), (c_4, c_1), (c_4, c_2), (c_4, c_3)\}$. Клиентский контур не обращается к хранилищу напрямую, а взаимодействует с ним исключительно через серверный контур. Управляющий контур имеет доступ на чтение ко всем остальным контурам для сбора метрик и журналов, но не участвует в прикладном потоке обработки данных.

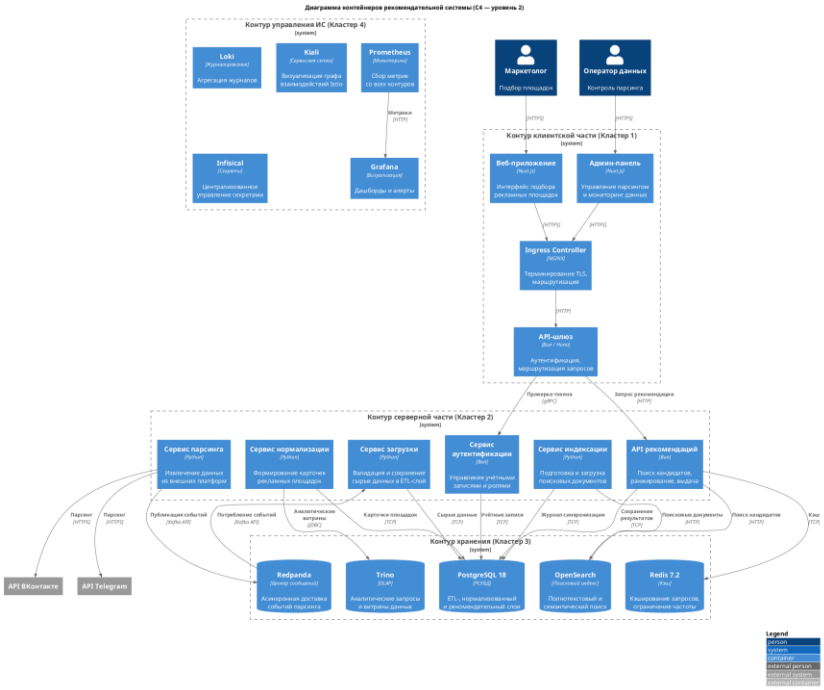

Рисунок 1 - Диаграмма контейнеров (C4, уровень 2): четыре функциональных контура системы [14]

Кластеры объединены наложенной VPN-сетью WireGuard с полносвязной топологией и единым адресным пространством [11]. WireGuard выбран в качестве транспортного уровня благодаря минимальной кодовой базе (менее 4 000 строк кода в ядре Linux), высокой пропускной способности и низкой латентности по сравнению с IPsec и OpenVPN. Каждый узел кластера получает уникальный IP-адрес из единого блока /16, что обеспечивает прозрачную маршрутизацию между подами разных кластеров.

Поверх сетевого уровня развёрнута сервисная сетка Istio в режиме multicluster [12], обеспечивающая прозрачное обнаружение сервисов между контурами через east-west шлюзы, обязательное mTLS-шифрование межкластерного и внутрикластерного трафика, авторизационные политики для ограничения набора допустимых взаимодействий и механизм автоматического прерывателя цепи при превышении порога ошибок. Конфигурация Istio включает централизованный корневой центр сертификации, от которого каждый кластер получает промежуточный сертификат, что позволяет поддерживать единую цепочку доверия при независимости управляющих плоскостей отдельных кластеров [15].

Контурная модель применена к распределённой рекомендательной системе подбора рекламных площадок. Функциональная цепочка системы включает сбор сведений из внешних платформ, асинхронную доставку событий через брокер сообщений, нормализацию данных, индексирование поисковых документов и выдачу рекомендаций по текстовому запросу пользователя [1, 3]. Вычислительная среда системы является гетерогенной и включает виртуальные серверы различных провайдеров, а также выделенный физический сервер для контура хранения [4].

Инфраструктура развёрнута на четырёх кластерах K3s. Клиентский кластер размещён на VPS с 2 vCPU и 4 ГБ ОЗУ и обслуживает входящий трафик через Ingress-контроллер Traefik. Серверный кластер использует VPS с 4 vCPU и 8 ГБ ОЗУ, обеспечивая достаточные вычислительные ресурсы для парсеров и ETL-конвейера. Контур хранения размещён на

выделенном физическом сервере с 32 ГБ ОЗУ и NVMe-накопителями, что критично для дисковых нагрузок PostgreSQL, OpenSearch и Redpanda. Управляющий кластер развёрнут на отдельном VPS с минимальными ресурсами (2 vCPU, 2 ГБ ОЗУ), поскольку инструменты наблюдаемости не требуют интенсивных вычислений [10].

В распределении компонентов по контурам клиентская часть включает веб-интерфейс, административную панель, API-шлюз и контроллер входящего трафика. Серверный контур объединяет парсер внешних платформ, сервис загрузки данных, сервис нормализации, индексатор поисковых документов, API рекомендаций и сервис аутентификации. Контур хранения включает PostgreSQL, Redis, Redpanda, Trino и OpenSearch [3, 4]. В управляющем контуре размещаются Prometheus, Grafana, Loki, Kiali и Infisical [13]. Такое распределение согласует архитектуру системы с различиями в типах нагрузки и требованиях к доступности отдельных подсистем.

Значение метрики Q(R) для данного размещения составляет 4,2 при 17 внутриконтурных и 8 межконтурных зависимостях. Для сравнения, размещение всех сервисов в двух кластерах (клиент + остальные) даёт Q = 1,8, а объединение всех компонентов в одном кластере — Q = 0 (отсутствие межконтурных рёбер делает метрику неприменимой). Таким образом, четырёхконтурная схема более чем вдвое превышает двухкластерную альтернативу по показателю функциональной сегрегации.

Одним из практических требований к модели выступает отсутствие каскадного отказа при нарушении межкластерной связности. Поэтому для каждого контура задаётся режим ограниченного функционирования, сохраняющий работоспособность оставшихся частей системы [4].

Если недоступен контур хранения, серверная часть переводит операции записи в локальную очередь повторной обработки, а пользовательский контур ограничивает часть функций до восстановления связности. При отказе серверного контура клиентская часть может продолжать обслуживание кэшированных результатов, тогда как сбор новых данных приостанавливается. Недоступность управляющего контура не должна останавливать прикладной конвейер, а приводит только к временной потере телеметрии. При отказе клиентского контура сбор и нормализация данных сохраняются, однако пользовательский доступ становится временно недоступным.

Механизм автоматического прерывателя цепи (circuit breaker), реализованный средствами Istio, при превышении порога ошибок временно прекращает направление трафика на неисправный эндпоинт и возобновляет его после восстановления [12]. В проведённом тестовом сценарии отключение сетевого канала между серверным и хранилищным контурами приводило к срабатыванию прерывателя в течение 5 секунд, после чего серверный контур переходил в режим локального кэширования запросов. Восстановление связности автоматически возобновляло нормальный поток обработки без вмешательства оператора.

Результаты апробации показывают, что контурная модель применима к системам с выраженной гетерогенностью вычислительных ресурсов и различием требований к доступности отдельных подсистем. Разделение на функциональные контуры позволяет независимо масштабировать каждую группу компонентов и проводить обновления без воздействия на остальные части системы. Вместе с тем модель вводит дополнительные

накладные расходы на межкластерный трафик и усложняет начальную настройку сетевой связности.

Выводы

В статье предложена контурная модель размещения компонентов распределённых информационных систем в мультикластерной среде Kubernetes. Модель формализует функциональное разделение кластеров по прикладным ролям и задаёт набор правил межконтурного взаимодействия.

Сопоставление с существующими средствами мультикластерного управления показало, что рассматриваемый подход закрывает пробел между инструментами эксплуатации кластеров и архитектурными принципами распределения прикладных компонентов.

Применение модели к рекомендательной системе показало, что функциональная сегрегация упрощает размещение компонентов в гетерогенной среде и позволяет локализовать последствия отказов. Предложенная метрика Q(R) даёт количественную основу для сравнения альтернативных вариантов размещения.

Ограничение работы состоит в проектном характере верификации модели. Дальнейшее развитие связано с экспериментальной оценкой производительности, анализом накладных расходов межконтурного взаимодействия и уточнением метрик эффективности контурного разделения.

Список литературы

1. Михайлов А.Н. Разработка рекомендательных систем: методы и алгоритмы для электронной коммерции // Вестник науки. 2024. URL: https://cyberleninka.ru/article/n/razrabotka-rekomendatelnyh-sistem-metody-i-algoritmy-dlya-elektronnoy-kommertsii (дата обращения: 10.04.2026).
2. Анализ современных подходов в проектировании рекомендательных систем / К.А. Разуваев [и др.] // Международный журнал прикладных наук и технологий «Integral». 2021. URL: https://cyberleninka.ru/article/n/analiz-sovremennyh-podhodov-v-proektirovanii-rekomendatelnyh-sistem (дата обращения: 10.04.2026).
3. Артамонов Ю.С., Востокин С.В. Разработка распределенных приложений сбора и анализа данных на базе микросервисной архитектуры // Известия Самарского научного центра Российской академии наук. 2016. URL: https://cyberleninka.ru/article/n/razrabotka-raspredelennyh-prilozheniy-sbora-i-analiza-dannyh-na-baze-mikroservisnoy-arhitektury (дата обращения: 10.04.2026).
4. Долматов Р.А., Сараджишвили С.Э. Оптимизация масштабируемости и отказоустойчивости микросервисов с применением Kubernetes // Системный анализ в проектировании и управлении. 2024. URL: https://cyberleninka.ru/article/n/optimizatsiya-masshtabiruemosti-i-otkazoustoychivosti-mikroservisov-s-primeneniem-kubernetes (дата обращения: 10.04.2026).
5. Моделирование масштабируемых микросервисных систем в условиях контейнерной виртуализации с использованием Kubernetes и Service Mesh (на примере Istio) / А.Э. Баженов [и др.] // Программные системы и вычислительные методы. 2025. URL: https://cyberleninka.ru/article/n/modelirovanie-masshtabiruemyh-mikroservisnyh-sistem-v-

usloviyah-konteynernoy-virtualizatsii-s-ispolzovaniem-kubernetes-i-service (дата обращения: 10.04.2026).
6. KubeFed User Guide [Электронный ресурс] // Kubernetes SIG Multicluster. URL: https://github.com/kubernetes-retired/kubefed/tree/master/docs/userguide (дата обращения: 10.04.2026).
7. Rancher Documentation. Architecture and Fleet [Электронный ресурс]. URL: https://ranchermanager.docs.rancher.com/ (дата обращения: 10.04.2026).
8. Cluster API Book. Introduction [Электронный ресурс]. URL: https://cluster-api.sigs.k8s.io/ (дата обращения: 10.04.2026).
9. Чикалева Ю.С. Анализ гранулярности микросервисов: эффективность архитектурных подходов // Программные системы и вычислительные методы. 2025. URL: https://cyberleninka.ru/article/n/analiz-granulyarnosti-mikroservisov-effektivnost-arhitekturnyh-podhodov (дата обращения: 10.04.2026).
10. K3s Documentation. Lightweight Kubernetes [Электронный ресурс]. URL: https://docs.k3s.io/ (дата обращения: 10.04.2026).
11. Donenfeld J.A. WireGuard: Next Generation Kernel Network Tunnel // Proceedings of the Network and Distributed System Security Symposium. 2017. URL: https://www.ndss-symposium.org/wp-content/uploads/2017/09/ndss2017_04A-3_Donenfeld_paper.pdf (дата обращения: 10.04.2026).
12. Install Multicluster [Электронный ресурс] // Istio Documentation. URL: https://istio.io/latest/docs/setup/install/multicluster/ (дата обращения: 10.04.2026).
13. Мясников И.В. Мониторинг компонентов Istio для обеспечения надежности и наблюдаемости Service Mesh // Вестник науки. 2025. URL: https://cyberleninka.ru/article/n/monitoring-komponentov-istio-dlya-obespecheniya-nadezhnosti-i-nablyudaemosti-service-mesh (дата обращения: 10.04.2026).
14. Корниенко Д.В., Никулин А.В. Визуализация архитектур информационных систем, основанных на микросервисах, с использованием данных OpenTelemetry // Computational nanotechnology. 2024. URL: https://cyberleninka.ru/article/n/vizualizatsiya-arhitektur-informatsionnyh-sistem-osnovannyh-na-mikroservisah-s-ispolzovaniem-dannyh-opentelemetry (дата обращения: 10.04.2026).
15. Ковтун Д.П., Лапонина О.Р. Использование управления доступом на основе атрибутов и mTLS в микросервисной архитектуре // International Journal of Open Information Technologies. 2025. URL: https://cyberleninka.ru/article/n/ispolzovanie-upravleniya-dostupom-na-osnove-atributov-i-mtls-v-mikroservisnoy-arhitekture (дата обращения: 10.04.2026).

References

1. Mikhailov, A.N. "Development of Recommender Systems: Methods and Algorithms for E-Commerce" // Science Bulletin. 2024. URL: https://cyberleninka.ru/article/n/razrabotka-rekomendatelnyh-sistem-metody-i-algoritmy-dlya-elektronnoy-kommertsii (Accessed: 10.04.2026).
2. Analysis of Modern Approaches to Designing Recommender Systems / K.A. Razuvaev et al. // International Journal of Applied Sciences and Technologies "Integral". 2021. URL:

https://cyberleninka.ru/article/n/analiz-sovremennyh-podhodov-v-proektirovanii-rekomendatelnyh-sistem (Accessed: 10.04.2026).

3. Artamonov Yu.S., Vostokin S.V. Development of distributed data collection and analysis applications based on microservice architecture // Bulletin of the Samara Scientific Center of the Russian Academy of Sciences. 2016. URL: https://cyberleninka.ru/article/n/razrabotka-raspredelennyh-prilozheniy-sbora-i-analiza-dannyh-na-baze-mikroservisnoy-arhitektury (date of access: 10.04.2026).
4. Dolmatov R.A., Sarajishvili S.E. Optimization of scalability and fault tolerance of microservices using Kubernetes // Systems analysis in design and management. 2024. URL: https://cyberleninka.ru/article/n/optimizatsiya-masshtabiruemosti-i-otkazoustoychivosti-mikroservisov-s-primeneniem-kubernetes (date of access: 10.04.2026).
5. Modeling scalable microservice systems in the context of container virtualization using Kubernetes and Service Mesh (using Istio as an example) / A.E. Bazhenov [et al.] // Software systems and computational methods. 2025. URL: https://cyberleninka.ru/article/n/modelirovanie-masshtabiruemyh-mikroservisnyh-sistem-v-usloviyah-konteynernoy-virtualizatsii-s-ispolzovaniem-kubernetes-i-service (date of access: 10.04.2026).
6. KubeFed User Guide [Electronic resource] // Kubernetes SIG Multicluster. URL: https://github.com/kubernetes-retired/kubefed/tree/master/docs/userguide (date of access: 10.04.2026).
7. Rancher Documentation. Architecture and Fleet [Electronic resource]. URL: https://ranchermanager.docs.rancher.com/ (date of access: 10.04.2026).
8. Cluster API Book. Introduction [Electronic resource]. URL: https://cluster-api.sigs.k8s.io/ (date of access: 10.04.2026).
9. Chikaleva Yu.S. Analysis of Microservice Granularity: Efficiency of Architectural Approaches // Software Systems and Computational Methods. 2025. URL: https://cyberleninka.ru/article/n/analiz-granulyarnosti-mikroservisov-effektivnost-arhitekturnyh-podhodov (accessed: 10.04.2026).
10. K3s Documentation. Lightweight Kubernetes [Electronic resource]. URL: https://docs.k3s.io/ (accessed: 10.04.2026).
11. Donenfeld J.A. WireGuard: Next Generation Kernel Network Tunnel // Proceedings of the Network and Distributed System Security Symposium. 2017. URL: https://www.ndss-symposium.org/wp-content/uploads/2017/09/ndss2017_04A-3_Donenfeld_paper.pdf (date accessed: 10.04.2026).
12. Install Multicluster [Electronic resource] // Istio Documentation. URL: https://istio.io/latest/docs/setup/install/multicluster/ (date accessed: 10.04.2026).
13. Myasnikov I.V. Monitoring Istio Components to Ensure Service Mesh Reliability and Observability // Science Herald. 2025. URL: https://cyberleninka.ru/article/n/monitoring-komponentov-istio-dlya-obespecheniya-nadezhnosti-i-nablyudaemosti-service-mesh (date of access: 10.04.2026).
14. Kornienko D.V., Nikulin A.V. Visualization of architectures of information systems based on microservices using OpenTelemetry data // Computational nanotechnology. 2024. URL: https://cyberleninka.ru/article/n/vizualizatsiya-arhitektur-informatsionnyh-sistem-

osnovannyh-na-mikroservisah-s-ispolzovaniem-dannyh-opentelemetry (date of access: 10.04.2026).

15. Kovtun D.P., Laponina O.R. Using attribute-based access control and mTLS in microservice architecture // International Journal of Open Information Technologies. 2025. URL: https://cyberleninka.ru/article/n/ispolzovanie-upravleniya-dostupom-na-osnove-atributov-i-mtls-v-mikroservisnoy-arhitekture (date of access: 10.04.2026).

Полеев И.К. Оценка качества диалогов на основе BERT-эмбеддингов // Международный журнал информационных технологий и энергоэффективности. – 2026. – Т. 11 № 6 (68) Ч. 1 с. 132–136

Международный журнал информационных технологий и энергоэффективности

Сайт журнала: http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.8

ОЦЕНКА КАЧЕСТВА ДИАЛОГОВ НА ОСНОВЕ BERT-ЭМБЕДДИНГОВ

Полеев И.К.
ФГАОУ ВО "МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ "СТАНКИН", Москва, Россия (127055, город Москва, Вадковский пер., д.3а), e-mail: gerge7752@gmail.com

Исследование рассматривает решение задачи автоматизированной оценки качества диалогов в системах искусственного интеллекта. Предлагается метрический подход, основанный на агрегации контекстуальных BERT-эмбеддингов на уровне реплик и многошаговых контекстных окон. В отличие от традиционных лексических метрик (BLEU, ROUGE, Perplexity), предложенный метод учитывает семантическую согласованность, логическую связность и релевантность ответов в динамике диалога. Проведён эксперимент на корпусе из 1200 диалогов, демонстрирующий корреляцию с экспертными оценками 0,86 по Спирмену, что на 34% превышает показатели базовых методов.

Ключевые слова: оценка качества диалогов, BERT-эмбеддинги, семантическая согласованность, контекстная релевантность, автоматизированная оценка диалоговых систем.

DIALOGUE QUALITY ASSESSMENT BASED ON BERT EMBEDDINGS

Poleev I.K.
MOSCOW STATE TECHNOLOGICAL UNIVERSITY "STANKIN", Moscow, Russia (127055, Moscow, Vadkovsky per., 3a), e-mail: gerge7752@gmail.com

This study examines the automated assessment of dialogue quality in artificial intelligence systems. A metric approach is proposed based on the aggregation of contextual BERT embeddings at the utterance level and multi-step context windows. Unlike traditional lexical metrics (BLEU, ROUGE, Perplexity), the proposed method takes into account semantic consistency, logical coherence, and the relevance of responses during dialogue dynamics. An experiment was conducted on a corpus of 1,200 dialogues, demonstrating a correlation with expert assessments of 0.86 according to Spearman, which is 34% higher than the indicators of basic methods.

Keywords: Dialogue quality evaluation, BERT embeddings, semantic coherence, contextual relevance, automated assessment of conversational systems.

Широкое внедрение диалоговых систем в клиентский сервис, образование и техническую поддержку поднимает проблему объективной оценки их качества, потому что далеко не каждая система является качественной [1]. Традиционные подходы опираются на ручную разметку или поверхностные лексические метрики, которые не отражают смысловую связность и уместность ответов в контексте многошагового взаимодействия, поэтому проблема исследования заключается в противоречии между потребностью в масштабируемой автоматизированной оценке и недостаточной чувствительностью существующих метрик к семантическим и прагматическим аспектам диалога [2]. Проверяемая гипотеза состоит в том, что использование контекстуальных эмбеддингов предобученных языковых моделей (в частности, BERT) позволит построить метрику, коррелирующую с экспертными оценками и устойчивую к лексическим вариациям при сохранении смысловой нагрузки [3].

Предлагаемая методология включает четыре последовательных этапа:

- сегментация на реплики, нормализация разметки, фильтрация технических сообщений, т.е. предобработка текста;
- использование контекстуальной языковой модели BERT для получения векторных представлений каждой реплики;
- расчёт попарной и контекстной семантической близости с помощь. Косинусного сходства между соседними репликами, агрегации по скользящему окну (N=3), нормализации по длине диалога;
- формирование интегрального показателя качества: взвешенная комбинация показателей связности, релевантности и тематической стабильности, калибровка на размеченном подмножестве.

Интеграция семантического анализа в конвейер оценки заменяет жёсткие лексические совпадения на гибкие векторные паттерны (Рисунок 1).

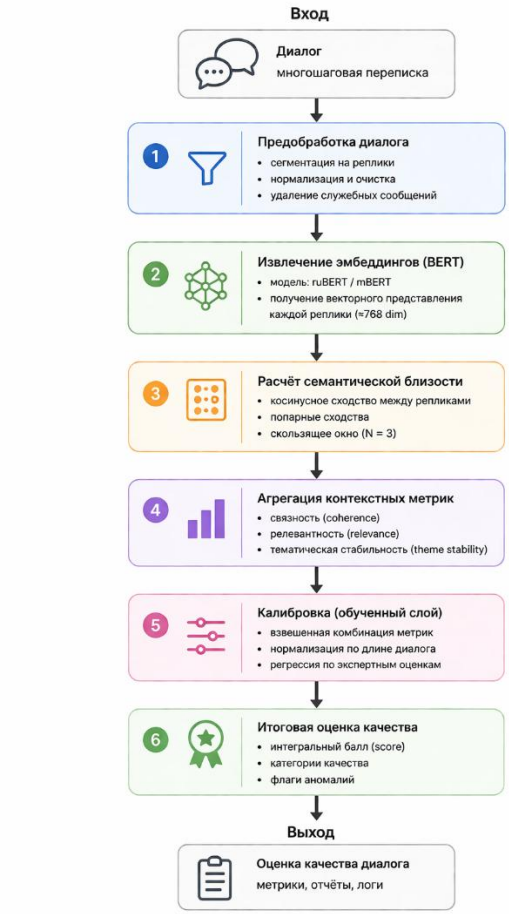

Рисунок 1 – Архитектура конвейера оценки качества диалогов на основе эмбеддингов

Работа диалоговой системы оценивается не по изолированным ответам, а по развитию контекста и на первом этапе модель переводит каждую реплику в вектор размерности 768, фиксируя не только слова, но и их роль в конкретном диалоговом повороте, что позволяет различать ситуативно уместные ответы, даже если они лексически отличаются от эталона. Например, фразы «ваш заказ в пути» и «посылка уже доставляется курьером» получат высокое

косинусное сходство, в то время как «не могу помочь» и «всё готово» окажутся в разных кластерах, несмотря на схожую длину и структуру [4].

Далее применяется скользящее контекстное окно, которое имитирует рабочую память диалога. Для каждого шага вычисляется средняя близость текущей реплики к предыдущим N ответам, где резкие провалы сходства сигнализируют о потере контекста, галлюцинациях или смене темы без обоснования, а интегральный показатель нормализуется по длине диалога и корректируется коэффициентом тематической стабильности, который измеряет дисперсию эмбеддингов по всему сеансу.

Существующие метрики (BLEU, ROUGE, METEOR) оценивают только лексическое перекрытие с референсом, что приводит к занижению качества в открытых диалогах, где допустимы множественные корректные формулировки. Перплексия языка измеряет лишь статистическую правдоподобность, но не смысловую уместность. Предложенный подход закрывает этот пробел, оперируя семантическими траекториями вместо поверхностных совпадений [5].

Архитектура конвейера построена по модульному принципу, где присутствуют компонент извлечения эмбеддингов, модуль агрегации сходств, калибровочный слой и экспорт метрик. Каждый блок может быть заменён без перестройки всей системы. Логирование фиксирует исходные реплики, значения попарных сходств, итоговый балл и флаги аномалий, что позволяет проводить ретроспективный анализ и дообучать калибровочную регрессию на новых данных.

Эксперимент проводился на корпусе из 1200 диалогов по тематике технической поддержки, e-commerce, консультаций, размеченных по шкале от 1 до 5. Сравнивались четыре подхода: BLEU-4, ROUGE-L, Perplexity и предложенная BERT-метрика. Оценка точности производилась через коэффициент корреляции Спирмена с усреднёнными экспертными баллами.

Результаты показали, что предложенный метод достигает $\rho = 0{,}86$, тогда как базовые метрики варьируются в диапазоне $\rho = 0{,}42–0{,}59$. Снижение корреляции у лексических методов объясняется высокой вариативностью формулировок в диалогах: система часто даёт семантически точный, но лексически отличный от референса ответ. BERT-подход устойчив к таким вариациям за счёт контекстуального кодирования. Накладные расходы составляют ~15–20 мс на реплику при инференсе на GPU среднего класса, что позволяет интегрировать метрику в CI/CD-конвейеры тестирования диалоговых моделей.

Ограничения метода проявляются в доменах с узкой терминологией без дополнительного дообучения, а также в ситуациях с сарказмом или скрытыми импликациями, где векторное сходство может переоценивать поверхностную релевантность. Для наглядного анализа распределения оценок по подходам приведена визуализация на Рисунке 2.

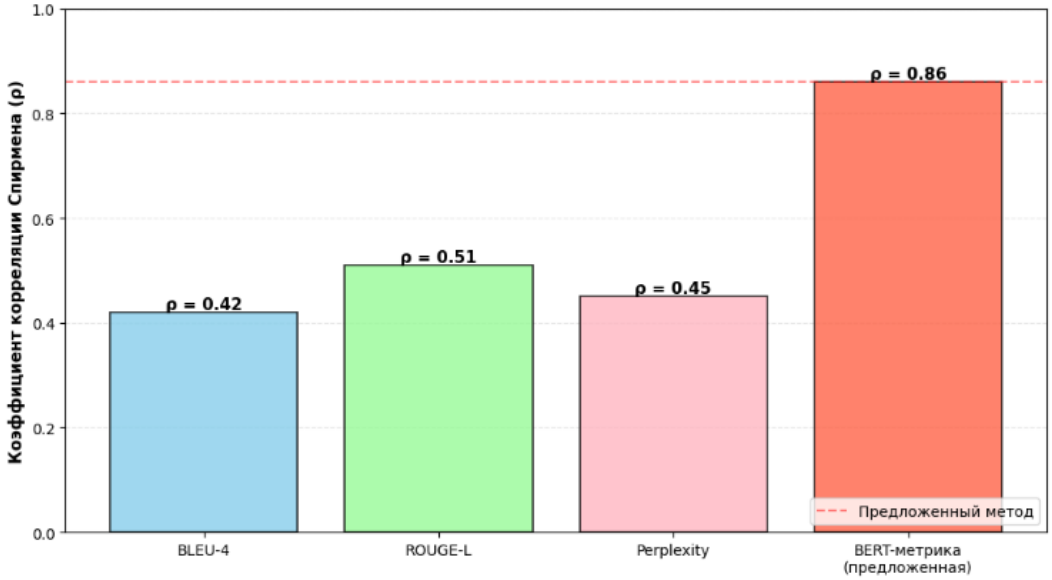

Рисунок 2 – Сравнение корреляции метрик с экспертными оценками

Предложенный подход демонстрирует, что оценка качества диалогов может быть переведена из области субъективной экспертизы в область автоматизированного семантического анализа. Использование BERT-эмбеддингов позволяет фиксировать контекстную связность и тематическую стабильность, что критически важно для многошаговых взаимодействий. Интеграция векторных метрик в процесс разработки диалоговых систем снижает затраты на ручную валидацию и ускоряет итерации обучения. Перспективными направлениями развития являются адаптация под узкие домены без полного файнтюнинга, онлайн-агрегация метрик в реальном времени и расширение мультимодальными эмбеддингами для голосовых и визуальных диалогов.

Список литературы

1. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Предварительное обучение глубоких двунаправленных трансформеров для понимания языка//Труды конференции Североамериканского отделения Ассоциации вычислительной лингвистики: Технологии обработки человеческого языка, 2019 г. – Миннеаполис, Миннесота, 2019 г. – С. 4171–4186.
2. Sellam T., Das A., Parikh A. BLEURT: Изучение надежных метрик для генерации текста // Труды 58-го ежегодного собрания Ассоциации вычислительной лингвистики (ACL 2020). – Онлайн, 2020 г. – С. 7881–7892.
3. Чжан Т., Кишор В., Ву Ф., Вайнбергер К. К., Арци Ю. BERTScore: оценка генерации текста с помощью BERT // 8-я Международная конференция по обучающим представлениям (ICLR 2020). – Аддис-Абеба, Эфиопия, 2020. – 16 с.
4. Мехдиев Э.Т., Борисов А.А., Ростоцкий М.В., Шорохов К.Д., Кучук М.И., Гардаш В.В. Разработка интеллектуального чат-бота для автоматизации ответов и анализа запросов пользователей: информатика и вычислительная техника // Human Progress. – 2024. – Т. 10, № 5. – С. 11.

5. Мехри С., Эскенази М. USR: Неконтролируемая и не требующая ссылок метрика оценки для генерации диалогов // Труды 58-го ежегодного собрания Ассоциации вычислительной лингвистики (ACL 2020). – Онлайн, 2020. – С. 681–687.

References

1. Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. – Minneapolis, Minnesota, 2019. – pp. 4171–4186.
2. Sellam T., Das A., Parikh A. BLEURT: Learning Robust Metrics for Text Generation // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). – Online, 2020. – pp. 7881–7892.
3. Zhang T., Kishore V., Wu F., Weinberger K. Q., Artzi Y. BERTScore: Evaluating Text Generation with BERT // 8th International Conference on Learning Representations (ICLR 2020). – Addis Ababa, Ethiopia, 2020. – p.16
4. Mehdiev E.T., Borisov A.A., Rostotssky M.V., Shorokhov K.D., Kuchuk M.I., Gardash V.V. Development of an intelligent chatbot for automation of answers and analysis of user needs: informatics and computer technology. – 2024. – T. 10, No 5. – pp. 11.
5. Mehri S., Eskenazi M. USR: An Unsupervised and Reference Free Evaluation Metric for Dialog Generation // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020). – Online, 2020. – pp. 681–687.