Министерство образования и науки РФ
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»
им. В.И. Ульянова (Ленина)

XIV ВСЕРОССИЙСКАЯ НАУЧНО- ПРАКТИЧЕСКАЯ КОНФЕРЕНЦИЯ С МЕЖДУНАРОДНЫМ УЧАСТИЕМ

«НАУКА НАСТОЯЩЕГО И БУДУЩЕГО»

ДЛЯ СТУДЕНТОВ, АСПИРАНТОВ И МОЛОДЫХ УЧЕНЫХ

Сборник материалов конференции

16 – 18 мая 2026

Том IV

Санкт-Петербург
2026

УДК 001.2

XIV Всероссийская научно-практическая конференция с международным участием «НАУКА НАСТОЯЩЕГО И БУДУЩЕГО» для студентов, аспирантов и молодых ученых.
Том 4. Сборник материалов конференции. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2026. 97 с.

Организаторы:
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Тематика конференции включает следующие направления

- *Информационные радиотехнические системы и устройства*
- *Информатика и управление в технических системах и ВТ*
- *Программная инженерия и автономные интеллектуальные системы*
- *Искусственный интеллект в прикладных областях*
- *Алгоритмическая математика*
- *Управление и обработка информации*
- *Приборостроение*
- *Лингвистика*
- *Электроника, нанотехнологии, наноматериалы*
- *Системный анализ и информационная безопасность*
- *Электрические системы, привод, автоматика и электротехнологии*
- *Биотехнические системы и технологии*
- *Техносферная безопасность*
- *Реклама и связи с общественностью*
- *Современные тренды управления качеством и цифровая экономика*
- *Инновационное проектирование: от реальных объектов к цифровым двойникам.*

Основной задачей конференции является развитие творческой активности студентов, привлечение их к решению актуальных задач в области науки и техники.

Сборник материалов содержит доклады, представленные на XIV Научно практической конференции с международным участием «Наука настоящего и будущего» для студентов, аспирантов и молодых ученых, состоявшейся 16 – 18 мая 2026 года в Санкт-Петербурге, рекомендованные к публикации. Все доклады проходят рецензирование.

Оглавление

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В ПРИКЛАДНЫХ ОБЛАСТЯХ

АВТОМАТИЧЕСКОЕ ИЗВЛЕЧЕНИЕ МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ И ТЕКСТА НА РУССКОМ ЯЗЫКЕ ИЗ ИЗОБРАЖЕНИЙ И ИХ КОНВЕРТАЦИЯ В КОД LATEX

Будило Е.А.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

Аннотация. В настоящей работе рассматривается задача автоматического извлечения математических выражений и текста на русском языке из изображений с последующей конвертацией результата в код LaTeX. В качестве базовой архитектуры выбрана модель TexTeller, основанная на связке ViT и TrOCR. Для повышения качества распознавания был сформирован смешанный набор данных, включающий русскоязычные и англоязычные тексты из Википедии, а также изображения формул из набора latex-formulas-80M. Помимо этого, токенизатор модели был расширен символами кириллицы. Модель была дообучена на наборе данных из 200 000 пар «изображение – код LaTeX» и оценена на отдельном тестовом наборе данных. В результате экспериментов были получены значения CER = 0.0313 и Exact Match = 0.709, что подтверждает эффективность предложенного подхода для распознавания русскоязычных научных материалов.

Ключевые слова: LaTeX, распознавание формул, машинное обучение, компьютерное зрение, обработка естественного языка, математические выражения

Введение

Преобразование изображений, содержащих текст и формулы, в читаемый формат представляет собой значимую проблему в области оцифровки документов. Актуальность исследования автоматического извлечения математических выражений и русскоязычного текста из изображений с конвертацией в код LaTeX обусловлена растущим объёмом научных материалов в цифровом формате (сканированные документы, PDF), где ручной ввод формул и текста требует значительных временных затрат и подвержен ошибкам. [1]

Существующие решения в области оптического распознавания символов (OCR) и распознавания формул показывают высокие результаты на английском языке и стандартных наборах данных, однако качество работы снижается при обработке русскоязычных текстов и смешанных документов, содержащих одновременно формулы и текст. Кроме того, многие модели ориентированы на ограниченный набор символов и плохо обрабатывают сложную типографику, нестандартные шрифты и шумы сканирования, что ограничивает их применение в задачах оцифровки научных материалов [2].

Цель работы – создание модели машинного обучения, способной автоматически извлекать математические выражения и текст на русском языке из изображений и конвертировать в код LaTeX. Для достижения цели были поставлены задачи: выбор базовой архитектуры модели, формирование набора данных для дообучения, обучение модели на данном наборе и проведение тестирования с оценкой качества распознавания.

Выбор базовой модели

Был проведён анализ аналогов в области конвертации изображений в код LaTeX. Среди рассмотренных аналогов в качестве базовой модели был выбран TexTeller [3], поскольку он обладает рядом преимуществ:

- Модель обучена на наборе данных из более чем 80 млн пар данных «изображение – код LaTeX», а также дообучена на наборах данных для распознавания текста на английском и китайском языках;
- Открытый исходный код с лицензией Apache-2.0 и открытые наборы данных, на которых обучена модель.

Архитектура TexTeller представляет собой encoder-decoder архитектуру и основана на моделях ViT (Vision Transformer) [4] и TrOCR (Transformer-based OCR – оптическое распознавание символов на основе трансформера) [5]. ViT используется для извлечения векторов-признаков из изображения, а TrOCR используется для декодирования в токены кода LaTeX. Дополнительно TexTeller включает модуль детекции формул для обработки смешанных изображений, содержащих одновременно текст и математические выражения.

Сбор набора данных

Поскольку целью работы является конвертация не только математических формул из изображений в код LaTeX, но и конвертация русскоязычного текста, то модель необходимо дообучить на изображениях с текстом на русском языке. При этом в этот набор данных необходимо добавить изображения с формулами и англоязычным текстом, чтобы модель не переобучалась и не теряла способность распознавать формулы и английский текст.

Для русского и английского языков использовался открытый набор данных с Википедии (20231101.ru и 20231101.en) [6]. После скачивания набора данных проводилась его очистка: все символы, кроме знаков препинания, цифр и соответствующего языка, исключаются из текста, чтобы не было лишних токенов, которых нет в токенизаторе. После очистки генерировалось изображение размером 900 × 420 пикселей с размером шрифта от 22 пт до 30 пт и случайным образом выбирался один из шрифтов: Arial, Times New Roman, Calibri, Cambria, DejaVu Sans и DejaVu Serif. После генерации изображения в качестве метки (кодом LaTeX) использовался текст, обёрнутый в команду LaTeX «\text{}».

Для математических выражений использовался набор данных latex-formulas-80M, содержащий пары «изображение – LaTeX-формула» [7].

При создании набора данных использовались следующие вероятности для языков: с вероятностью 0.2 текст был на английском языке, с вероятностью 0.8 – на русском. При этом формула может содержать и другой язык, который присутствует в токенизаторе.

После выбора языка при создании набора данных использовались вероятности: с вероятностью 0.35 на изображении присутствует только текст, с вероятностью 0.45 – на изображении только LaTeX-формула, с вероятностью 0.2 – на изображении и текст, и формула. В случае генерации изображения с текстом и формулой текст генерируется отдельно, а затем изображение текста объединяется с изображением формулы из набора данных latex-formulas-80M.

Пример сгенерированного изображения представлен на рисунке 1.

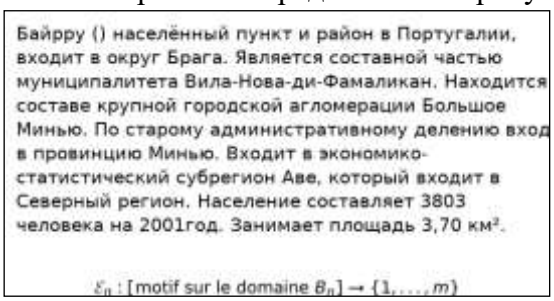


Рис. 1. Пример сгенерированного изображения для набора данных

Таким образом сформирован набор данных из 200 000 пар «изображение – код LaTeX».

Обучение и тестирование

Перед обучением модели TexTeller на собранном датасете в токенизатор необходимо добавить токены с прописными и строчными буквами алфавита русского языка. Таким образом, исходный токенизатор со словарём в 15 000 токенов дополняется 66 токенами.

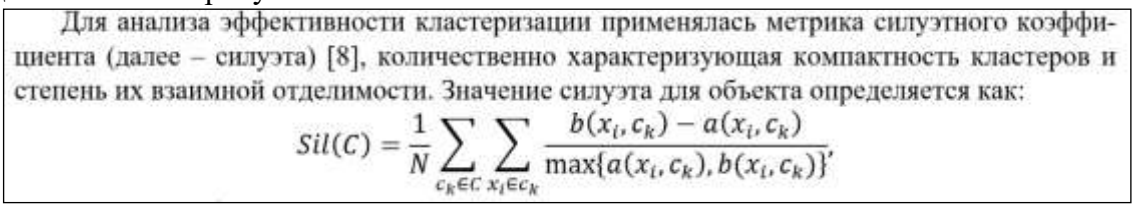
Для обучения модели использовался фреймворк машинного обучения PyTorch [8]. В качестве функции потерь использовалась Cross-Entropy Loss, которая измеряет, насколько предсказанные моделью вероятности токенов отличаются от правильных. При обучении использовался планировщик скорости обучения CosineAnnealingLR [9] и оптимизатор AdamW [10]. Размер обучающего пакета – 1. Начальная скорость обучения – 0.5×10^{-4}.

После обучения была проведена оценка модели на тестовом наборе данных. Тестовый набор данных был сгенерирован отдельно от основного набора и содержит 10 000 пар «изображение – код LaTeX». Для оценки качества модели использовались метрики Character Error Rate (CER) и Exact Match (EM). CER показывает долю ошибок на уровне символов (вставки, удаления, замены) между предсказанным и эталонным кодом LaTeX, а EM отражает долю примеров, где результат модели совпадает с правильной строкой LaTeX.

На тестовой выборке получены значения CER = 0.0313 и EM = 0.709. Это означает, что в среднем модель допускает около 3.13% символьных ошибок, а также полностью правильно восстанавливает LaTeX-код примерно в 70.9% случаев.

Низкое значение CER указывает на высокую точность распознавания и небольшое количество ошибок даже при несовпадении строк, тогда как EM остаётся более строгой метрикой и её значение снижается из-за чувствительности к пробелам, эквивалентным вариантам записи формул и мелким синтаксическим отличиям LaTeX.

В качестве примера работоспособности модели на вход подаётся изображение, представленное на рисунке 2.


Для анализа эффективности кластеризации применялась метрика силуэтного коэффициента (далее – силуэта) [8], количественно характеризующая компактность кластеров и степень их взаимной отделимости. Значение силуэта для объекта определяется как:

$$Sil(C) = \frac{1}{N} \sum_{c_k \in C} \sum_{x_i \in c_k} \frac{b(x_i, c_k) - a(x_i, c_k)}{\max\{a(x_i, c_k), b(x_i, c_k)\}},$$

Рис. 2. Пример работоспособности модели

Вывод модели для примера представлен на рисунке 3. Скомпилированный код LaTeX представлен на рисунке 4.

```
\text{Для анализа эффективности кластеризации применялась метрика силуэтного коэффи-циента
 (далее - силуэта) [8], количественно характеризующая компактность кластеров и степень их
взаимной отделимости. Значение силуэта для объекта определяется как.} \[Sil(C)=\frac{1}{N}
\sum_{c_{k}\in\mathcal{C}}\sum_{x_{i}\in c_{k}}\frac{b(x_{i},c_ {k})-a(x_{i},c_{k}))}{\max\
{a(x_{i},c_{k}),b(x_{i},c_{k}))\}},\]
```

Рис. 3. Вывод модели примера

Для анализа эффективности кластеризации применялась метрика силуэтного коэффи-циента (далее - силуэта) [8], количественно характеризующая компактность кластеров и степень их взаимной отделимости. Значение силуэта для объекта определяется как.

$$Sil(C) = \frac{1}{N} \sum_{c_k \in \mathcal{C}} \sum_{x_i \in c_k} \frac{b(x_i, c_k) - a(x_i, c_k)}{\max\{a(x_i, c_k), b(x_i, c_k)\}},$$

Рис. 4. Скомпилированный вывод примера

Из результата работы модели видно, что в русскоязычном тексте и формуле отсутствуют смысловые ошибки. Однако наблюдаются две неточности: разрыв слова «коэффициента» и ошибочное распознавание знака препинания, при котором вместо двоеточия была распознана точка.

Ошибка разрыва слова присутствует, так как собранный набор данных генерировался без переносов, поэтому модель этому не научилась.

Вторая ошибка допущена по той причине, что большинство текстов в наборе данных заканчивались точкой, а последующая формула никак не была связана с текстом выше. В связи с этим модель обучилась закономерности, что в конце текста должны быть точка и закрывающая фигурная скобка «.}».

Заключение

Была достигнута цель – создание модели, способной автоматически извлекать математические выражения и текст на русском языке из изображений и конвертировать в код LaTeX. В качестве базовой модели для дообучения использовался TexTeller [3]. Для дообучения был собран набор данных из LaTeX-формул и текстов на русском и английском языках, состоящий из 200 000 пар. Модель была дообучена на собранном наборе данных и протестирована на отдельном наборе данных объёмом 10 000 пар, показав метрики CER = 0.0313 и EM = 0.709. Таким образом, модель демонстрирует высокую точность распознавания изображений, однако полное совпадение предсказания с эталоном достигается в 70.9% случаев, поскольку метрика EM является строгой и чувствительна к синтаксическим различиям в записи LaTeX.

Перспективами дальнейших исследований являются расширение набора данных, например, включением большего числа используемых шрифтов, а также использования полужирного, курсивного и подчёркнутого стилей форматирования текстов. Также при обучении можно использовать различные аугментации, чтобы сгенерированные изображения с текстом были приближены к реальным сканам документов, и обученная модель на таких данных могла бы конвертировать в код LaTeX.

Список литературы

1. AlKendi W. et al. Advancements and challenges in handwritten text recognition: A comprehensive survey //Journal of Imaging. – 2024. – Т. 10. – №. 1. – С. 18.
2. Orji E. Z. et al. Advancing ocr accuracy in image-to-latex conversion — a critical and creative exploration //Applied Sciences. – 2023. – Т. 13. – №. 22. – С. 12503.

3. OleehyO. TexTeller. [Электронный ресурс] // GitHub. – 2025. – URL: https://github.com/OleehyO/TexTeller (дата обращения: 03.05.2026)
4. Dosovitskiy A. et al. An image is worth 16x16 words: Transformers for image recognition at scale //arXiv preprint arXiv:2010.11929. – 2020.
5. Li M. et al. Trocr: Transformer-based optical character recognition with pre-trained models //Proceedings of the AAAI conference on artificial intelligence. – 2023. – Т. 37. – №. 11. – С. 13094-13102.
6. Wikimedia Wikipedia Dataset (20231101.ru) [Электронный ресурс]. – Режим доступа: https://huggingface.co/datasets/wikimedia/wikipedia (дата обращения: 03.05.2026).
7. OleehyO. latex-formulas-80M [Электронный ресурс]. – URL: https://huggingface.co/datasets/OleehyO/latex-formulas-80M (дата обращения: 03.05.2026).
8. Paszke A. Pytorch: An imperative style, high-performance deep learning library // arXiv preprint arXiv:1912.01703. – 2019.
9. Loshchilov I., Hutter F. Sgdr: Stochastic gradient descent with warm restarts // arXiv preprint arXiv:1608.03983. – 2016.
10. Loshchilov I., Hutter F. Decoupled weight decay regularization // arXiv preprint arXiv:1711.05101. – 2017.

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ОЦЕНКИ КАЧЕСТВА ДАТАСЕТОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ ЗАДАЧ ОБНАРУЖЕНИЯ ОБЪЕКТОВ

ЖМУРЕНКО В.К.

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Аннотация. Разработана локальная система автоматизированной оценки качества YOLO-датасетов для задач обнаружения объектов. Система выявляет нарушения структуры датасета, ошибки целостности пар image/label, некорректные label-файлы, недопустимые class_id и ошибки bounding boxes. Тестирование на исходном и модифицированном вариантах Aquarium Dataset подтвердило работоспособность системы и ее способность обнаруживать заранее внесенные дефекты до этапа обучения модели.

Ключевые слова: компьютерное зрение, датасет, YOLO, object detection, аннотация, bounding box, аудит данных, качество данных.

Введение

Качество аннотированных данных напрямую влияет на надежность моделей компьютерного зрения. В задачах обнаружения объектов ошибки в разметке, некорректные bounding boxes, отсутствующие label-файлы, битые изображения и нарушения структуры датасета могут искажать процесс обучения и снижать воспроизводимость результатов.

В современных подходах к разработке систем искусственного интеллекта все большее внимание уделяется качеству данных. Это соответствует направлению data-centric AI, в котором данные рассматриваются как ключевой объект анализа и улучшения [1]. Для object detection особенно важны корректность классов, локализации объектов и файлов разметки [2].

Ручная проверка датасета трудоемка и плохо масштабируется, поэтому актуальна разработка автоматизированных средств первичного аудита до этапа обучения модели.

Целью работы является разработка локальной автоматизированной системы оценки качества датасетов компьютерного зрения для задач обнаружения объектов, выявляющей типовые структурные и аннотационные ошибки в YOLO-датасетах.

Для достижения цели были поставлены задачи: проанализировать структуру YOLO detection dataset; выделить ошибки, обнаруживаемые без обучения нейронной сети; реализовать загрузку датасета и чтение конфигурации; разработать правила проверки изображений, label-файлов, классов и bounding boxes; сформировать механизм агрегации нарушений и экспорт результатов аудита.

Методика системы

Разработанная система ориентирована на локальную проверку датасетов object detection в YOLO-формате. В этом формате каждому изображению соответствует label-файл, содержащий строки вида class_id x_center y_center width height, где координаты bounding box задаются в нормализованном виде. Согласно документации Ultralytics, значения координат должны находиться в диапазоне от 0 до 1, а идентификаторы классов должны соответствовать списку классов датасета [3].

Методика проверки строится как последовательный pipeline. На вход системе передается путь к корню локального YOLO-датасета. Сначала проверяется наличие и корректность файла data.yaml, из которого извлекаются список классов и количество классов. Затем определяется структура директорий. В текущей реализации поддерживаются две распространенные схемы организации датасета:

- train/images, train/labels, val/images, val/labels
- images/train, labels/train, images/val, labels/val.

После определения структуры строится индекс элементов датасета. Каждый элемент описывает предполагаемую пару изображения и label-файла, split, пути к файлам, факт наличия изображения и аннотации, метаданные изображения, список распарсенных аннотаций и список ошибок парсинга. Для изображений выполняется проверка читаемости с использованием Pillow, на этом этапе система не анализирует содержимое изображения с точки зрения семантики, а проверяет, может ли файл быть корректно открыт и верифицирован как изображение. Это позволяет выявлять битые файлы, которые имеют допустимое расширение, но не могут быть использованы при последующей подготовке или обучении модели.. Для label-файлов выполняется построчный разбор YOLO-разметки.

На этапе применения правил система выявляет несколько групп ошибок.

1. Правило целостности фиксирует отсутствующие изображения, отсутствующие label-файлы, неподдерживаемые расширения изображений, битые изображения и нечитаемые файлы аннотаций.

2. Правило аннотаций проверяет пустые label-файлы, пустые строки, неверное число токенов, нечисловые значения, некорректные размеры bounding box, выход координат за нормализованные границы и выход bounding box за пределы изображения после денормализации.

3. Правило классов проверяет соответствие class_id диапазону классов, объявленных в data.yaml.

4. Дополнительно реализованы проверки split-структуры, базовой статистики датасета и точных дубликатов изображений по SHA-256-хэшу.

Типы ошибок приведены в таблице 1.

Таблица 1

Типы выявляемых ошибок и способы их обнаружения

Тип ошибки	Способ обнаружения	Finding
Отсутствующее изображение	Label-файл есть, ожидаемое изображение не найдено	missing_image
Отсутствующий label-файл	Изображение есть, соответствующий .txt не найден	missing_label
Битое изображение	Ошибка чтения или проверки через Pillow	corrupted_image
Неверная строка разметки	В строке не 5 токенов или есть нечисловые значения	invalid_annotation_line
Недопустимый класс	class_id вне диапазона из класса data.yaml	invalid_class_id
Некорректный bbox	width <= 0 или height <= 0	invalid_bbox_dimensions
Bbox вне границ	Координаты вне [0, 1] или выход за пределы изображения	bbox_out_of_bounds

Результатом проверки является набор findings. Каждый finding содержит тип нарушения, уровень критичности, рекомендуемое действие, сообщение, имя правила, split, пути к изображению и label-файлу, а также дополнительные детали.

Тестирование

Для проверки работы системы использовались два варианта одного набора данных: исходный Aquarium Dataset [6] и его модифицированная версия aquarium_mutated. Датасет Aquarium размещен на платформе Roboflow и относится к задаче обнаружения объектов: он содержит изображения аквариумных объектов и соответствующую YOLO-разметку.

Первый запуск выполнялся на исходном датасете aquarium. Он использовался как контрольный сценарий для проверки загрузки структуры датасета, чтения изображений, разбора label-файлов, проверки классов и bounding boxes, а также формирования отчетов. Результаты запуска представлены на рис. 1 и рис. 2. По итогам проверки исходного датасета было обработано 637 изображений, 637 label-файлов и 4817 аннотаций; валидатор сформировал 3 finding, включая 2 critical finding.

Рис. 1. Результат запуска валидатора на исходном датасете

Рис. 2. Сгенерированный отчет запуска на исходном датасете

Второй запуск выполнялся на датасете aquarium_mutated, подготовленном на основе исходного датасета. В него были намеренно внесены контролируемые ошибки: отсутствие изображения, отсутствие label-файла, поврежденное изображение, некорректные строки

YOLO-разметки, нечисловые значения, class_id вне допустимого диапазона, bounding boxes с нулевой или отрицательной шириной/высотой, а также bounding boxes, выходящие за допустимые границы. Результаты запуска представлены на рис. 3 и рис. 4. В этом случае система обработала 636 изображений, 636 label-файлов и 4785 аннотаций; было сформировано 15 findings, из которых 13 имеют уровень critical.

Рис. 3. Результат запуска валидатора на датасете с умышленно внесенными ошибками

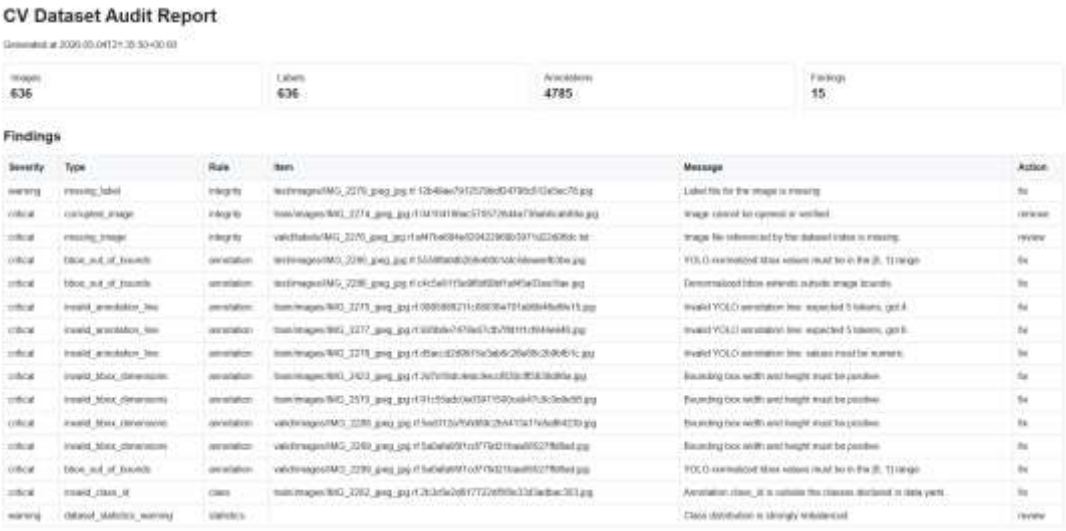


Рис. 4. Сгенерированный отчет запуска на датасете с умышленно внесенными ошибками

Таким образом, тестирование подтвердило работоспособность системы: валидатор корректно обработал исходный датасет и выявил контролируемые ошибки, внесенные в модифицированную версию. Полученные результаты показывают, что система выполняет заявленную задачу автоматизированного обнаружения дефектов YOLO-датасета до этапа обучения модели.

Выводы

Разработанная система подтвердила возможность автоматизированного выявления типовых ошибок YOLO-датасетов до этапа обучения модели. Полученные результаты показывают, что такая валидация может использоваться как отдельный этап подготовки данных для задач обнаружения объектов, улучшая их качество.

Список литературы

1. Jarrahi M. H., Memariani A., Guha S. The Principles of Data-Centric AI //Communications of the ACM. URL: https://arxiv.org/abs/2211.14611 (дата обращения: 01.05.2026).
2. Tkachenko U., Thyagarajan A., Mueller J. ObjectLab: Automated Diagnosis of Mislabeled Images in Object Detection Data [Электронный ресурс]. URL: https://arxiv.org/abs/2309.00832 (дата обращения: 01.05.2026).
3. Ultralytics. Object Detection Datasets Overview [Электронный ресурс]. URL: https://docs.ultralytics.com/datasets/detect/ (дата обращения: 01.05.2026).

4. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection [Электронный ресурс]. URL: https://arxiv.org/abs/1506.02640 (дата обращения: 01.05.2026).
5. Northcutt C. G., Jiang L., Chuang I. Confident Learning: Estimating Uncertainty in Dataset Labels //Journal of Artificial Intelligence Research. [Электронный ресурс]. URL: https://research.google/pubs/confident-learning-estimating-uncertainty-in-dataset-labels/ (дата обращения: 05.05.2026).
6. Aquarium Dataset [Электронный ресурс]. URL: https://universe.roboflow.com/brad-dwyer/aquarium-combined (дата обращения: 05.05.2026).

АЛГОРИТМ СОЗДАНИЯ СЛАБОСТРУКТУРИРОВАННЫХ СИСТЕМНЫХ ЖУРНАЛОВ С РАЗМЕТКОЙ СОБЫТИЙ С ИСПОЛЬЗОВАНИЕМ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Зазуля И. А., Семидолин А. Д.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

Аннотация. В работе представлен алгоритм для создания слабоструктурированных системных журналов с разметкой событий с использованием большой языковой модели для формирования текстового сообщения. Алгоритм предусматривает эволюционирование журналов со временем и предоставляет возможность задавать долю аномалий среди создаваемых записей. Экспериментальная оценка показала, что при различной доле аномалий средняя скорость генерации составила около 1762 записей/с. Полученные результаты могут быть использованы для восполнения наборов данных системных журналов и применены для улучшения алгоритмов в задачах обнаружения аномалий и поиска дублирующихся проблем по системным журналам.

Ключевые слова: системные журналы, большие языковые модели, наборы данных, автоматическая разметка

Введение

Современное программное обеспечение (ПО) с каждым годом становится всё сложнее и масштабнее. Так, например, всесторонне используемое ядро Linux на начало 2026 года содержало более 40 миллионов строк кода и последние годы прирастало на миллионы строк кода. С другой стороны, ПО операционной системы (ОС) может состоять из тысяч компонентов, зависимых друг от друга, каждый из которых имеет свою собственную версию и зависимости. ОС могут быть установлены на нескольких физических узлах и объединены в единый кластер для обеспечения избыточности и распределения нагрузки. Все эти факторы, создающие сложность подобных систем, требуют наличия инструментов и механизмов для обеспечения их эффективного обслуживания и разработки инженерами.

Одним из таких инструментов являются системные журналы, которые создаются и наполняются во время работы ПО и содержат информацию о выполнении программы. Данная информация прописывается разработчиками ПО и записывается в момент выполнения программы. Системные журналы в основном представляют собой слабоструктурированные текстовые файлы, содержащие записи, описывающие события, составляющие состояния системы в некоторые моменты времени. Эти записи могут включать сообщения об ошибках, информацию о работе программных модулей, данные о действиях пользователей и другие сведения. Системные журналы служат источником для анализа и мониторинга работы ПО инженерами различных уровней.

Проблематика

Для создания эффективных инструментов анализа системных журналов, как например, инструментов обнаружения аномалий, крайне важно использовать качественные и

репрезентативные данные. На текущий момент, с разметкой аномалий представлены такие открытые наборы данных, как HDFS (Hadoop Distributed System) [1, 2], BGL (BlueGene/L) [3], Thunderbird [3], Spirit [3], Open Stack [1, 4]. Данные наборы данных открыты и доступны для дальнейших исследований, их характеристики представлены в таблице 1.

Таблица 1

Характеристики существующих размеченных наборов данных

Наименование	Суммарная продолжительность журналов	Количество записей журналов	Размер необработанных журналов	Год появления
HDFS [1, 2]	38.7 часов	11 175 629	1,47 ГиБ	2009
BGL [3]	7 месяцев	4 747 963	708,76 МиБ	2007
Thunderbird [3]	8 месяцев	211 212 192	29,61 ГиБ	2007
Spirit [3]	1.5 года	272 298 969	37,35 Гиб	2007
Open Stack [1, 4]	29.5 часов	207 820	58,6 МиБ	2017

Представленные наборы данных частично устарели, не отражают всего многообразия ПО, не содержат современных подходов к записи событий в системные журналы и не учитывают изменчивость записей с течением времени в связи с обновлением компонентов ПО. С другой стороны, системные журналы с промышленных систем зачастую недоступны для исследования из-за наличия чувствительных данных в них или если доступны, то не размечаются из-за больших трудозатрат на разметку. Именно поэтому предложен алгоритм и разработано приложение для создания слабоструктурированных системных журналов с разметкой с динамическим изменением событий.

Алгоритм создания слабоструктурированных системных журналов

В результате разработана программа, в которой создание системных журналов формулируется как задача построения конечной последовательности записей на основе текстового описания предметной области. При этом учитываются минимально требуемый объём выборки и, при необходимости, заданная доля аномальных записей. Входными данными алгоритма служат пользовательский запрос (P), минимальное число записей (M) и параметр, задающий желаемый процент аномалий ($A \in \{0, \ldots, 100\}$]). Результатом работы является упорядоченное множество записей $E=\langle x_1, x_2, \ldots, x_m\rangle$, где $m \geq M$, сформированное в соответствии с заданными требованиями.

Каждая запись x_i имеет вид $x_i=(e_i, m_i, c_i)$, где e_i обозначает идентификатор события, m_i текст сообщения, c_i контекст записи, включающий временную метку, уровень важности, идентификаторы процесса и пользователя, коды состояния, признаки аномальности и другие служебные атрибуты.

Первым этапом работы алгоритма по текстовому описанию P с использованием большой языковой модели формируется множество сценариев $S = \{s_1, s_2, \ldots, s_n\}$. Каждый сценарий представляется кортежем $s_i=\{(n_i, \lambda_i, \omega_i, \pi_i, \gamma_i), \lambda_i$, где n_i имя сценария, λ_i уровень важности, $\omega_i > 0$ исходный вес, $\pi_i=\langle e_{i1}, e_{i2}, \ldots, e_k\rangle$ последовательность событий, γ_i базовый контекст. Если отдельные поля контекста не заданы, то они достраиваются автоматически на основе содержимого сценария. Таким образом, после этапа нормализации формируется конечное множество допустимых сценариев.

Для эффективного создания событий множество сценариев преобразуется в префиксный ориентированный граф $G=(V, A)$, в котором каждая вершина соответствует некоторому префиксу сценария, а каждое ребро $a \in A$ помечено событием. Вес ребра определяется суммой весов сценариев, проходящих через данное ребро. Пусть $A(v)$ множество исходящих рёбер вершины v, тогда вероятность выбора ребра $a \in A$, задаётся $p(a|v)=\frac{w(a)}{\sum_{b\in A(v)} w(b)}$, где $w(a)$ вес ребра. Для каждого сценария вычисляется эффективный вес по формуле 1:

$$W_i=w_i*\beta_i*(1+f_1+\overline{r_i})*\left(1+\frac{\eta}{1+u_i}\right) (1)$$

где w_i базовый вес сценария, β_i поправочный коэффициент по классу сценария, f_i текущая оценка приспособленности, $\overline{r_i}$ среднее накопленное вознаграждение, u_i число использований сценария, η коэффициент исследования.

Поскольку граф строится как префиксная структура, полная вероятность выбора сценария сводится к нормированному эффективному весу $P(s_i)=\frac{W_i}{\sum_{j=1}^{r} W_j}$.

Для каждого события в сценарии s_i определяется, относится ли оно к аномальному, если в результате анализа ему соответствует уровень важности **Warning**, **Error** или код состояния ошибки по формуле 2:

$$X(\mathrm{e}_{ij})=\begin{cases}1, \text{если событие является аномальным}\\ 0, \text{иначе}\end{cases} (2)$$

Тогда аномальность сценария вычисляет как доля аномальных событий в его последовательности по формуле 3:

$$a_i=\frac{1}{|\pi_i|}\sum_{j=1}^{|\pi_i|} X(\mathrm{e}_{ij}) (3)$$

Основной цикл алгоритма формирует итоговую выборку пакетами. На каждой итерации из графа выбирается одна последовательность событий π_i. После этого для всей последовательности строится общий контекст потока γ_i.

После выбора последовательности событий для каждого из них формируется текстовое сообщение журнала с использованием большой языковой модели. В базовом режиме сообщение строится на основе данных самого события и контекста. Идентификатор события разбивается на смысловые части, по которым определяется предметная область, компонент системы и характер действия. Затем из этих данных формируется короткое техническое сообщение, соответствующее выбранному событию. После вычисления служебных атрибутов каждая запись получает бинарный признак аномальности.

В процессе создания сценарный граф может периодически изменяться с вероятностью p_{evo}, что позволяет получать новые варианты последовательностей событий, а не ограничиваться только исходными сценариями.

Для оценки производительности разработанного алгоритма был проведён эксперимент, направленный на измерение скорости генерации записей системного журнала при различной заданной доле аномальных событий. В ходе эксперимента изменялся параметр целевой доли аномалий от 0 до 100 %, после чего фиксировалась скорость генерации записей журнала в секунду.

Таблица 2

Скорость генерации системных журналов при различной доле аномальных событий

Аномалий, %	0	10	20	30	40	50	60	70	80	90	100
Скорость генерации, записей/с	1667	2127	1851	1923	1562	2500	1639	1851	1538	1204	1515

По результатам эксперимента скорость генерации составила от 1204 до 2500 записей/с, среднее значение — около 1762 записей/с. Прямой зависимости между ростом доли аномалий и снижением производительности не выявлено, что подтверждает возможность использования алгоритма при различных целевых значениях доли аномальных событий.

Заключение

В работе предложен алгоритм создания слабоструктурированных системных журналов с автоматической разметкой событий на основе больших языковых моделей. Алгоритм формирует последовательности записей по текстовому описанию предметной области, учитывает контекст событий, позволяет задавать целевую долю аномалий и изменять сценарный граф в процессе генерации.

Экспериментальная оценка показала, что алгоритм сохраняет работоспособность при разных значениях доли аномальных событий и обеспечивает скорость генерации от 1204 до 2500 записей/с. Полученные результаты подтверждают применимость подхода для формирования размеченных наборов данных, используемых при разработке и тестировании методов анализа системных журналов и обнаружения аномалий.

Список литературы

1. J. Zhu, S. He, P. He, J. Liu and M. R. Lyu, "Loghub: A Large Collection of System Log Datasets for AI-driven Log Analytics," 2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE), Florence, Italy, 2023, pp. 355-366, doi: 10.1109/ISSRE59848.2023.00071.
2. Xu, W., Huang, L., Fox, A., Patterson, D., Jordan, M.I.: Detecting large-scale system problems by mining console logs. In: Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles. (2009).
3. Adam J. Oliner, Jon Stearley. What Supercomputers Say: A Study of Five System Logs, in Proc. of IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2007.
4. Du, M., Li, F., Zheng, G., Srikumar, V.: DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. (2017).

НОВЫЙ МАТЕМАТИЧЕСКИЙ МЕТОД ОБНАРУЖЕНИЯ АНОМАЛИЙ В ПРИРОДНЫХ ВРЕМЕННЫХ РЯДАХ

КАЛМАК Д.А.1, МАНДРИКОВА Б.С.1,2, МАНДРИКОВА О.В.2

СПбГЭТУ «ЛЭТИ»1, ИКИР ДВО РАН2

Аннотация. В статье представлен новый математический метод анализа природного временного ряда, основанный на применении нейронной сети архитектуры Колмогорова-Арнольда и пороговых функциях. Исследовались данные нейтронных мониторов [nmdb.eu], отражающие интенсивность космических лучей. На основе нейронной сети решается задача регрессии временного ряда. С помощью пороговых функций выполняется обнаружение аномалий во временном ряде, возникающих накануне и во время магнитных бурь. Результат важен для задач космической погоды.

Ключевые слова: нейронная сеть Колмогорова-Арнольда, машинное обучение, природные временные ряды, космические лучи, космическая погода.

Введение

Задачи прогноза возмущений в околоземном космическом пространстве и магнитных бурь с каждым годом набирают актуальность, поскольку солнечно-земные явления напрямую влияют на критически важную инфраструктуру и повседневную жизнь, а современные системы все сильнее зависят от надёжных предупреждений и управляемых мер. Известно, что в результате прохождения через Землю облаков магнитных и ударных структур, возникающих после событий на Солнце (высокоскоростных потоков солнечного ветра, Солнечных вспышек, корональных выбросов масс) в данных нейтронных мониторов могут возникать Форбуш-эффекты, характеризующиеся падением интенсивности космических лучей. Для обнаружения Форбуш-эффектов по данным нейтронных мониторов используются различные методы анализа данных. Например, метод [1] включает использование конструкций кратномасштабного анализа и кластерных нейронных сетей типа Learning vector quantization. Метод [1] реализовывает подавление помех и классификацию состояния околоземного космического пространства по регистрируемым данным на спокойное, слабо-возмущенное и возмущенное. К недостатку подхода [1] можно отнести временные затраты на выбор вейвлет-базиса и определение оптимальных параметров метода. GSM-метод [2] включает методы функций связи, траекторных расчётов и сферического анализа. Позволяет детектировать аномальные изменения в данных нейтронных мониторов разных станций, возникшие накануне и в период события. Метод может обнаруживать Форбуш-эффекты, предшествующие магнитным бурям и служащие их предикторами, однако требует комплексного подхода к анализу данных, включая дополнительные параметры (например, данные о межпланетной среде). Метод анализа градиента изменения потока нейтронов [3] анализирует градиент изменения потока за 6-часовой интервал. При обнаружении стабильной отрицательной динамики ниже определённого порога (например, −3%) системе присваивается метка «Форбуш». Такой подход снижает вероятность ложных срабатываний, однако зависит от корректности входных данных и настройки порога анализа.

С учетом существующих аналогов, в настоящей работе предложен математический метод анализа данных нейтронных мониторов, основанный на применении нейронной сети архитектуры Колмогорова-Арнольда и пороговых функций. На основе нейронной сети решается задача регрессии временного ряда. Вейвлет-фильтрация способствует точности регрессии, усилению полезного сигнала и убирает шум из данных. С помощью порогового отклонения выполняется обнаружение аномалий во временном ряде, возникающих накануне и во время магнитных бурь. Метод реализован программно и может быть использован в режиме онлайн. Результат важен для задач космической погоды.

Предлагаемый подход

Типичная архитектура сети KAN реализует теорему Колмогорова-Арнольда (теорему суперпозиции): многомерная непрерывная функция с ограниченной вариацией может быть представлена как конечная суперпозиция непрерывных одномерных функций:

$$Y(T) = Y(t_1, \dots, t_n) = \sum_{q=1}^{2n+1} \Theta_q \left(\sum_{p=1}^{n} \theta_{q,p}(t_p) \right). \quad (1)$$

Нейронная сеть KAN обучается на основе алгоритма обратного распространения ошибки. В процессе обучения сеть KAN параметризует уравнение (1). Слой сети KAN с n_{in} входами и выходами n_{out} определяется с помощью функциональной матрицы, имеющей вид $\Theta = \{\theta_{q,p}\}, p = 1,2, \dots, n_{in}, q = 1,2, \dots, n_{out}$.

Типичная архитектура сети KAN представляет собой композицию двух слоев KAN, показанную на рис. 1. Каждая функция активации $\theta_{q,p}(t)$, по аналогии с многослойным

перцептроном, является суммой базисной функции $b(t) = t/(1 + e^{-t})$ и сплайновой функции (n $spline(t) = \sum_i c_i B_i(t)$ ($B_i(t)$ — B-сплайны, c_i — параметры обучения):

$$\theta_{q,p}(t) = \upsilon_q b(t) + \upsilon_p spline(t), \upsilon_b, \upsilon_s \text{ — параметры обучения.}$$

Глубокая сеть KAN представляет собой композицию слоев KAN.

$$Y = (\Theta_{L-1} \circ \Theta_{L-2} \circ \ldots \circ \Theta_1 \circ \Theta_0)T.$$

В предложенном методе используется архитектура KAN, представленная на рис. 2. Выбран кубический сплайн и пять интервалов сетки.

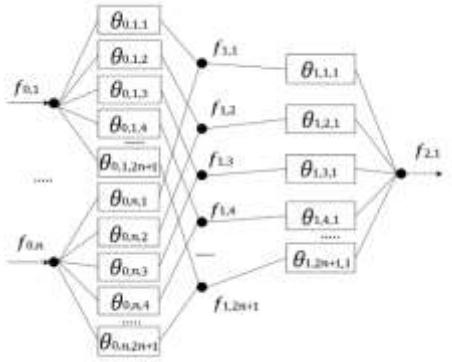


Рис. 1. Типичная архитектура НС KAN.

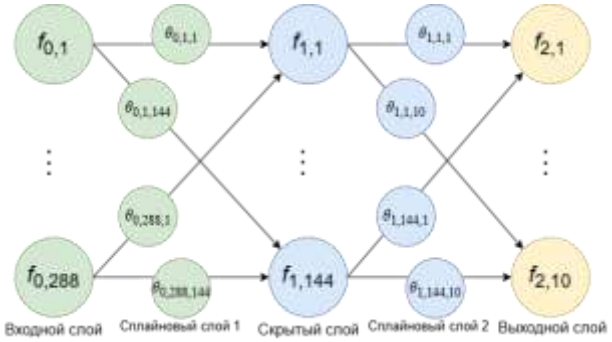


Рис. 2. Архитектура НС KAN.

Вейвлет-фильтрация основана на вейвлете Добеши 3-го порядка и жесткой пороговой функции. Схема обнаружения аномалий в данных нейтронного монитора представлена на рис. 3.

Рис. 3. Схема программы для обнаружения природных аномалий.

Результаты

Оценки регрессии модели представлены в таблице 1.

Таблица 1

Метрики регрессии

Метрики / Эксперимент	MSE	MAE	RMSE	MAPE, %	MASE	L1/L2
Тестовые данные	0.060	0.245	0.245	0.15	0.287	1171.635 / 20.572
Случайный участок тестовых данных на 60 точек	0.115	0.214	0.340	0.19	0.304	12.843 / 2.630
Наименее благоприятный случай (нефильтрованные, неизвестные данные неизвестного диапазона)	1.804	1.072	1.343	1.09	1.055	309.725 / 22.833

Разработанная модель обладает высокой точностью и стабильностью на данных тестовой выборки, а также сохраняет приемлемое качество на отдельных участках выборки. При переходе к наименее благоприятному сценарию: нефильтрованным данным, выходящим за пределы обучающего распределения – качество прогнозирования может существенно снижаться. Данная проблема решаема улучшенным обучением нейронной сети и улучшением архитектуры за счет больших вычислительных мощностей, а также компенсируется модулем, определяющим, когда реальное значение считается аномалией в сравнении с предсказанным. Например, увеличение размера скрытого слоя повышает точность.

Обнаружение аномалий производилось в экстремальных условиях: на данных приближенных к неблагоприятному случаю метрики регрессии. Оценки обнаружения аномалий представлены в таблице 2.

Таблица 2

Метрики обнаружения аномалий

Эксперимент / Метрики	GLE	GLE (<%)	GLE (>%)	FD	FD (до min)	FD (<%)
Покрытие индексов (Recall), %	100.00	100.00	100.00	71.91	76.12	83.15
Ложные срабатывания, %	4.88	17.07	0	4.89	4.89	19.56
Сдвиги начала	0	0	0	2	2	0
Precision, %	97.55	91.92	100.00	85.33	82.26	62.71
F1-score, %	98.76	95.79	100.00	78.05	79.07	71.50
IoU, %	97.55	91.92	100.00	64.00	65.38	55.64

При экстремальных условиях метод обнаружения показал внушительный результат сравнимый с аналогами, когда аномалии проявляют свои особенности, затрудняющие работы их обнаружения, а поступающие данные являются наименее благоприятным случаем для модели, и даже превосходящие результаты при аномалии без проблемных особенностей. Нужно учитывать, что в обнаружении участвует как модель, которая предсказывает значения по реальным значения, так и модуль, который решает, когда реальное значение отклоняется от предсказанного в категории аномалии, поэтому модель и модуль дополняют друг друга. Процент отклонения может компенсировать предсказания модели, но он связан с ложными срабатываниями. Эффективность достигается за счет совместной работы модели прогнозирования, фильтрации и модуля обнаружения аномалий, при этом настройка порога отклонения представляет собой компромисс между полнотой обнаружения и уровнем ложных срабатываний, а регрессия модели может стать точнее кроме фильтрации за счет высоких технических мощностей, требуемых для более точного обучения.

В ходе выполнения данной работы реализован программный комплекс, который позволяет обучать, тестировать, валидировать модель нейронной сети KAN, обрабатывать и визуализировать данные, а графическое приложение позволяет работать в режиме с файлом данных нейтронного монитора или в режиме реального времени с нейтронным монитором. Пример использования приложения в реальной ситуации с поступлением данных нейтронного монитора в режиме реального времени посредством встроенной имитации на неизученном диапазоне представлен на рис. 4. При обнаружении аномалии

пользователю выводится мигающее сообщение, которое информирует об обнаружении аномалии и будет мигать до тех пор, пока пользователь не нажмет кнопку «Зафиксировать аномалию», а потом подтвердит, что хочет зафиксировать. Мигающее сообщение появится вновь, если будет зафиксирована новая аномалия.

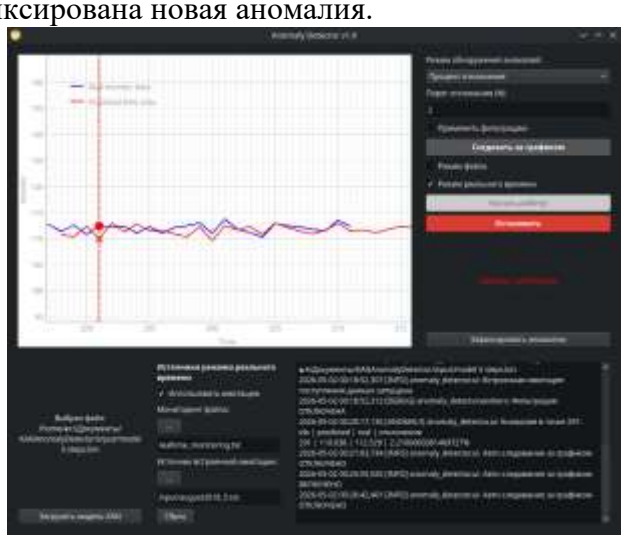


Рис. 4. Работа приложения в режиме реального времени.

Разработанный метод, обученная модель и программный комплекс могут применяться в научно-исследовательских организациях и центрах мониторинга космической погоды, а также в прикладных областях, связанных с эксплуатацией спутниковых, навигационных и других технических систем, подверженных влиянию космических факторов. Ввиду модульной структуры, в дальнейшем решение может улучшаться за счет улучшения модели и соответственно метрик регрессии, добавления других режимов в модуль обнаружения аномалий помимо процента отклонения, расширение модуля фильтрации на другие типы фильтрации, а также развития программного комплекса в целом, представляющего полноценную среду для практической работы с решением.

Доступность кода

Исходный код системы доступен по ссылке: https://github.com/proitshnik/KANAnomalyDetector.

Список литературы

1. Geppener V., Mandrikova B. Automated method for cosmic ray data analysis and detection of sporadic effects //Computational Mathematics And Mathematical Physics. 2021. Т. 61. № 7. С. 1129-1139.
2. Belov A, Eroshenko E, Yanke V, Oleneva V, Abunina M, Abunin A. Method of global survey for the world network of neutron monitors // GaA 2018; 58(3): 374-389. DOI: 10.7868/S0016794018030082.
3. Щербулова Е.Е. Автоматическая фильтрация ложных срабатываний в системе gle-оповещения на фоне Форбуш-эффектов // Sciences of Europe, no. 166, 2025, pp. 42-46.

АДАПТИВНЫЙ ПОРОГ СЕМАНТИЧЕСКОЙ БЛИЗОСТИ В ГИБРИДНОЙ СИСТЕМЕ КЛАССИФИКАЦИИ НАМЕРЕНИЙ С ПРИВЛЕЧЕНИЕМ ЯЗЫКОВОЙ МОДЕЛИ

Котов И. И., Куприянова А. М., Куприянов Н. М.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. В статье представлен метод адаптивного порога семантической близости для гибридной системы классификации намерений, объединяющей векторный поиск на основе эмбеддинг-модели и языковую модель (LLM) в роли резервного классификатора. Предложенный подход позволяет динамически корректировать порог срабатывания на основе статистических характеристик распределения косинусных сходств, снижая долю запросов, направляемых к языковой модели, при сохранении высокой точности распознавания намерений пользователя.

Ключевые слова: классификация намерений, семантическая близость, языковая модель, гибридная система, адаптивный порог, векторный поиск, косинусное сходство, диалоговые системы.

Введение

Задача классификации намерений занимает центральное место в архитектуре современных диалоговых систем и чат-ботов. Традиционные подходы, основанные исключительно на правилах или статистических моделях, всё больше уступают место гибридным решениям, сочетающим быстрый векторный поиск с мощными языковыми моделями. Ключевой проблемой подобных систем остается выбор порогового значения “уверенности” модели в намерении: заниженный порог влечет избыточную нагрузку на LLM, а завышенный — приводит к ошибочным определениям намерений. [1, 2] Нужен адаптивный механизм, который корректирует порог в режиме реального времени, опираясь на актуальную статистику отработанных действий. В данной статье приводится такой механизм с его оценкой на открытом наборе данных о намерениях.

Архитектура гибридной системы классификации намерений

Предлагаемая система состоит из трех взаимодействующих компонентов. На первом уровне входящий запрос пользователя преобразуется в вектор с помощью эмбеддинг-модели (например, paraphrase-multilingual-MiniLM-L12-v2). [7] Полученный вектор сопоставляется с индексом эталонных фраз намерений посредством приближенного поиска ближайших соседей. [3]

На втором уровне принятия решения вычисленное косинусное сходство сравнивается с адаптивным порогом Θ. Если сходство превышает порог, намерение считается распознанным и результат возвращается немедленно. В противном случае запрос передается на третий уровень — языковой модели, которая обрабатывает неоднозначный запрос по системному и пользовательскому промптам, включающим в себя список допустимых классов намерений, а также текущий контекст диалога. [4]

Подобная маршрутизация позволяет обрабатывать типичные запросы с минимальной задержкой (2–5 мс на эмбеддинг и поиск), используя легкую эмбеддинг-модель, оставляя более дорогостоящие по времени и ресурсам вызовы LLM для семантически неоднозначных случаев.

Метод адаптивного порога семантической близости

Ключевая идея метода состоит в непрерывном отслеживании распределения максимальных косинусных сходств по скользящему окну из N последних запросов. На каждом шаге вычисляются среднее значение косинусного сходства μ и стандартное отклонение σ выборки:

$\Theta(t) = \mu(t) - \alpha \cdot \sigma(t)$, где α — настраиваемый гиперпараметр чувствительности ($0.5 \leq \alpha \leq 2.0$). Малые значения α соответствуют строгому порогу и большему количеству LLM-вызовов, большие — мягкому порогу и уменьшению числа запросов к языковой модели. Приближенно нормальное распределение сходств позволяет интерпретировать такой порог

как параметрический квантиль: например, при α = 1.0 около 16% наблюдений оказывается ниже порога и направляется к языковой модели. Таким образом, α напрямую задаёт ожидаемую долю LLM-вызовов, а выбранный диапазон покрывает практически оправданные режимы — от умеренной фильтрации (α = 0.5, 32% запросов к LLM) до жёсткой (α = 2.0, 2.8% запросов).

Порог ограничен снизу и сверху гарантийными значениями Θ_min = 0.60 и Θ_max = 0.95 для исключения вырожденных режимов работы. Нижняя граница предотвращает падение порога в область, где косинусное сходство уже не обеспечивает уверенного распознавания намерения. Верхняя граница, напротив, не позволяет порогу приближаться к 1.0 в узких тематиках, что иначе приводило бы к почти полному отключению векторного поиска и избыточной нагрузке на LLM.

Алгоритм функционирует в трёх режимах в зависимости от заполнения скользящего окна:

а) Холодный старт (менее 10 запросов): используется статически заданный порог Θ_0 = 0,80. Это консервативный выбор, пока данных очень мало: в начале работы лучше лишний раз вызвать LLM, чем ошибиться.

б) Разогрев (10–500 запросов): порог рассчитывается по текущей выборке без полного скользящего окна. Данных уже достаточно для статистики, но окно ещё не заполнено. Формула работает, но на неполной выборке — оценки μ и σ менее стабильны.

в) Рабочий режим (более 500 запросов): полноценная формула с экспоненциальным сглаживанием для устойчивости к выбросам.

Экспериментальная оценка

Оценка проводилась на наборе данных CLINC150 [5], содержащем 150 классов намерений и 22 500 обучающих образцов. Тестовая выборка (4 500 примеров) разбита на три временны́х сегмента для имитации тематического дрейфа: банковская тематика, туризм и общие бытовые запросы. Базовой линией служила система с фиксированным порогом Θ = 0,80 — значение, рекомендованное в ряде промышленных решений. [6]

На рисунке 1 продемонстрирован сам механизм: фиксированный порог не реагирует на смену тематики, а адаптивный Θ(t) опускается в сегменте B «Туризм», следуя за сдвигом распределения.

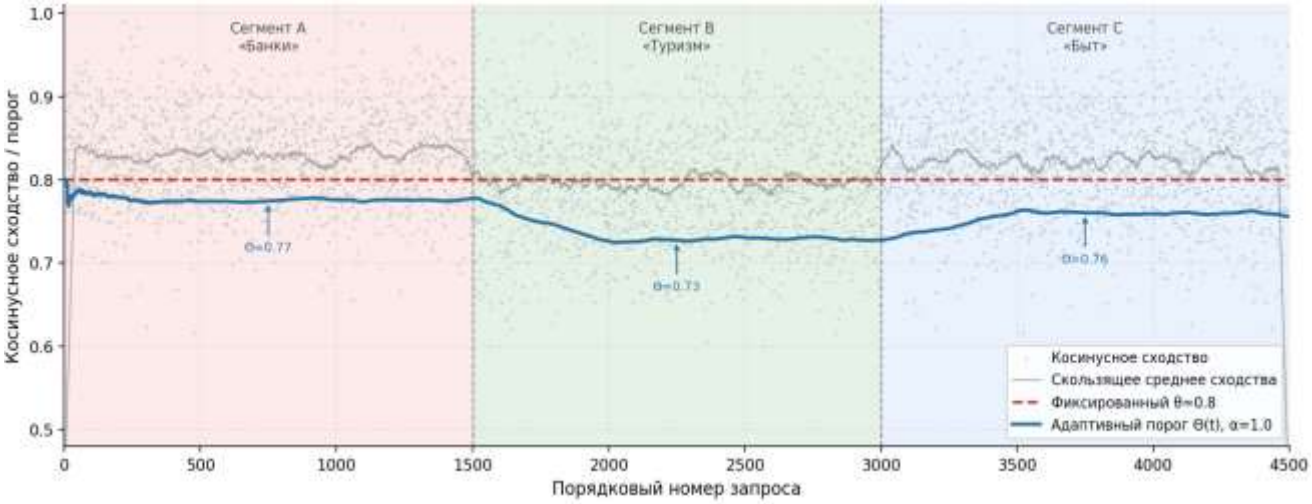


Рис. 1. Динамика адаптивного порога

Результаты показали следующие преимущества адаптивного порога по сравнению с фиксированным:

1. Снижение доли LLM-вызовов. На рисунке 2 видно, что при α = 1.0 доля LLM-вызовов падает с 39.2% до 16.0%, а точность при этом растёт до 93.1%.

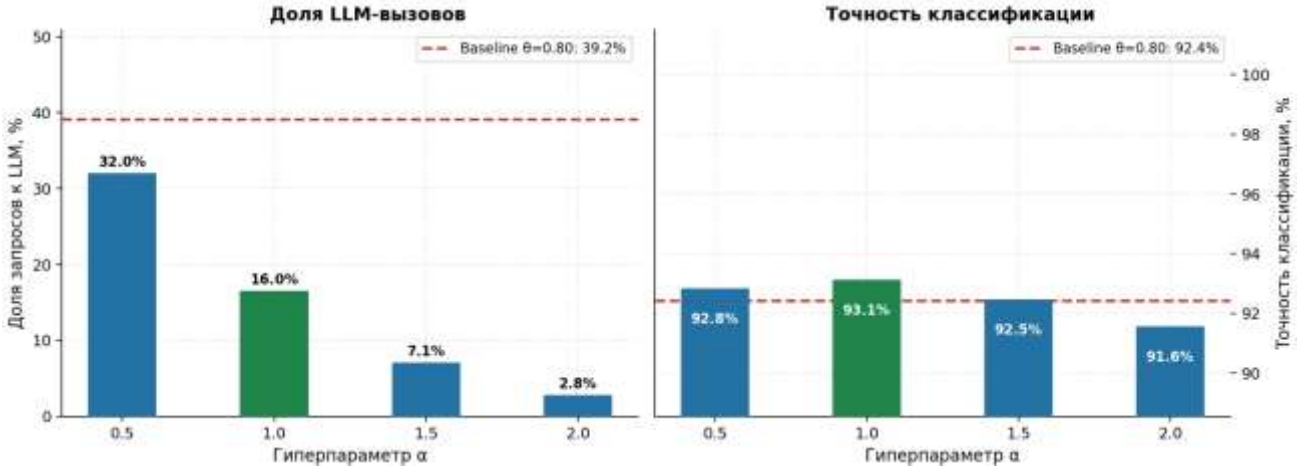


Рис. 2. Компромисс LLM и точность

2. Устойчивость к тематическому дрейфу. Левая часть рисунка 3: фиксированный порог в сегменте B проседает. Правая часть: фиксированный порог в сегменте B отправляет 53.1% запросов к LLM, а адаптивный — лишь 19.1%. Также видно, что при смене тематики фиксированный порог «не замечает», что сходства изменились, и начинает ошибаться. Адаптивный, в свою очередь, подстраивается и сохраняет качество.

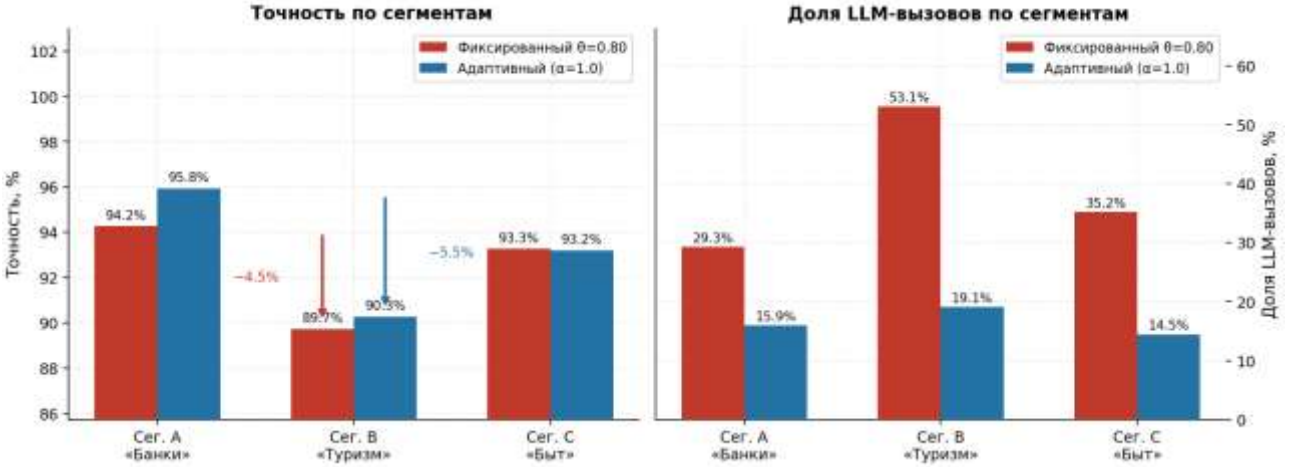


Рис. 3. Устойчивость к тематическому дрейфу

3. Сокращение задержки. Среднее время ответа системы снизилось с 87 мс до 53 мс за счет уменьшения числа обращений к API языковой модели.

Заключение

Предложенный метод адаптивного порога семантической близости доказал свою эффективность в условиях тематического дрейфа входящих запросов. Интеграция механизма в гибридную систему классификации намерений позволила сократить число обращений к языковой модели почти вдвое при незначительном изменении метрики

точности. Гиперпараметр α обеспечивает гибкое управление компромиссом между экономией ресурсов и качеством классификации, что делает подход применимым в широком спектре производственных диалоговых систем. Направлением дальнейших исследований является разработка алгоритма автоматической настройки α на основе целевых ограничений на бюджет LLM-запросов.

Список литературы

1. Liu X., Eshghi A., Swietojanski P., Rieser V. Benchmarking Natural Language Understanding Services for Building Conversational Agents [Электронный ресурс] // Hugging Face. URL: https://huggingface.co/papers/1903.05566 (дата обращения: 10.02.2026).
2. Devlin J., Chang M., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [Электронный ресурс] // arXiv. URL: https://arxiv.org/abs/1810.04805 (дата обращения: 12.02.2026).
3. Zhang Z., Han X., Liu Z. et al. ERNIE: Enhanced Language Representation with Informative Entities [Электронный ресурс] // arXiv. URL: https://arxiv.org/abs/1905.07129 (дата обращения: 20.02.2026)..
4. Hendrycks D., Gimpel K. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks [Электронный ресурс] // arXiv. URL: https://arxiv.org/abs/1610.02136 (дата обращения: 25.02.2026).
5. Larson S., Mahendran A., Peper J. J. et al. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction [Электронный ресурс] // Hugging Face. URL: https://huggingface.co/papers/1909.02027 (дата обращения: 07.03.2026).
6. Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks [Электронный ресурс] // Hugging Face. URL: https://huggingface.co/papers/1908.10084 (дата обращения: 10.03.2026).
7. sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 [Электронный ресурс] // Hugging Face. URL: https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2 (дата обращения: 10.03.2026).

ПРИМЕНЕНИЕ МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ПЕРСОНАЛИЗАЦИИ ОБРАЗОВАТЕЛЬНЫХ ТРАЕКТОРИЙ В ВЫСШЕЙ ШКОЛЕ

КРЕКИНА С.А.

Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

Аннотация. Статья посвящена использованию искусственного интеллекта (ИИ) для построения индивидуальных образовательных маршрутов студентов. Показано, что традиционная система обучения не позволяет в полной мере учитывать различия в темпе, стиле и начальной подготовке учащихся. Авторами предлагается архитектура адаптивной системы на основе методов машинного обучения, классификации студентов и интеллектуальных рекомендаций.

Ключевые слова: искусственный интеллект, персонализация, образовательная траектория, машинное обучение, высшая школа, адаптивное обучение.

Введение

Современная система высшего образования характеризуется противоречием между декларируемым принципом индивидуального подхода и реальной практикой унифицированного построения учебного процесса. Традиционная модель предполагает единый темп изучения дисциплины, стандартный набор заданий и линейную последовательность освоения тем, что не позволяет в полной мере учитывать различия в когнитивных стилях, уровне предварительной подготовки и психофизиологических особенностях обучающихся. Преподаватель, работающий с потоком численностью 100–200

человек, объективно не способен осуществлять непрерывный мониторинг прогресса каждого студента и оперативно корректировать его траекторию.

Одним из перспективных направлений преодоления указанного противоречия является применение методов искусственного интеллекта (ИИ). Технологии ИИ позволяют автоматизировать сбор и анализ цифрового следа студента (частота обращений к электронной образовательной среде, временные затраты на выполнение заданий, характер ошибок и др.), на основе которого формируется профиль обучающегося и осуществляется динамическая подстройка содержания образовательного курса.

Цель настоящей работы — систематизация методов ИИ, применимых для персонализации образовательных траекторий в условиях массового высшего образования, и представление архитектуры адаптивной системы, реализующей указанные методы.

Методы искусственного интеллекта для персонификации образовательного процесса

На основе анализа практики функционирования электронных образовательных сред и специализированных адаптивных платформ могут быть выделены четыре группы методов ИИ, наиболее релевантных для построения индивидуальных образовательных траекторий.

Прогнозирование успеваемости. Алгоритмы анализируют поведение студента в течении первых недель обучения и предсказывают, кто рискует не сдать сессию. Например, если студент редко заходит в электронную среду или делает много однотипных ошибок, система выдаёт сигнал куратору. Это позволяет вмешаться заранее и скорректировать процесс обучения для данного студента.

Разделение студентов на группы. Студенты даже в одной группе различаются по уровню знаний, умений, навыков, мотивации, темпу обучения и психо-физиологическим параметрам. Алгоритмы автоматически распределяют учащихся по небольшим группам со сходными стилями восприятия и темпами работы. Для каждой группы система подбирает подходящий формат заданий и объяснений.

Адаптивная последовательность изучения материала. В обычном курсе темы идут строго по порядку. ИИ может менять последовательность обучения, увеличивая или уменьшая количество основных и дополнительных заданий и отработок.

Интеллектуальные рекомендации. ИИ рекомендует студенту дополнительные курсы, книги, видеолекции или даже выбор элективов на следующий семестр. Рекомендация строится на основе успехов студента и предпочтений похожих на него учащихся.

Модель системы персонализации

В качестве первого подхода к построению адаптивной системы обучения на основе искусственного интеллекта может быть предложена четырёхуровневая модель. Такая модель является универсальной и может быть внедрена в классический образовательный процесс любого вуза, где используется электронная образовательная среда (например, Moodle или аналоги).

Рисунок 1 – Модель системы

Первый уровень – сбор данных.

Система фиксирует: дату и время входа, длительность работы с каждой лекцией, результаты тестов, число попыток решения задач, использование подсказок. Никакие личные данные (ФИО, группа) не передаются — всё обезличивается.

Второй уровень – профиль студента.

На основе собранных данных программа определяет:

- текущий уровень знаний по предмету;
- темп усвоения (быстрый, средний, медленный);
- предпочтения по формату заданий.

Также вычисляется «зона риска» – вероятность получить неудовлетворительную оценку, если ничего не менять.

Третий уровень – подбор следующего действия.

Система решает, что дать студенту прямо сейчас: новую лекцию, тренировочный тест, дополнительное объяснение сложного места или продвинутую задачу. Решение принимается на основе математической модели, которая стремится максимизировать рост знаний и при этом не перегружать студента.

Четвёртый уровень – обратная связь.

Система показывает студенту, почему она предложила именно это задание. Также студент может оценить рекомендацию, это помогает системе самообучаться.

Внедрение системы персонализации в институте

Внедрение ИИ в реальный учебный процесс сталкивается с несколькими трудностями:

- технические: обработка данных 1000 студентов требует мощных серверов (несколько современных компьютеров с хорошими видеокартами). Кроме того, нужны программисты, которые настроят систему
- методические: не все преподаватели готовы доверять компьютеру. Многие справедливо замечают, что алгоритм может ошибиться или «загнать» студента в слишком узкую колею, где тот никогда не встретится со сложными, но важными темами.
- этические: сбор данных о поведении студента может восприниматься как слежка. Нужна прозрачность: студент должен знать, какие данные собираются, и иметь возможность отказаться (например, выбрать режим только с оценками, без отслеживания стиля работы).

На основе проведенного анализа в Муромском институте ВлГУ предложен практический план из пяти шагов для внедрения системы персонализации в институте:

1. выбрать полигон, начать с одной дисциплины, где большой поток (более 100 человек) и чёткие измеримые результаты — например, «Информатика» или «Математика»;

2. настроить сбор данных, для этого обязать всех преподавателей использовать единую электронную среду, а также, настроить автоматический экспорт логов в хранилище;

3. разработать профили совместно с преподавателями кафедры, при этом достаточно выделить 2–3 типа студентов по темпу и стилю, обучить простую модель классификации.

4. провести пилот: одна группа учится по ИИ-рекомендациям, контрольная – по обычной программе, замерить результаты через семестр;

5. масштабировать: если пилот успешен, разворачивать систему на другие дисциплины. Одновременно создать «центр образовательной аналитики» из ИТ-специалистов и методистов.

Необходимо понимать, что в данном случае ИИ работает как помощник, а не как замена преподавателю, поэтому на первоначальном этапе все изменения траектории должны утверждаться преподавателем. В дальнейшем накопление данных позволит автоматизировать систему персонификации в рамках отдельных дисциплин.

Выводы

В результате проведенной работы можно сделать вывод, что методы искусственного интеллекта создают технологическую основу для персонализации обучения в условиях массового высшего образования без увеличения кадровой нагрузки на преподавателя.

Предложенная четырёхуровневая модель системы персонификации образовательного процесса в вузе обеспечивает замкнутый цикл управления образовательной траекторией и может быть реализована на базе существующих электронных сред типа Moodle. Ключевыми препятствиями для такого внедрения являются: недостаток вычислительных ресурсов и профильных специалистов, методическое сопротивление преподавателей, а также этические риски, связанные со сбором поведенческих данных.

Практическая реализация рекомендуется в формате пилотных проектов на потоковых дисциплинах с последующим масштабированием и обязательным сохранением решающей роли преподавателя в утверждении образовательной траектории.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ АДАПТАЦИИ ТРАНСФОРМЕРНЫХ МОДЕЛЕЙ ДЛЯ КЛАССИФИКАЦИИ ТЕКСТА

Куприянова А. М., Котов И. И., Куприянов Н.М.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. В работе представлена практическая оценка компромисса «качество-вычислительные затраты» при адаптации трансформерных моделей в задаче бинарной классификации текста. Рассматриваются три подхода: полное дообучение (Full Fine-Tuning), параметро-эффективный метод LoRA (Low-Rank Adaptation) и подход с замороженными параметрами (Frozen Baseline). Показано, что LoRA обеспечивает оптимальный баланс между качеством и вычислительными затратами, демонстрируя деградацию качества не более 2–3% относительно полного дообучения при сокращении числа обучаемых параметров на порядок.

Ключевые слова: трансформерные модели, fine-tuning, LoRA, машинное обучение, обработка текста, классификация, оптимизация моделей, параметро-эффективный подход, замороженная модель, frozen baseline.

Введение

Трансформерные модели являются ключевым инструментом в задачах обработки естественного языка, однако их адаптация к прикладным задачам сопряжена со значительными вычислительными затратами. [1] Полное дообучение обеспечивает высокое качество, но требует существенных ресурсов памяти и времени, что ограничивает его применение в условиях ограниченной вычислительной инфраструктуры.

В последние годы активно развиваются параметро-эффективные методы дообучения (Parameter-Efficient Fine-Tuning, PEFT), позволяющие адаптировать модели за счёт

обновления лишь небольшой доли параметров. Несмотря на широкое распространение таких подходов, их практическая эффективность в условиях ограниченных ресурсов и на задачах умеренного масштаба требует дополнительного анализа.

Целью данной работы является не просто сравнение методов адаптации трансформерных моделей, а практическая оценка компромисса «качество–вычислительные затраты» при использовании полного дообучения, LoRA и замороженного подхода. Особое внимание уделяется анализу применимости параметро-эффективных методов в сценариях с ограниченным доступом к вычислительным ресурсам.

Описание методов дообучения

В работе рассматриваются три стратегии адаптации предобученной трансформерной модели к задаче бинарной классификации текста, различающиеся по степени изменения параметров модели и вычислительной стоимости. Их сопоставление позволяет количественно оценить, в какой мере снижение числа обучаемых параметров влияет на качество модели и оправдывает ли себя с практической точки зрения.

Полное дообучение (Full Fine-Tuning)

При полном дообучении все параметры модели (θ) инициализируются весами предобученной модели и далее оптимизируются на целевой задаче. Цель обучения формулируется как минимизация эмпирического риска:

$$\mathcal{L}(\theta) = -1/N \sum [y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i)],$$

где y_i — истинная метка, $\hat{y}_i = f\theta(x_i)$ — предсказание модели, N — число примеров. В силу большого числа параметров (~125 млн у RoBERTa-base [2]) метод требует значительных вычислительных ресурсов, однако обычно обеспечивает наивысшее качество.

LoRA (Low-Rank Adaptation)

LoRA относится к параметро-эффективным методам (PEFT) и основана на гипотезе, что изменения весов при дообучении обладают низким внутренним рангом. [4] Для каждого обучаемого линейного слоя исходная матрица весов $W_0 \in \mathbb{R}^{(d\times k)}$ остаётся замороженной, а адаптивное возмущение представляется в виде произведения двух низкоранговых матриц:

$$W = W_0 + BA,$$

где $A \in \mathbb{R}^{(r\times k)}$, $B \in \mathbb{R}^{(d\times r)}$ и ранг $r \ll \min(d,k)$. В ходе обучения обновляются только матрицы A и B, что радикально сокращает число тренируемых параметров.

В экспериментах используются следующие параметры LoRA: ранг $r = 8$, коэффициент масштабирования $\alpha = 32$. Адаптеры применяются к слоям внимания (query, key, value).

Замороженная модель (Frozen Baseline)

Самый лёгкий с точки зрения ресурсов подход: вся предобученная основа, включая энкодер, остаётся неизменной, а обучению подвергается только классификационная голова — линейный слой, проецирующий выходной вектор (представление токена CLS) в логиты двух классов. Число обучаемых параметров не превышает 0,01% от общего объёма модели. Метод служит нижней границей качества, позволяя оценить, насколько хорошо предобученные представления решают задачу без адаптации внутренних слоёв.

Экспериментальная установка

Эксперименты проведены на датасете IMDB (бинарная классификация отзывов по тональности). [3] Использована случайная подвыборка из 50000 примеров, разделённая на

обучающую (40500 примеров) и тестовую (4500 примеров) выборки, а также валидационная (5000 примеров) для подбора гиперпараметров. Такое разбиение близко к типовому компромиссу: обучающая выборка остаётся достаточно большой для настройки 124 млн параметров, а тестовый набор обеспечивает статистическую значимость метрик и позволяет надежно сравнивать методы.

Базовая модель — RoBERTa-base (124,6 млн параметров). Аппаратное обеспечение: GPU NVIDIA A100, 40 ГБ.

Для каждого метода выполнено по 5 запусков с различными случайными начальными инициализациями seed для оценки стабильности метрик. Оптимизация — AdamW с learning rate (скоростью обучения) 2e–5 для полного дообучения и LoRA и 1e–3 для замороженной головы. Размер батча равен 16, обучение длится в течение 3 эпох. Измеряются метрики accuracy, F1-score, время обучения (в минутах) и количество обучаемых параметров.

Результаты исследования

Таблица 1 объединяет основные результаты исследования оценки качества и вычислительных затрат для каждого из методов.

Таблица 1.

Сравнение методов адаптации

Метод	Accuracy	F1-score	Обучаемые параметры	Время обучения (мин)
Full Fine-Tuning	0,943 ± 0,004	0,941 ± 0,005	124,6 млн	42,0 ± 1,1
LoRA	0,921 ± 0,006	0,918 ± 0,006	3,2 млн	18,2 ± 0,8
Frozen Baseline	0,781 ± 0,010	0,776 ± 0,011	0,5 млн	6,0 ± 0,3

Полное дообучение ожидаемо даёт наилучшие метрики, однако требует обновления всех параметров и наибольшего времени. LoRA демонстрирует незначительное снижение качества (всего ~2-3% accuracy) при сокращении числа обучаемых параметров в 39 раз и ускорении обучения в 2,3 раза. Наблюдаемое снижение качества LoRA по сравнению с полным дообучением, вероятно, обусловлено ограничением ранга адаптации, что снижает выразительную способность вносимых изменений весов модели. Замороженный подход значительно уступает по качеству (падение accuracy ~17,2% относительно полного дообучения), хотя обучается в 7 раз быстрее и требует лишь 0,5 млн параметров.

На рисунке 1 представлено соотношение качества и времени обучения для рассматриваемых методов. Видно, что LoRA занимает промежуточное положение, обеспечивая близкое к максимальному качество при существенно меньших вычислительных затратах.

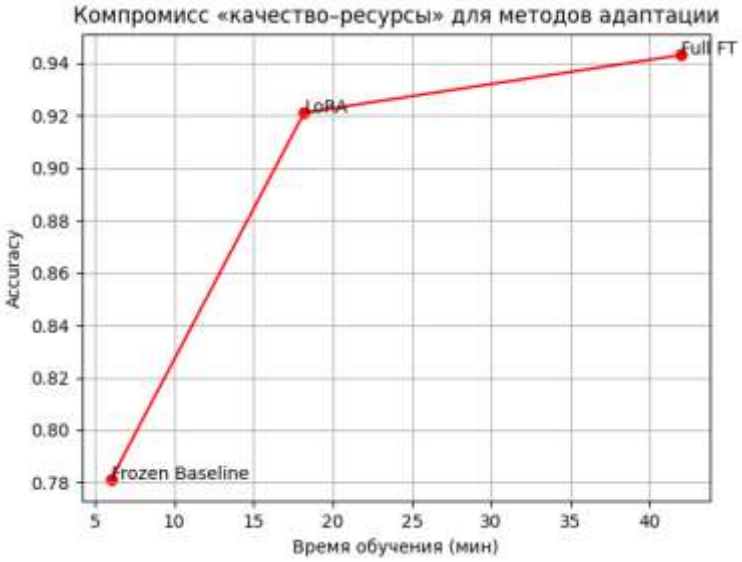


Рис. 1 Компромисс «качество–вычислительные затраты» для различных методов адаптации трансформерных моделей

Полученные результаты согласуются с современными представлениями о параметро-эффективном дообучении. LoRA достигает почти эталонного качества, внося модификации только в низкоранговые подпространства весов, в то время как полное дообучение избыточно для многих задач классификации, а замороженный подход недостаточно гибок.

Следует отметить, что все эксперименты выполнены в контролируемых условиях на одном датасете. Для общности выводов требуется валидация на других задачах и моделях. Кроме того, выбор ранга LoRA и множества адаптируемых слоёв может влиять на баланс, что является направлением дальнейших исследований.

Заключение

Проведённое экспериментальное сравнение трёх стратегий адаптации RoBERTa-base на задаче бинарной классификации тональности показало, что метод LoRA достигает наилучшего компромисса между точностью и вычислительными издержками. Отставание от полностью дообученной модели не превышает 2–3 процентных пунктов по accuracy и F1-score, тогда как объём модифицируемых параметров уменьшается почти в 40 раз, а длительность обучения — в 2,3 раза.

Полное дообучение по-прежнему фиксирует верхнюю планку качества, а «замороженный» режим, будучи самым легковесным, не обеспечивает приемлемой точности и может служить лишь простейшей базовой линией.

Таким образом, в условиях ограниченных вычислительных ресурсов LoRA является наиболее прагматичным выбором, позволяющим получать качественные классификаторы без необходимости обновлять сотни миллионов весов.

Список литературы

1. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., Polosukhin I. Attention is All You Need [Электронный ресурс] // Advances in Neural Information Processing Systems. 2017. URL: https://arxiv.org/abs/1706.03762 (дата обращения: 17.04.2026).

2. Liu Y., Ott M., Goyal N., Du J., Joshi M., Chen D., Levy O., Lewis M., Zettlemoyer L., Stoyanov V. RoBERTa: A Robustly Optimized BERT Pretraining Approach [Электронный ресурс] // arXiv preprint arXiv:1907.11692. 2019. URL: https://arxiv.org/abs/1907.11692 (дата обращения: 28.04.2026).

3. Maas A. L., Daly R. E., Pham P. T., Huang D., Ng A. Y., Potts C. Learning Word Vectors for Sentiment Analysis [Электронный ресурс] // Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. 2011. URL: https://aclanthology.org/P11-1015 (дата обращения: 01.05.2026).

4. Hu E. J., Shen Y., Wallis P., Allen-Zhu Z., Li Y., Wang S., Chen W. LoRA: Low-Rank Adaptation of Large Language Models [Электронный ресурс] // International Conference on Learning Representations (ICLR). 2022. URL: https://arxiv.org/abs/2106.09685 (дата обращения: 13.04.2026).

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В АВТОМАТИЗИРОВАННОМ ОТБОРЕ РЕЗЮМЕ: АРХИТЕКТУРА, ВОЗМОЖНОСТИ И ОГРАНИЧЕНИЯ

Купцов А.Р.

СПбГЭТУ «ЛЭТИ»

Аннотация. Рассмотрены системы автоматизированного отбора резюме, в которых применяются методы искусственного интеллекта. Показано, что такая ATS объединяет парсинг, нормализацию, семантическое сопоставление, скоринг и ранжирование. Цель работы - описать архитектуру системы и ее ограничения в прикладной области HR. Методика основана на анализе научных публикаций, отраслевых отчетов и нормативных документов. Результатом стала девятиэтапная схема обработки резюме. Выделены риски: ошибки парсинга, потеря контекста, жесткие фильтры, смещение данных и слабая объяснимость. Предложены меры контроля для разработчиков и рекрутеров.

Ключевые слова: искусственный интеллект, ATS, автоматизированный отбор, рекрутинг, машинное обучение, семантическое сопоставление, алгоритмическая справедливость.

1. Введение

Искусственный интеллект все чаще применяется в HR-процессах. В отборе резюме он помогает искать совпадения между вакансией и профилем кандидата. Такая задача хорошо подходит для автоматизации. Система работает с текстом, анкетами и структурированными полями.

При этом ИИ в рекрутинге не принимает решение в пустоте. Он зависит от качества резюме, словаря навыков, правил фильтрации и истории найма. Ошибка на раннем этапе переходит в итоговый рейтинг. Поэтому систему нельзя оценивать только по скорости.

Цель статьи - показать архитектуру ИИ-ориентированной ATS и ограничения ее применения. Методика включает анализ работ по информационному поиску, алгоритмическому найму и управлению рисками ИИ. Основной результат - схема обработки резюме. Ее можно использовать при аудите таких систем.

2. Эволюция и архитектура ATS

Первые ATS решали учетную задачу. Они фиксировали вакансию, отклик, источник и этап подбора. Затем появились интеграции с карьерными сайтами, почтой и внутренними HR-системами. Следующим этапом стали парсинг резюме, фильтрация, скоринг и аналитика.

ИИ изменил роль ATS. Система стала не только хранилищем откликов. Она начала выделять признаки кандидата, сравнивать их с вакансией и предлагать порядок просмотра. В современных решениях для этого применяются правила, модели машинного обучения и векторные представления текста.

Архитектура такой системы включает несколько модулей. Входной модуль принимает файлы и анкеты. Парсер извлекает текст и структуру документа. Нормализатор приводит

навыки, даты и должности к единому виду. Блок сопоставления сравнивает кандидата с вакансией. Блок скоринга и ранжирования готовит список для рекрутера.

С технической точки зрения ATS близка к системе информационного поиска. Описание вакансии можно рассматривать как запрос. Резюме можно рассматривать как документ. Методы ранжирования строят оценку соответствия и сортируют результаты[1]. В новых системах этот подход дополняется эмбеддингами и моделями машинного обучения[5].

3. Схема работы ATS

Обобщенная схема работы ATS включает девять этапов. Она описывает не конкретный продукт, а типовой конвейер обработки данных. Такая схема показывает, где ИИ дает пользу и где возникают риски.

Таблица 1

Типовой конвейер обработки резюме в ИИ-ориентированной ATS

Этап	Техническое действие	Контроль риска
1. Входные данные	Получение резюме, анкеты кандидата и описания вакансии.	Поддержка форматов и полнота полей.
2. Парсинг	Извлечение текста, контактов, опыта, навыков и образования.	Сравнение результата с исходным файлом.
3. Нормализация	Приведение дат, должностей, навыков и языков к единому виду.	Единый словарь навыков и синонимов.
4. Признаки	Формирование признаков: стаж, стек, отрасль, уровень, сертификаты.	Запрет лишних и косвенно дискриминирующих признаков.
5. Сопоставление	Сравнение резюме и вакансии по ключевым словам и векторам.	Проверка на тестовых парах резюме и вакансий.
6. Фильтрация	Отбор по обязательным требованиям и порогам.	Проверка причин отказа и доли потерь.
7. Скоринг	Расчет оценки на основе правил, признаков или модели.	Калибровка веса признаков и порогов.
8. Ранжирование	Сортировка кандидатов по оценке и приоритетам вакансии.	Проверка устойчивости итогового списка.
9. Вывод	Показ карточки, объяснений, статуса и рекомендации.	Решение человека и журнал действий.

4. Возможности и ограничения

Основная возможность ИИ в ATS - быстрый поиск релевантных резюме. Система может учитывать не только точные ключевые слова. Она может находить близкие формулировки и похожий опыт. Это полезно, когда кандидаты по-разному описывают один и тот же навык.

Главное ограничение связано с качеством входного документа. Резюме часто содержит таблицы, колонки, изображения и нестандартные заголовки. Парсер может потерять часть опыта или неверно связать дату с местом работы. После этого модель оценивает уже искаженный профиль.

Второе ограничение связано с контекстом. Векторные модели и эмбеддинги уменьшают зависимость от точных слов. Но они не всегда понимают смысл карьерного пути. Например, похожие названия должностей могут означать разные задачи. Поэтому такие модели требуют проверки на точность и устойчивость[5].

Третье ограничение связано с фильтрами. Жесткое правило может убрать подходящего кандидата. Частые причины - перерыв в опыте, отсутствие точного названия должности или нетипичная траектория обучения. Исследование Harvard Business School показывает, что часть работников становится скрытой для работодателей из-за практик отбора и применяемых технологий[4].

Четвертое ограничение связано со смещением данных. Если исторические решения были неравными, модель может воспроизвести это неравенство. Это касается целевых переменных, признаков и выборки обучения. Исследования по алгоритмическому найму отмечают, что заявления о снижении смещения требуют проверки, а не доверия по умолчанию[2]. Смещение может возникать не в одной точке, а на цепочке малых решений[3].

Пятое ограничение связано с объяснимостью. Рекрутер должен понимать, почему кандидат получил высокий или низкий балл. Кандидат также должен иметь возможность исправить ошибку в данных. Значимость этого вопроса подтверждается регулированием. В Регламенте ЕС 2024/1689 системы ИИ для подбора и оценки кандидатов отнесены к высокорисковым областям[7].

5. Рекомендации

Технические меры должны начинаться с контроля данных. Для каждого отклика нужно сохранять исходный файл, извлеченный текст и результат парсинга. Это помогает найти место ошибки. Также нужен список поддерживаемых форматов и проверка пустых полей.

Скоринг лучше строить как двухэтапный процесс. На первом этапе система должна искать кандидатов с высоким охватом. На втором этапе можно применять точные правила. Обязательные требования надо отделять от желательных. Это снижает риск случайного отказа.

Модель следует проверять на тестовой выборке. Нужны метрики качества, устойчивости и справедливости. Важно смотреть не только средний результат. Ошибки надо оценивать по группам вакансий, источникам откликов и типам резюме. Логи версий модели и правил должны храниться вместе с решением.

Организационные меры не менее важны. Автоматический рейтинг не должен быть единственным основанием отказа. Часть кандидатов ниже порога нужно просматривать вручную. Для сложных вакансий полезен второй эксперт. Рекрутеры должны знать, какие данные использует система.

Внутри компании нужен регулярный аудит ATS. Подход NIST AI RMF подходит для этого, так как связывает управление, картирование рисков, измерение и меры контроля[6]. Кандидатам также стоит давать понятную обратную связь. Полезна возможность исправить неверно прочитанные данные.

6. Заключение

ИИ в автоматизированном отборе резюме является прикладной технологией для HR. Он ускоряет обработку откликов и помогает работать с большим числом кандидатов. Но такая система не является нейтральным судьей. Она строит вывод на данных, правилах и моделях.

Предложенная схема показывает, что ошибка может появиться на любом этапе. Наиболее опасны ошибки парсинга, жесткие фильтры и смещение исторических данных. Поэтому автоматизация должна сочетаться с человеческой оценкой. Лучший результат дает не замена рекрутера, а управляемая поддержка его решения.

Список литературы

1. Manning C.D., Raghavan P., Schutze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press, 2008. 544 p. URL: https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf.
2. Raghavan M., Barocas S., Kleinberg J., Levy K. Mitigating Bias in Algorithmic Hiring: Evaluating Claims and Practices // Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. New York: ACM, 2020. P. 469-481. DOI: 10.1145/3351095.3372828.
3. Rieke A., Bogen M. Help Wanted: An Examination of Hiring Algorithms, Equity, and Bias. Washington: Upturn, 2018. URL: https://www.upturn.org/work/help-wanted/.
4. Fuller J.B., Raman M., Sage-Gavin E., Hines K. Hidden Workers: Untapped Talent. Boston: Harvard Business School Project on Managing the Future of Work; Accenture, 2021. URL: https://www.hbs.edu/managing-the-future-of-work/Documents/research/hiddenworkers09032021.pdf
5. Bevara R.V.K., Mannuru N.R., Karedla S.P., Lund B., Xiao T., Pasem H., Dronavalli S.C., Rupeshkumar S. Resume2Vec: Transforming Applicant Tracking Systems with Intelligent Resume Embeddings for Precise Candidate Matching // Electronics. 2025. Vol. 14, No. 4. Article 794. DOI: 10.3390/electronics14040794.
6. National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework (AI RMF 1.0). Gaithersburg: NIST, 2023. DOI: 10.6028/NIST.AI.100-1.
7. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence. URL: https://eur-lex.europa.eu/eli/reg/2024/1689/oj/eng.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ АРХИТЕКТУР ГЛУБОКОГО ОБУЧЕНИЯ ДЛЯ СЕГМЕНТАЦИИ ПОДЖЕЛУДОЧНОЙ ЖЕЛЕЗЫ НА КТ-ИЗОБРАЖЕНИЯХ

КУШНЕРОВ Д.А., БЕНДЕРСКАЯ Е.Н.

Санкт-Петербургский политехнический университет Петра Великого,

Институт компьютерных наук и кибербезопасности,

Высшая школа компьютерных технологий и информационных систем,
г. Санкт-Петербург

Аннотация. В статье рассматривается задача автоматической сегментации поджелудочной железы на КТ-изображениях при ограниченном объеме размеченных данных. Проведено экспериментальное сравнение архитектур глубокого обучения на датасете NIH Pancreas-CT (82 КТ-исследования). Реализован экспериментальный конвейер на базе PyTorch и MONAI, включающий предобработку данных, патчевое обучение 3D-моделей, инференс скользящим окном и расчет метрик качества. Сравнивались Attention U-Net в 2D-режиме, а также 3D-модели BasicUNet, Attention U-Net, SegResNet и Swin UNETR. SegResNet обеспечивает наилучший результат по коэффициенту Дайса (DSC = 0,783), а Attention U-Net 3D демонстрирует наибольшую точность границ (HD95 = 15,01 мм). В условиях эксперимента SegResNet оказалась устойчивее Swin UNETR.

Ключевые слова: глубокое обучение, компьютерная томография, поджелудочная железа.

Автоматическая сегментация анатомических структур на компьютерных томограммах (КТ) используется для клинической диагностики и планирования лечения. Среди органов брюшной полости поджелудочная железа представляет наибольшую сложность для ее выделения на КТ ввиду малого объема (менее 0,5% от объема КТ-скана), высокой анатомической вариабельности и низкого контраста с окружающими тканями [1]. За последние годы в этой задаче применялись как сверточные сети (CNN), так и трансформерные модели (Transformer). CNN хорошо извлекают локальные признаки, тогда как трансформерные модели лучше учитывают глобальный контекст, но требуют большего объема данных и вычислительных ресурсов.

В работе проведено экспериментальное сравнение архитектур глубокого обучения на задаче сегментации поджелудочной железы на датасете NIH Pancreas-CT [6] (82 КТ-изображения) для определения наиболее эффективной архитектуры при ограниченном

объеме размеченных данных. Качество сегментации оценено по метрикам DSC, HD95, Recall и Precision.

U-Net [2] стала одной из базовых архитектур сегментации медицинских изображений благодаря симметричной структуре «кодировщик-декодировщик» с пропускными соединениями; BasicUNet (MONAI) реализует эту схему для 3D-данных. В Attention U-Net [3] блоки внимания фильтруют признаки в пропускных соединениях и направляют модель на области, относящиеся к целевому органу. Архитектура SegResNet продолжает развитие сверточного подхода и использует остаточные блоки. Такие блоки облегчают обучение глубоких сетей и позволяют устойчиво извлекать локальные пространственные признаки. Для задачи сегментации малого органа это важно, поскольку модель должна одновременно сохранять чувствительность к небольшой целевой области и подавлять ложные срабатывания на окружающих тканях.

Swin UNETR [4] объединяет Swin Transformer-кодировщик с CNN-декодировщиком. Кодировщик моделирует глобальные зависимости за счет самовнимания в сдвигаемых окнах, а декодировщик восстанавливает пространственную структуру и формирует маску сегментации. Такая схема позволяет учитывать связи между удаленными участками КТ-изображения при меньшей вычислительной сложности по сравнению с классическими трансформерными архитектурами.

Несмотря на успехи применения указанных архитектур на крупных мультиорганных датасетах (BTCV, MSD), их сравнительная эффективность в условиях ограниченной выборки (менее 100 сканов) и на задаче сегментации одного малого органа остается мало изученной. Настоящая работа направлена на сравнение моделей в указанных условиях.

Разработана экспериментальная схема на базе PyTorch и MONAI. Исходные КТ-изображения и бинарные маски были представлены в формате NumPy-массивов. Для каждого пациента формировалась пара «изображение – маска». Выборка была разделена по пациентам. Для обучения использовались 70 КТ-сканов, а 12 КТ-сканов были выделены для контрольной оценки качества. Такое разделение выполнялось на уровне полных КТ-исследований, а не отдельных срезов, поэтому срезы пациента не могли одновременно попасть в обучающую и контрольную часть.

На этапе предобработки выполнены ресемплинг к разрешению 1,5×1,5×2,0 мм и нормализация интенсивностей в окне мягких тканей [-160, 240] HU. Ресемплинг необходим для приведения исследований к одинаковому масштабу, так как исходные исследования могут отличаться толщиной срезов и размером пикселей. Нормализация в HU-окне мягких тканей позволяет выделить диапазон значений, наиболее информативный для анализа органов брюшной полости и одновременно снижает влияние нерелевантных значений, соответствующих воздуху, костным структурам и артефактам.

Выбор функции потерь DiceFocalLoss обусловлен выраженным классовым дисбалансом: объем поджелудочной железы составляет менее 0,5% от объема скана. Для дополнительной балансировки применен сэмплер RandCropByPosNegLabeld с равным соотношением позитивных и негативных патчей. Аугментация включала случайные отражения и аффинные преобразования.

Обучение 3D-моделей выполнялось не на полном КТ-исследовании, а на патчах размером 96×96×96 вокселей. Такой подход снижает требования к GPU-памяти и одновременно позволяет чаще предъявлять модели области, содержащие поджелудочную железу. Использование позитивных и негативных патчей особенно важно при сегментации

малого органа, поскольку при случайном выборе фрагментов большинство из них содержало бы только фон.

На первом этапе проведена 2D-сегментация (Attention U-Net), которая показала DSC = 0,015, что обосновало переход к объемной обработке и потребовало полной переработки пайплайна. В 3D-режиме обучены модели Attention U-Net, BasicUNet, SegResNet и Swin UNETR (до 30 эпох, AdamW, lr = 10^{-4}, смешанная точность AMP, NVIDIA GeForce RTX 5060 Ti 16 GB). Для эффективного использования GPU-памяти (16 ГБ) применены кэширование данных в оперативной памяти (CacheDataset) и инференс методом скользящего окна с перекрытием 0,5. Для автоматического сбора метрик реализован бенчмарк-скрипт с расчетом DSC, HD95, Precision и Recall на контрольной выборке из 12 пациентов. Результаты представлены в табл. 1.

Для оценки качества сегментации использовались четыре взаимодополняющие метрики: DSC, HD95, Recall и Precision. Коэффициент DSC характеризует степень пространственного перекрытия между предсказанной маской и экспертной разметкой. Метрика HD95 отражает геометрическую ошибку сегментации и показывает расстояние между границами предсказанной и истинной масок с учетом 95-го процентиля, что снижает влияние единичных выбросов. Recall показывает, какая доля реального объема поджелудочной железы была обнаружена моделью, а Precision отражает, какая часть области, отмеченной моделью как поджелудочная железа, действительно относится к ней. Такое сочетание метрик позволяет оценивать не только общее перекрытие масок, но и характер ошибок модели. Среди них можно выделить пропуски целевой области, избыточную сегментацию и неточность границ органа.

Таблица 1

Сравнительные результаты сегментации поджелудочной железы (mean ± std)

Модель	Тип	DSC	HD95 (мм)	Recall	Precision
Attention U-Net	2D	0,015 ± 0,040	54,91 ± 24,62	0,008 ± 0,022	0,164 ± 0,303
BasicUNet	3D	0,678 ± 0,114	184,80 ± 51,66	0,866 ± 0,079	0,568 ± 0,137
Attention U-Net	3D	0,742 ± 0,047	15,01 ± 13,52	0,912 ± 0,077	0,632 ± 0,075
Swin UNETR	3D	0,557 ± 0,238	172,22 ± 73,28	0,863 ± 0,080	0,450 ± 0,245
SegResNet	3D	0,783 ± 0,051	55,61 ± 63,38	0,807 ± 0,084	0,767 ± 0,057

В эксперименте с Attention U-Net в 2D-режиме получено крайне низкое качество сегментации (DSC = 0,015). Это связано с отсутствием межсрезового контекста, особенно важного при выделении малого органа с изменчивой формой.

При переходе к 3D-сегментации результаты существенно улучшились. Наилучшее значение DSC показала модель SegResNet (DSC = 0,783) при наибольшей точности (Precision = 0,767), что свидетельствует о низком уровне ложноположительных предсказаний. Attention U-Net 3D заняла второе место по DSC = 0,742 и достигла лучшего значения HD95 = 15,01, что указывает на высокую точность контуров сегментации. Данная модель также показала наивысшую полноту (Recall = 0,912), то есть наиболее полно покрывает объем поджелудочной железы.

Модель Swin UNETR продемонстрировала наиболее низкий DSC среди 3D-моделей (0,557), что, вероятнее всего, обусловлено малым объемом обучающей выборки, так как трансформерные архитектуры требуют значительно большего количества данных для обучения по сравнению с CNN [5]. Кроме того, высокая дисперсия метрик (σ(DSC) = 0,238) указывает на нестабильность модели на данном датасете.

BasicUNet (DSC = 0,678) без механизмов внимания ожидаемо уступил Attention U-Net и SegResNet, однако продемонстрировал высокую полноту (Recall = 0,866), что связано со склонностью модели к избыточной детекции при низкой точности (Precision = 0,568).

Сравнение Precision и Recall показывает, что высокая полнота не всегда соответствует высокому качеству сегментации. BasicUNet и Swin UNETR находят значительную часть целевого органа, однако одновременно формируют больше ложноположительных областей, что снижает Precision. Для клинически ориентированных задач такое поведение нежелательно, поскольку избыточная сегментация может искажать оценку объема органа и пространственное положение его границ.

Помимо средних значений, в проведенных экспериментах анализировался разброс метрик между пациентами. Высокое стандартное отклонение у Swin UNETR по DSC указывает на зависимость качества от конкретного случая и нестабильность при ограниченном числе обучающих примеров. У SegResNet разброс DSC ниже, что делает модель более устойчивой по интегральной метрике перекрытия. В то же время высокий разброс HD95 у SegResNet свидетельствует о том, что отдельные случаи могут сопровождаться локальными ошибками определения границы.

Для качественной проверки результатов была реализована 3D-визуализация предсказанной и эталонной масок. На основе бинарных объемов строилась поверхностная реконструкция, позволяющая визуально оценить совпадение формы органа, наличие пропущенных участков и ложноположительных фрагментов. Такой этап не заменяет количественные метрики, но дополняет их, так как позволяет интерпретировать причины роста HD95 и различия между DSC и Precision.

Выводы. В настоящей работе проведено сравнение CNN и трансформерных архитектур для сегментации поджелудочной железы в условиях ограниченной выборки (82 скана). SegResNet показала лучшие значения DSC (0,783) и Precision (0,767), поэтому в данной постановке эксперимента ее можно рассматривать как базовый вариант. Attention U-Net 3D точнее локализует границы органа по HD95 (15,01 мм). Анализ полученных результатов экспериментов показал, что 2D Attention U-Net существенно уступила 3D-моделям, что подтверждает важность учета межсрезового контекста при сегментации малых органов. Swin UNETR показала нестабильность результатов, характерную для трансформерных моделей при дефиците данных. Полученные результаты могут быть использованы при разработке вспомогательных систем предоперационного планирования и лучевой терапии. В качестве перспективы работы рассматривается ансамблирование наиболее эффективных архитектур.

Список литературы

1. Roth H.R., Lu L., Farag A., Shin H.-C., Liu J., Turkbey E.B., Summers R.M. DeepOrgan: Multi-level deep convolutional networks for automated pancreas segmentation / arXiv preprint arXiv:1506.06448. – 2015.
2. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation / MICCAI. – 2015. – P. 234–241.
3. Oktay O., Schlemper J., Le Folgoc L., Lee M., Heinrich M., Misawa K., Mori K., McDonagh S., Hammerla N.Y., Kainz B., Glocker B., Rueckert D. Attention U-Net: Learning Where to Look for the Pancreas / arXiv preprint arXiv:1804.03999. – 2018.

4. Hatamizadeh A., Nath V., Tang Y., Yang D., Roth H.R., Xu D. Swin UNETR: Brain Tumor Segmentation with Swin Transformers / BrainLes@MICCAI. – 2022. – P. 272–284.
5. Hatamizadeh A., Tang Y., Nath V., Yang D., Myronenko A., Landman B., Roth H.R., Xu D. UNETR: Transformers for 3D Medical Image Segmentation / WACV. – 2022. – P. 574–584.
6. Roth H.R., Farag A., Turkbey E.B., Lu L., Liu J., Summers R.M. Data from Pancreas-CT / The Cancer Imaging Archive. – 2016.
7. Myronenko A. 3D MRI Brain Tumor Segmentation Using Autoencoder Regularization / BrainLes@MICCAI. – 2019. – P. 311–320.

ПРИМЕНЕНИЕ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ БЕСПИЛОТНЫХ ЛЕТАТЕЛЬНЫХ АППАРАТОВ НА ОСНОВЕ РАДИОЧАСТОТНОГО АНАЛИЗА

Мякшинов[1] Н.А, Голубев., И.М. [1],. Трофимов[1] Ю.В

1. Федеральное государственное бюджетное образовательное учреждение высшего образования «Университет «Дубна»

Аннотация. В данной работе рассмотрен метод детекции БПЛА на основе радиочастотных сигналов. В статье предложен метод предварительной обработки сигналов и представления их в виде спектральной плотности мощности. Разработана модель 1D CNN для бинарной классификации сигналов. Работа направлена на исследование возможности применения машинного обучения для задачи детекции БПЛА.

Ключевые слова: детекция БПЛА, машинное обучение, метод Велча, I/Q сигналы, спектральная плотность мощности, радиочастотный анализ

Введение

Беспилотные летательные аппараты (БПЛА) представляют собой одну из серьезнейших проблем безопасности объектов и населения. Разработка методов обнаружения БПЛА и раннего предупреждения об их опасности является перспективным направлением для использования машинного обучения. Методы машинного обучения могут позволить обнаруживать БПЛА с нестандартными сигналами в условиях сильных помех [1].

Методы детекции БПЛА

Существует несколько методов детекции БПЛА, в которых могут применяться методы машинного обучения: радарный, акустический, оптический и радиочастотный (РЧ)[2]. РЧ-метод, был выбран для разработки системы обнаружения БПЛА из-за наличия достаточного количества записей сигналов для обучения модели. Основан на анализе радиосигналов управления и передачи данных, не реагирует на биологические объекты; эффективность и дальность зависят от характеристик передатчика БПЛА и условий распространения сигнала. Подход требует сложной обработки в условиях низкого отношения сигнал к шуму.

Подготовка данных

Для обучения был выбран датасет DroneRF [3]. В нем представлены записи сигналов работы трех различных моделей БПЛА (Parrot AR.Drone, Parrot Bebop, DJI Phantom 3) и записи фоновых сигналов (См. Рис. 1), при этом записей фоновых сигналов значительно меньше, чем записей сигналов дронов, поэтому возникает сильный дисбаланс классов. Сигналы записаны с помощью двух приемников, каждый из которых имеет максимальную

полосу пропускания 80 МГц, для покрытия всего диапазона 2.4 ГГц. Один приемник захватывает нижнюю половину диапазона частот, а второй - верхнюю половину, при этом диапазоны частот приемников частично перекрываются для того, чтобы два сигнала можно было объединить в один.[4].

Сначала I/Q (In-phase/Quadrature) сигналы верхнего и нижнего диапазонов обрабатываются отдельно. Происходит сегментация каждого временного сигнала, по 10000 отсчетов в каждом сегменте. Затем над I/Q сигналами выполняется оценка спектральной плотности мощности по методу Велча. После этого происходит нормализация путем вычитания среднего, затем происходит объединение сигналов с нижнего диапазона и верхнего в один. Далее мощность сигнала переводится в логарифмическую шкалу, после чего сигнал проходит фильтрацию по порогу мощности, происходит разметка данных.

Отдельно стоит остановиться на фильтрации. Она работает только с сигналами дронов и не обрабатывает фон. В исходных записях сигналов дронов есть отдельные участки, где нет сигнала БПЛА, и присутствуют только фоновые сигналы, или же сигналов нет вообще. Для правильной разметки данных и очистки датасета применяется пороговая фильтрация.

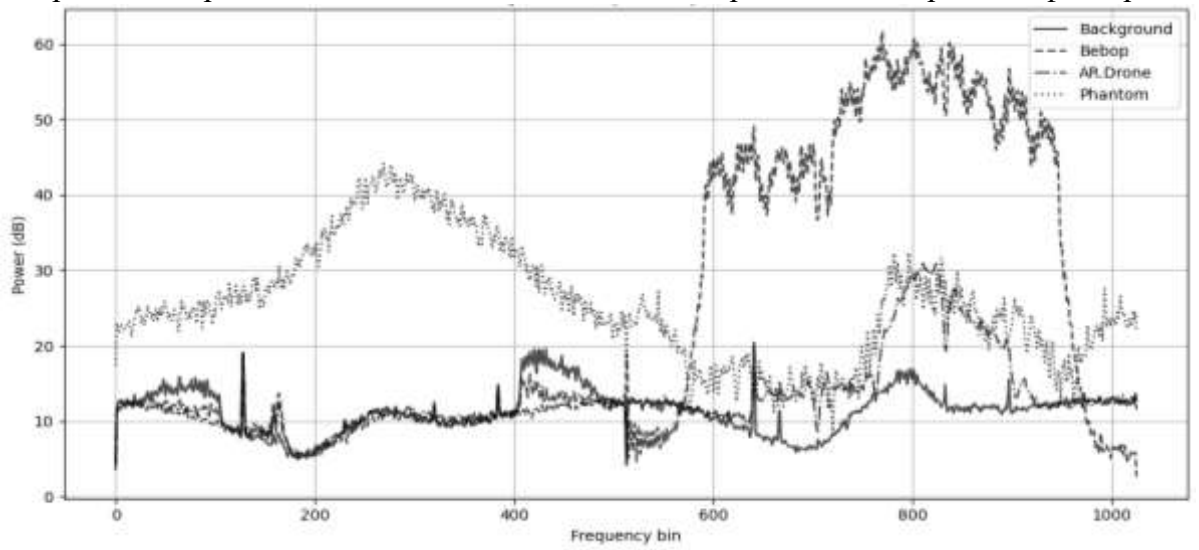


Рис.1 Сравнение фоновых сигналов и сигналов различных дронов

Тестовая модель

Разработанная модель имеет два сверточных слоя, пакетную нормализацию и два пулинга (См. Рис. 2). Основной проблемой является переобучение и сильный дисбаланс классов, записей сигналов фоновой активности в используем датасете очень мало по сравнению с сигналами БПЛА.

Параметры обучения были выбраны следующие: функция потерь – бинарная кросс-энтропия, оптимизатор – Adam, скорость обучения – 0.0005, количество эпох – 20, размер партии 64, вес фона был увеличен в 1.7 раз. Обычное разделение на обучающую и тестовую выборку не подходит из-за возможной утечки данных, поэтому используется метод разбиения данных, учитывающий групповую структуру исходных сигналов. Для корректной оценки обобщающей способности модели используется модифицированная процедура разбиения. Данный подход сочетает в себе принципы группового и стратифицированного разбиения.

На первом этапе формируется множество уникальных групп и определяется соответствие каждой группы множеству классов, присутствующих в ней. Далее

выполняется итеративный процесс случайного разбиения групп на обучающую и тестовую выборки с использованием фиксированного генератора случайных чисел.

На каждой итерации группы случайным образом перемешиваются и последовательно распределяются между обучающей и тестовой выборками с учётом заданной доли тестовой выборки (20%). После формирования разбиения проверяется выполнение следующих условий:

- все классы должны быть представлены в обучающей выборке;
- максимальное покрытие классов в тестовой выборке.

В случае, если найдено разбиение, удовлетворяющее указанным условиям, процесс останавливается. В противном случае выбирается наилучшее разбиение по критерию максимального разнообразия классов в тестовой выборке.

После определения наборов групп, формируются индексы обучающей и тестовой выборок таким образом, что все сегменты, принадлежащие одной конкретной записи сигала, целиком попадают либо в обучающую, либо в тестовую выборку. Это гарантирует отсутствие пересечения данных между выборками на уровне исходных сигналов

Сами сигналы от одного дрона при конкретном режиме работы (включен и подключен, зависание, полет, полет и передача видео) имеют очень схожую сигнатуру, так как конкретный дрон обычно использует одни и те же протоколы связи, поэтому для более точной модели нужно увеличивать число записей сигналов различных типов дронов, а не конкретного дрона. Архитектура модели несложная, ее можно использовать для портативных детекторов, но для более точной детекции необходимо усложнить архитектуру, например использовать трансферное обучение [5].

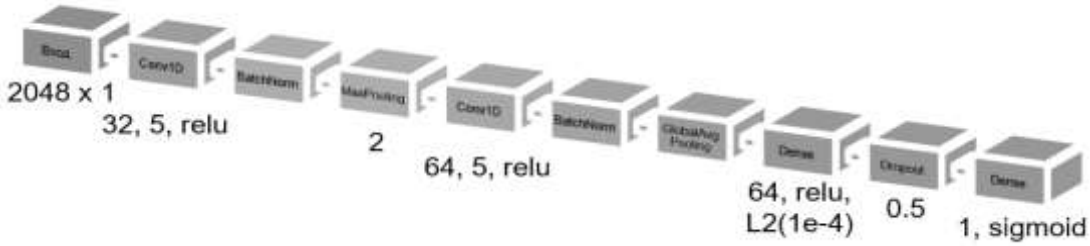


Рис.2 Архитектура модели

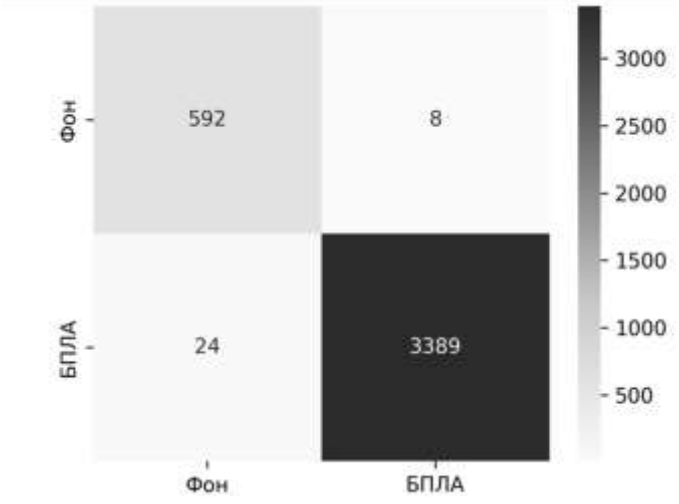


Рис.3 Матрица ошибок разработанной модели

Таблица 1

Общие метрики

Accuracy	Precision	Recall	F1	ROC AUC
0.9920	0.9976	0.9930	0.9953	0.9993

Таблица 2

Метрики по режимам работы и типам дронов

Класс	Количество образцов	Accuracy	Precision	Recall	F1	ROC AUC
Фон	600	0.007974	0.002355	0.013333	0.004003	0.000744
Bebop подключен	171	0.196113	0.050339	1.000000	0.095852	0.495105
Bebop зависание	442	0.262646	0.129526	0.995475	0.229226	0.609684
Bebop полет	254	0.215799	0.074183	0.992126	0.138044	0.701079
Bebop полет и видео	547	0.289808	0.161024	1.000000	0.277383	0.655172
AR подключен	379	0.247944	0.111569	1.000000	0.200742	0.577029
AR зависание	375	0.246449	0.110097	0.997333	0.198303	0.609787
AR полет	445	0.263892	0.130704	0.997753	0.231130	0.595416
AR полет и видео	434	0.253676	0.123050	0.963134	0.218220	0.520932
Phantom подключен	366	0.243708	0.107153	0.994536	0.193463	0.444869

Метрики показывают, что разработанная модель хорошо справляется с задачей бинарной классификации дронов (См. Таблица 1, Рис.3), но плохо различает сигналы по типам дронов и режимам работы (См. Таблица 2). Для фоновых сигналов метрики получились хуже всего, из-за сильного дисбаланса классов.

Заключение

Создан базовый бинарный классификатор дронов, предлагаемый метод позволяет детектировать дроны с FAR < 8% при SNR = –5 дБ, что подтверждено экспериментально. Данный классификатор можно использовать в комбинированных методах. Также был предложен метод предварительной обработки сигналов.

Список литературы

1. Semenyuk V., Kurmashev I., Lupidi A., Alyoshin D., Kurmasheva L., Cantelli-Forti A. Advances in UAV detection: integrating multi-sensor systems and AI for enhanced accuracy and efficiency // International Journal of Critical Infrastructure Protection. 2025. Vol. 49. P. 100744. DOI: 10.1016/j.ijcip.2025.100744.
2. Swinney C.J., Woods J.C. Unmanned Aerial Vehicle Operating Mode Classification Using Deep Residual Learning Feature Extraction // Aerospace. 2021. Vol. 8. P. 79. DOI: 10.3390/aerospace8030079.
3. Al-Sa'd M., Allahham M., Amr M., Al-Ali A., Tamer K., Erbad A. DroneRF dataset: a dataset of drones for RF-based detection, classification, and identification [Электронный ресурс] // Mendeley Data. – URL: https://data.mendeley.com/datasets/f4c2b4n755/1 (дата обращения: 29.03.2026).
4. Al-Sa'd M., Al-Ali A., Amr M., Tamer K., Erbad A. RF-based drone detection and identification using deep learning approaches: an initiative towards a large opensource drone database // Future Generation Computer Systems. 2019. P. 86-97. DOI: 10.1016/j.future.2019.05.007.
5. Swinney C.J., Woods J.C. Unmanned Aerial Vehicle Flight Mode Classification using Convolutional Neural Network and Transfer Learning // 16th International Computer Engineering Conference (ICENCO). Cairo, 2020. P. 83-87. DOI: 10.1109/ICENCO49778.2020.9357368.

ВЫЯВЛЕНИЕ АНОМАЛИЙ ВО ВРЕМЕННЫХ РЯДАХ НА ОСНОВЕ ТОПОЛОГИЧЕСКОГО АНАЛИЗА ДАННЫХ

НАДВОДНЮК Я.О.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. В статье проводится сравнительный анализ эффективности и вычислительной ресурсозатратности подхода на основе MTD-GAN и подхода, основанного на топологическом анализе и графовых представлениях данных, для задачи обнаружения аномалий в многомерных временных рядах. Исследование проводилось на наборах данных двух различных систем: киберфизической системы и системы мониторинга ИТ-инфраструктуры. Полученные результаты подтверждают, что подход, основанный на графовых представлениях и топологическом анализе данных, представляет собой применимую альтернативу генеративно-состязательным сетям.

Ключевые слова: многомерные временные ряды, обнаружение аномалий, топологический анализ данных

Введение

Задача обнаружения аномалий во многомерных временных рядах возникает в широком круге прикладных областей: мониторинг серверов, обнаружение атак в киберфизических системах, медицинская диагностика, финансовый анализ. Аномалии представляют собой отклонения от ожидаемого поведения системы, которые могут сигнализировать о критических событиях, требующих своевременной идентификации. Сложность задачи обусловлена наличием нелинейных взаимосвязей между различными переменными, которые могут динамически меняться во времени.

Настоящее исследование посвящено сравнительному анализу двух принципиально различных подходов к обнаружению аномалий: модели MTD-GAN (Multi-Task Discriminator Generative Adversarial Network), основанной на глубоком обучении с состязательной архитектурой, и подхода на основе топологического анализа данных (TDA) с использованием графовых представлений. Цель работы – оценить применимость, эффективность и вычислительную ресурсозатратность каждого подхода на двух различных наборах данных SMD (Server Machine Dataset) и HAI (HIL-based Augmented ICS Security Dataset), чтобы понять, есть ли преимущества у топологического подхода.

Подход с использованием MTD-GAN

Архитектура состоит из генератора и дискриминатора. Генератор представляет из себя LSTM-автоэнкодер. На вход принимается окно временного ряда размером 50xN, где N – количество признаков. Многозадачный дискриминатор имеет общий слой-энкодер и 3 головы, которые выполняют разные задачи: 1) классификация настоящее окно или сгенерированное, 2) реконструкция входного окна (декодер), 3) прогнозирование последнего временного шага окна по предыдущим.

Для обнаружения аномалий вычисляется аномальный скор. Это взвешенная сумма двух ошибок: ошибки реконструкции дискриминатора, ошибки прогнозирования. Оптимальные веса подбираются через grid search по метрике F1-score (без point-adjust) на тестовых данных.

Методы на основе генеративно-состязательных сетей являются одним из активно развивающихся и широко применяемых подходов к задаче обнаружения аномалий во

временных рядах. Поэтому MTD-GAN был выбран как бейзлайн для сравнительного анализа с топологическим подходом.

Подход с использованием графовых представлений и топологического анализа

Топологический подход реализован как многоэтапный вычислительный пайплайн. На первом этапе каждое скользящее окно временного ряда преобразуется в граф, где вершинами выступают признаки, а рёбра отражают меру сходства между ними внутри данного окна. Исследовано три способа построения графа корреляция Пирсона, косинусное сходство, евклидово расстояние [1].

На втором этапе для каждой матрицы расстояний вычисляются персистентные гомологии с использованием фильтрации Вьеториса-Рипса (с использованием библиотеки giotto-tda). Фильтрация последовательно увеличивает пороговое значение ε: при ε = 0 граф состоит из D изолированных вершин, при увеличении ε появляются рёбра, компоненты связности сливаются, могут возникать и замыкаться циклы. Каждое событие представляется парой («рождение», «смерть») – значениями ε, при которых структура возникла и исчезла. Для окна извлекаются диаграммы нулевого порядка, которые отображают «рождение» и «смерть» для компонент связности, и первого порядка – для петель.

Далее идет этап векторизации персистентных диаграмм. Он представляет из себя преобразование набора точек переменной длины в вектор фиксированной размерности. Для исследования были применены два метода векторизации. Метод Persistence Images, где каждая точка диаграммы создаёт гауссово пятно на 2D сетке (для экспериментов задан размер 15x15). Метод Betti Curves – функции, показывающие число «живых» топологических структур при каждом ε (количество точек на оси ε равно 50).

На четвёртом этапе вычисляются два сигнала аномалии. Первый (в таблице обозначен как w_t) – топологическое расстояние, оцениваемое как расстояние Махаланобиса векторизованной диаграммы от центра распределения нормальных окон, с оценкой ковариации методом MinCovDet [2]. Второй (в таблице обозначен как w_g) – ошибка восстановления матрицы смежности графа с помощью автоэнкодера, обученного на нормальных графах. Итоговый аномальный скор формируется как взвешенная сумма двух сигналов, веса подбираются через grid search.

Результаты

Таблица 1

Результаты подхода с использованием графовых представлений и топологического анализа для набора данных SMD

Метод для графов	pearson		cosine		euclidian	
TDA метод	pers_image	betti_curve	pers_image	betti_curve	pers_image	betti_curve
ROC AUC	0.7592	0.7592	0.8593	0.8406	0.8787	0.8787
PR AUC	0.312	0.312	0.5365	0.5203	0.4874	0.4874
F1	0.3317	0.3317	0.6573	0.6204	0.5001	0.5001
F1 PA	0.6333	0.6333	0.7702	0.7584	0.6809	0.6809

Веса	'w_g': 0.1, 'w_t': 0.0, 'pct': 93	'w_g': 0.1, 'w_t': 0.0, 'pct': 93	'w_g': 0.1, 'w_t': 0.0, 'pct': 93	'w_g': 1.0, 'w_t': 0.1, 'pct': 93	'w_g': 0.1, 'w_t': 0.0, 'pct': 93	'w_g': 0.1, 'w_t': 0.0, 'pct': 93
Сегменты	6/8	6/8	8/8	8/8	8/8	8/8
Время, сек	31.89	15.03	37.25	15.95	39.20	14.18
TDA время, сек	25.75	8.78	30.96	9.69	33.07	8.07
Обучение авто-энкодера, сек	14.50	14.50	10.70	10.70	17.10	17.10
Векторизация, сек	4.56	0.32	4.38	0.29	4.57	0.22

Можно сделать вывод, что комбинация косинусного сходства и Persistence Images проявила себя лучше других комбинаций. Конкретный выбор w_g = 0.1, w_t = 0 (или эквивалентный w_g = 0, w_t = 0.1) определяется порядком перебора в grid search и не должен интерпретироваться как доказательство преимущества одного источника сигнала над другим. На SMD топологические признаки и графовая реконструкция одинаково информативны, но избыточны друг относительно друга.

Таблица 2

Результаты подхода с использованием графовых представлений и топологического анализа для набора данных HAI

Метод для графов	pearson		cosine		euclidian	
TDA метод	pers_image	betti_curve	pers_image	betti_curve	pers_image	betti_curve
ROC AUC	0.6884	0.6218	0.7288	0.7673	0.8462	0.8462
PR AUC	0.1169	0.0746	0.1437	0.1575	0.2822	0.2822
F1	0.1562	0.1494	0.2481	0.3194	0.4198	0.4198
F1 PA	0.6168	0.4194	0.6384	0.6513	0.6651	0.6651
Веса	w_g': 0.0, 'w_t': 0.1, 'pct': 93	w_g': 0.1, 'w_t': 1.0, 'pct': 93	w_g': 0.0, 'w_t': 0.1, 'pct': 93	w_g': 0.1, 'w_t': 0.4, 'pct': 93	w_g': 0.1, 'w_t': 0.0, 'pct': 93	w_g': 0.1, 'w_t': 0.0, 'pct': 93
Сегменты	2/2	2/2	2/2	2/2	2/2	2/2
Время, сек	46.97	29.40	44.54	22.29	57.04	22.54
TDA время, сек	34.47	17.06	32.20	9.70	44.72	10.24
Обучение авто-энкодера, сек	295.30	295.30	356.10	356.10	182.80	182.80
Векторизация, сек	7.01	1.59	6.91	1.65	4.63	0.36

Можно увидеть, что комбинация евклидова расстояния и Betti Curves проявила себя лучше других комбинаций. На HAI ситуация отличная от SMD. В нескольких конфигурациях оптимальные веса распределяются между обоими компонентами с заметным вкладом топологии, что свидетельствует о дополняющем характере сигналов на этом датасете. Топологические признаки предоставляют независимый сигнал и компенсирует недостаток информации при определённых выборах построения графа.

Далее были проведены эксперименты для метода с MTD-GAN. Результаты сравнения обоих подходов на датасетах SMD и HAI представлены в таблице 3. Для топологического подхода приведены результаты лучшей комбинации методов.

На обоих наборах данных MTD-GAN превзошёл топологический подход по большинству метрик качества. Оба подхода обнаружили все аномальные сегменты, включая короткие сегменты длиной 2-3 точки в наборе данных SMD. Это подтверждает применимость TDA подхода к задаче, хотя MTD-GAN остаётся более точным. Для HAI разрыв в PR AUC более существенный, чем на SMD. Это указывает на значительный рост ложноположительных срабатываний при увеличении размерности данных.

Таблица 3

Сравнение результатов обоих подходов

Метрика	SMD		HAI	
	Результат для MTD-GAN	Результат cosine и pers_image	Результат для MTD-GAN	Результат euclidian и betti_curve
ROC AUC	0.9445	0.8593	0.9378	0.8477
PR AUC	0.6850	0.5365	0.5584	0.3004
F1	0.6275	0.6573	0.6502	0.3875
F1 (point-adjust)	0.9118	0.7702	0.8214	0.6646
Обнаружено сегментов	8/8	8/8	2/2	2/2
Время детекции аномалий (без обучения), сек	26.51	37.25	12.47	10.24
Время обучения, сек	473.20	30.96	861.60	182.80

Сравнение ресурсоемкости

MTD-GAN и автоэнкодер обучаются на GPU, тогда как giotto-tda и MinCovDet (для расстояния Махаланобиса) используют только CPU. Хотя обучение MTD-GAN более длительное, чем у топологического подхода, его, как и детекцию, можно распараллелить на GPU. Обучение также имеет линейную вычислительную сложность относительно n.

Для подхода, основанного на графовых представлениях и топологическом анализе данных, только задачи обучения автоэнкодера и получения ошибки реконструкции графа могут выполняться на GPU. Некоторые операции выполняются на CPU. Построение графов, которое имеет квадратичную вычислительную сложность; фильтрация Вьеториса-Рипса, которая имеет сложность близкую к кубической (в худшем случае выполняется $O(n^9)$); вычисление оценки топологических признаков, которое имеет вычислительную сложность $O(k^3 + m*k^2)$, где m – количество окон, k – размерность вектора топологических признаков. Вычислительная сложность определялась относительно количества признаков n.

Заключение

Эксперименты на SMD и HAI показали, что MTD-GAN в целом демонстрирует лучшие значения ROC AUC, PR AUC и F1-score, однако топологический подход остаётся конкурентоспособным и в ряде конфигураций обеспечивает близкое качество детекции. Подход, основанный на графовых представлениях и топологическом анализе данных, хуже масштабируется при большом количестве признаков, и зависит от выбора метода векторизации, но по времени детекции сопоставим с MTD-GAN.

Список литературы

1. Мляхилу Н. Дж., Новикова Е. С. Подходы к построению графа для обнаружения аномалий во временных рядах с помощью графовых нейронных сетей // XXVIII Международная конференция по мягким вычислениям и измерениям (SCM’2025): материалы конференции. – Санкт-Петербург, 2025. – С. 181–184.

2. Chazal F., Royer M., Levrard C. Topological Analysis for Detecting Anomalies (TADA) in Time Series [Электронный ресурс]. – arXiv, 2024. – arXiv:2406.06168.

ИССЛЕДОВАНИЕ ОСОБЕННОСТЕЙ ОБУЧЕНИЯ МАСШТАБИРУЕМЫХ АРХИТЕКТУР ИМПУЛЬСНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ПЕРИФЕРИЙНЫХ УСТРОЙСТВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

. Пономаренко О.Г, Андреева Н.В.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

Аннотация. В статье исследуется влияние квантования синаптических весов импульсной нейронной сети на точность решения задачи распознавания изображений на примере архитектуры CoLaNET. Рассмотрены кусочно-линейная и ступенчатая функции зависимости веса синапса от синаптического ресурса. Показано, что квантование веса на 16 уровней (4 бита) не приводит к существенному снижению точности.

Ключевые слова: импульсные нейронные сети, нейроморфные архитектуры, периферийный ИИ, CoLaNET.

Работа выполнена при финансовой поддержке Министерства науки и высшего образования Российской Федерации – государственное задание в области научной деятельности FSEE-2025-0005.

1. Введение

Одним из основных требований, предъявляемых к архитектурам нейронных сетей для периферийных устройств искусственного интеллекта, является их эффективное обучение при значительно упрощённых в интересах аппаратной реализации составных элементах. Для импульсных нейронных сетей (ИмНС) таковыми являются модели нейрона и синапса. В статье на примере ИмНС архитектуры CoLaNET [1] рассмотрено такое аппаратно-ориентированное упрощение модели синапса, как квантование веса. Подобное решение даст возможность эмулировать вес синапса несколькими дискретными значениями, а не аналоговым уровнем сигнала, что существенно сократит требуемые аппаратные ресурсы при реализации ИмНС на ПЛИС.

Цель данной работы – оценка изменения метрики точности ИмНС архитектуры CoLaNET в режимах обучения и инференса при квантовании синаптических весов на примере решения задачи распознавания изображений базы данных MNIST [2].

2. Краткое описание архитектуры CoLaNET и конфигурация сети для работы с MNIST

Архитектура импульсных нейронных сетей CoLaNET (Columnar-Layered NETwork) предназначена для решения задач классификации и моделируется с помощью эмулятора ИмНС ArNI-X [3]. Сеть этой архитектуры состоит из колонок, разделённых на функциональные слои. Каждая колонка отвечает за распознавание определённого класса.

Рис. 1. Структура сети архитектуры CoLaNET для задачи классификации MNIST

Синаптическая пластичность сети основана на двух одинаковых по силе воздействия, но противоположных по знаку, механизмах: дофамин-модулированном (усиливающем) и анти-Хеббовском (ингибирующем).

Структура сети рассматриваемой архитектуры, функциональные особенности слоёв, а также взаимодействие механизмов пластичности, подробно рассмотрены в [1].

Для задачи классификации монохроматических изображений рукописных цифр базы данных MNIST была сконфигурирована сеть, структура которой приведена на рис. 1. Сеть имеет 10 колонок, каждая из которых отвечает за распознавание своей цифры, в каждой колонке по 20 L-нейронов для распознавания подклассов (в нашем случае подклассы – заметно различимые способы и почерки написания одной и той же цифры). Стрелками на рисунке показаны различные виды синапсов и их групп.

3. Обзор видов весовых функций синапсов

В архитектуре CoLaNET механизмы пластичности (дофамин-модулированной и анти-Хеббовской) воздействуют не напрямую на вес синапса, а на синаптический ресурс, связанный с весом некоторой функцией. Классический вариант такой функции – *smooth*, описывается формулой (1) и проиллюстрирован на рис. 2а.

$$w_{smooth}(W) = w_{min} + \frac{(w_{max} - w_{min})\max(W,0)}{w_{max} - w_{min} + \max(W,0)} \quad (1)$$

Функция *smooth* равна w_{min} при значениях синаптического ресурса (W) меньше нуля, и асимптотически приближается к w_{max} при значениях W больше нуля. Такой вид функции способствует снижению влияния явления катастрофического забывания, так как при достаточно удалённых от нуля значениях синаптического ресурса изменения веса при изменениях синаптического ресурса будут достаточно малы в течении некоторого количества актов действия механизмов пластичности.

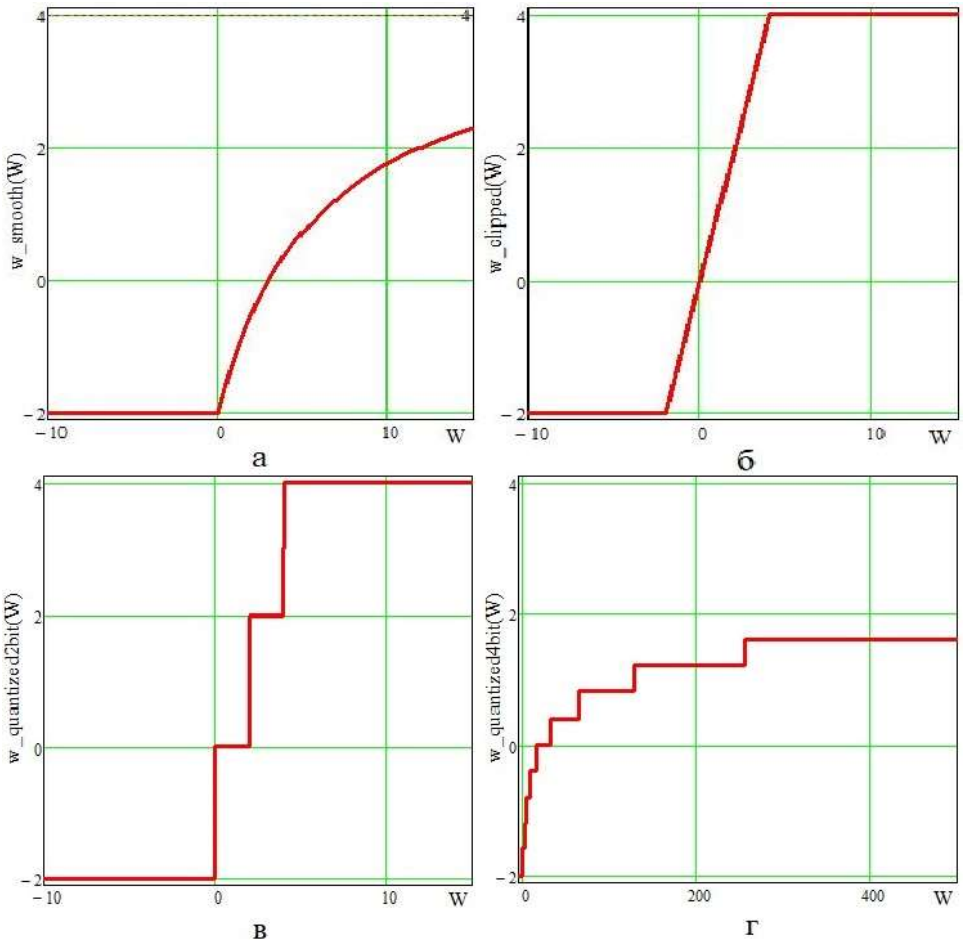


Рис. 2. Графики функций smooth (а), clipped (б) и quantized для 4 (в) и 16 (г) дискретных значений (w_{min} = -2, w_{max} = 4)

Упрощённым вариантом этой функции является её кусочно-линейное приближение – функция *clipped* (2), являющаяся промежуточным вариантом между квантующимся весом и функцией *smooth*. Её преимуществом является простота реализации её вычисления: используется лишь операция сравнения.

$$w_{clipped}(W) = \max(w_{min}, \min(w_{max}, W)) \quad (2)$$

И, наконец, функция *quantized* (3) с дискретными возможными значениями веса. В проведённом исследовании использовалась функция с 4 (2 бита) и 16 (4 бита) уровнями, графики которых приведены на рисунке 2в и 2г соответственно.

$$w_{quantized}(W) = \begin{cases} w_{min}\ if\ W < 0 \\ Q_{\lfloor \log_2 W \rfloor}\ if\ \log_2 W < |Q| + 1 \\ w_{max}\ o/w \end{cases} \quad (3)$$

где Q – конечное множество возможных значений весов (не включая значения w_{min} и w_{max}).

4. Методология исследования

Для проведения сравнения точности сетей с различными функциями зависимости веса синапса от синаптического ресурса за базовое значение взята точность 96,07%, достигнутая сетью на базе функции *smooth* с набором гиперпараметров, указанных в таблице 1.

Важно отметить, что обучение сетей при изменении весовой функции, в том числе введении квантования веса, происходило с одним и тем же набором гиперпараметров, приведенным в таблице 1. Кроме того, данные выборок также не перемешивались. Так как

гиперпараметры не менялись для запуска цикла обучения разных сетей, результаты являются статистически независимыми друг от друга. Такая методология гарантирует непосредственное сравнение эффективности сетей с абсолютно идентичной архитектурой, параметрами и обучающими данными, но различными функциями зависимости веса синапса от синаптического ресурса.

Таблица 1

Гиперпараметры исходной сети

Название гиперпараметра	Обозначение в ArNI-X	Значение
Постоянная времени утечки мембранного потенциала	chartime	3
Минимальный мембранный потенциал	minpotential	0
Стимуляция шумом Пуассона	stochastic_stimulation	2.21207
Постоянная анти-Хеббовской пластичности	weight_inc	-0.176526
Коэффициент изменения стабильности	stability_resource_change_ratio	0.0497573
Минимальный вес	minweight	-0.253122
Максимальный вес	maxweight	0.0923957
Количество “тихих” синапсов	nsilentsynapses	27
Коэффициент избыточного синаптического веса	threshold_excess_weight_dependent	0.217654
Вес reward-синапсов	weight	0.179376

5. Результаты и выводы

Результирующие метрики точности приведены в таблице 2. Там же указано суммарное время работы сети от начала обучения до окончания работы в режиме инференса. Это время не может считаться количественным показателем, так как полностью зависит от параметров системы эмуляции. Однако, так как все запуски были произведены на одной и той же системе (эмуляция на процессоре AMD Ryzen 5 1600 Six-Core Processor), этот параметр можно считать качественным показателем скорости обучения.

Таблица 2

Результирующая точность и время работы

Вид функции w(W)	Точность, %	Время работы
smooth	96,07	3ч 18мин 26с
clipped	81,9	3ч 9мин 54с
quantized 2-bit	89,22	3ч 11мин 56с
quantized 4-bit	95,16	3ч 15мин 50с

Исходя из полученных результатов можно сделать вывод о том, что в рамках архитектуры CoLaNET при необходимости возможно использовать модель синапса с квантованием веса на 16 уровней (4 бита) без значительной потери точности (менее 1%). Кроме того, время обучения и скорость эмуляции работы сети мало зависят от выбора вида функции зависимости веса синапса от синаптического ресурса.

Список литературы

1. Kiselev M. CoLaNET – A Spiking Neural Network Learning to Classify BT - Advances in Neural Computation, Machine Learning, and Cognitive Research IX под ред. B. Kryzhanovsky [и др.]., Cham: Springer Nature Switzerland, 2026.С. 107–122.
2. LeCun Y. L. Bottou, Y. Bengio, and P. Haffner. Gradient-Based Learning Applied to Document Recognition. Proc. IEEE 1998
3. Киселёв М. В. The ArNI-X spiking neural network simulator [Электронный ресурс]. URL: https://arni-x.pro

(дата обращения: 03.05.2026).

МОДЕЛЬ АНАЛИЗА И РАСПОЗНОВАНИЯ НОМЕРОВ АВТОТРАНСПОРТА ДЛЯ ДОПУСКА НА ТЕРРИТОРИЮ ОРГАНИЗАЦИИ

ПОЧЕКУНИН А.П.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. Исследование направленно на разработку нейросетевой модели для автоматического распознавания и анализа номера автотранспорта для организации контроля доступа. В работе использован датасет, содержащий изображения автотранспорта с выделенными номерами, и датасет, содержащий модифицированные символы, используемые в номерах. В качестве основы была использована нейросетевая архитектура, включающая детекции номера и последующее распознавание. Для повышения точности были использованы нормализация, фильтрация и выравнивания области распознавания. Результаты подтверждают возможность применения разработанной модели для автоматизации допуска автотранспорта на территорию организации.

Ключевые слова: нейронные сети, машинное обучение, YOLO, VGG19, сегментация, компьютерное зрение, распознавание номеров, ANPR.

Введение

Автоматизация процессов контроля доступа автотранспорта является важной задачей для повышения безопасности и эффективности работы. Современные системы контроля доступа требуют высокой скорости обработки данных и надёжности при принятии решений о допуске автотранспорта. Традиционные методы допуска, основанные на ручной проверке или пропускном режиме, характеризуются значительными временными затратами и подвержены ошибкам, что увеличивает потребность в разработке автоматизированных решений, способных самостоятельно выполнять идентификацию автотранспорта. В последние годы методы глубокого обучения активно применяются в задачах анализа изображений. В частности, свёрточных нейросетей на базе YOLO [1], используемых для быстрой и точной локализации объектов на изображении, и на базе VGG19 [2], применяемых для классификации изображений. Данные архитектуры демонстрируют высокую точность обработки изображений за счёт способности устойчиво работать при разных искажениях.

Материалы и методы обработки

В рамках исследования для обучения детекции был выбран набор изображений «Car Number Plate Dataset» с сайта kaggle, состоящий из 433 номерных знаков с координатами ограничивающих рамок (рис. 1), а также был создан свой датасет на 6900 изображений, полученных путём модификации эталонных символов, используемых в номерных знаках автотранспорта (рис. 2). Объём выборки составил 300 изображений для каждого класса символов.

Рис. 1. Пример данных

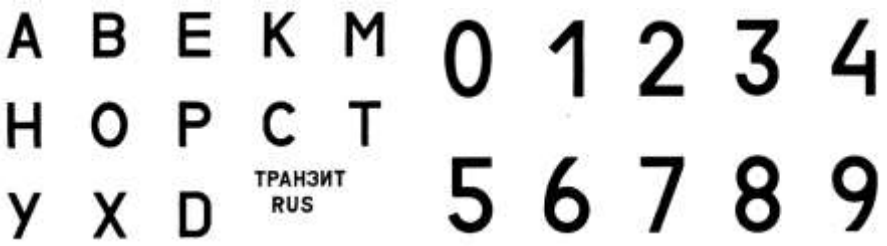


Рис. 2. Пример данных

Обработка данных выполняется в несколько последовательных этапов. На первом осуществляется детекция номера с использованием модели YOLO, позволяющей выделить область на изображении. Далее выполняется предобработка изображения, включающая нормализацию, подавление шумов и выравнивание номерной области. После этого применялась сегментация символов на основе анализа связных компонент и морфологических операций. Следующим этапом идёт распознавание символов с использованием модели VGG19, адаптированной под задачу классификации символов российских номеров автотранспорта. Для получения вероятностей принадлежности символа к определённому классу использовалась многопеременная логистическая функция:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_i}}$$

В качестве метрик для оценки модели использовались точность классификации символа и IoU accuracy для оценки качества детекции.

Метрика IoU accuracy рассчитывается по следующей формуле:

$$IoU = \frac{TP}{TP + FP + FN}$$

, где TP – это количество истинноположительных предсказаний, TP – это количество истиннонегативных предсказаний, FP – количество ложноположительных предсказаний, FN – количество ложноотрицательных предсказаний.

Дополнительно для оценки итогового результата распознавания использовалась усреднённая уверенность по символам:

$$\bar{P} = \frac{1}{n}\sum_{i=1}^{n} p_i$$

Для обучения модели датасет был разбит на обучающую, валидационную и тестовую выборки в соотношении 80%, 10% и 10% соответственно. В качестве оптимизатора был выбран Adam [3] для обеспечения эффективной настройки параметров нейросети.

Результаты

Итоговая модель выполняет детекцию номерного знака, сегментацию символов и их последующее распознавание как на одном изображении, так и на видеопотоке (рис. 3). Оценка работы системы проводилась на тестовой выборке.

Рис. 3. Пример работы модели

Для оценки эффективности системы были использованы метрики качества детекции и распознавания. На тестовой выборке модель показала следующие результаты, продемонстрированные на таблице 1.

Таблица 1

Результаты тестовой выборки

Метрика	Среднее значение
IoU	0.91
Точность распознования символов	0.94
Средняя уверенность распознования	0.92
Время обработки кадра (1 изображения)	46 микросекунд

Полученные результаты показывают, что модель обеспечивает высокую точность выделения номеров и корректное распознавание символов.

Заключение

Практическая значимость работы заключается в создании инструмента для автоматизации систем контроля доступа автотранспорта, что позволяет повысить уровень безопасности и снизить нагрузку на персонал.

Модель корректно выполняет детекцию номерных знаков и распознавание символов. Присутствуют незначительные ошибки, связанные с распознаванием отдельных символов при неблагоприятных условиях, однако они не оказывают существенного влияния на итоговую точность модели. В целом модель демонстрирует высокие показатели точности и стабильную работу в подходящих условиях, а также обеспечивает необходимую скорость обработки, чтобы систему можно было использовать в режиме реального времени. На примере видно, что модель способна распознать номер даже при повороте машины на 20 градусов.

Использование ограниченного набора данных, и наличие автоматически сгенерированных изображений символов может снижать обобщающую способность модели при работе в различных условиях эксплуатации. Отсутствие широкого разнообразия реальных изображений может влиять на точность распознавания в сложных условиях. Тем не менее, предложенная модель показала эффективность в задачах автоматического распознавания номеров автотранспорта.

Дальнейшие исследования целесообразно направить на расширение датасетов за счёт реальных изображений, улучшение устойчивости модели к внешним факторам и оптимизацию алгоритмов предобработки и сегментации. Также перспективным направлением является адаптация модели для работы с различными форматами номерных знаков и интеграция с существующими системами контроля доступа.

Предложенный подход демонстрирует эффективность использования методов глубокого обучения для решения задач автоматического распознавания номерных знаков, однако его внедрение в реальные системы требует дальнейшего совершенствования и расширения обучающей выборки за счёт более разнообразных и репрезентативных данных.

Список литературы

1. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi You Only Look Once: Unified, Real-Time Object Detection (2015)
2. Karen Simonyan, Andrew Zisserman Very Deep Convolutional Networks for Large-Scale Image Recognition (2014)
3. Diederik P. Kingma, Jimmy Ba Adam: A Method for Stochastic Optimization (2014)
4. Du S., Ibrahim M., Shehata M., Badawy W. Automatic License Plate Recognition (ALPR): A State-of-the-Art Review. (2013)
5. Silva S., Jung C. Real-time Brazilian license plate detection and recognition using deep convolutional neural networks. (2017)

РАЗРАБОТКА И ВАЛИДАЦИЯ РЕЖИМНО-АДАПТИВНОЙ МОДИФИКАЦИИ TEMPORAL FUSION TRANSFORMER ДЛЯ КРИПТОВАЛЮТНЫХ ВРЕМЕННЫХ РЯДОВ

СЕРГЕЕВ А.П.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. Разработана и экспериментально проверена CRISP-TFT – авторская режим-ориентированная модификация Temporal Fusion Transformer для прогнозирования 4-часовых криптовалютных рядов BTCUSDT и ETHUSDT. Гипотеза исследования состояла в том, что явный детектор рыночных режимов, учет неопределенности прогноза и Sharpe-ориентированный торговый слой улучшат риск-метрики по сравнению с базовым TFT. Для проверки гипотезы сформирован воспроизводимый датасет, реализованы бенчмарки LSTM, GRU, Transformer+GRU и TFT, а также собран последовательный протокол train/validation/test без случайного перемешивания. В ходе экспериментов обнаружены аномально высокие значения Sharpe, вызванные утечкой целевой переменной и ошибками протокола оценки. После фиксации каузального сдвига и пересборки пайплайна преимущество CRISP-TFT в текущей реализации не подтвердилось. Основной результат работы – воспроизводимая инфраструктура бенчмарка и методологические ограничения режим-ориентированной оптимизации финансовых временных рядов.

Ключевые слова: финансовые временные ряды, Temporal Fusion Transformer, CRISP-TFT, рыночный режим, MMD, утечка данных, коэффициент Шарпа.

Введение

Финансовые временные ряды характеризуются нестационарностью, высокой волатильностью и сменой рыночных режимов. Для криптовалют это проявляется особенно резко: периоды направленного роста, падения и консолидации различаются распределением доходностей, уровнем риска и реакцией на внешние события. Поэтому модель, обученная на усредненном историческом распределении, может деградировать при структурном сдвиге даже при приемлемой ошибке прогноза на валидации.

Temporal Fusion Transformer (TFT) является сильной архитектурой для мультигоризонтного прогнозирования: она объединяет рекуррентную обработку локальной динамики, механизм выбора переменных и интерпретируемое multi-head attention [1]. В базовой версии TFT разделяет входы на статические признаки, наблюдаемые прошлые ряды и известные будущие ковариаты, но не получает отдельного сигнала текущего рыночного режима. Кроме того, стандартная оптимизация по прогнозной ошибке не совпадает с финансовой целью, где важны доходность после издержек, Sharpe, Calmar и максимальная просадка.

Цель исследования – разработать и проверить CRISP-TFT (Causal/Regime-aware, Incertitude-aware, Sharpe-Prioritized Temporal Fusion Transformer). CRISP-TFT – это модификация TFT: в нее добавлены онлайн-переменная режима z_t, выходная голова неопределенности и торговый слой, переводящий прогноз в непрерывную позицию. Проверяемая гипотеза формулируется так: при корректном каузальном протоколе CRISP-TFT должна улучшить метрики относительно базового TFT за счет адаптации к режимам и снижения позиции при высокой неопределенности.

Данные и воспроизводимый пайплайн

Датасет собирался из публичных исторических klines Binance Data Vision [2]. Сборщик параметризуется торговой парой и таймфреймом 4h: через S3-запрос к каталогу monthly klines он получает список помесячных ZIP-архивов, скачивает каждый архив, извлекает CSV, объединяет файлы, проверяет timestamp, сортирует свечи по времени и удаляет дубликаты. Если каталог временно недоступен, скрипт генерирует адреса архивов по стандартному шаблону Binance. Такой порядок делает датасет повторяемым: достаточно указать BTCUSDT или ETHUSDT и заново запустить сборщик, после чего будут получены OHLCV-поля Open, High, Low, Close, Volume, quote volume, число сделок и taker-buy объемы.

После выгрузки рассчитывались логарифмические доходности $r_t = ln(\frac{C_t}{C_{t-1}})$, производные объемные признаки и технические индикаторы: скользящие средние, EMA, волатильность на окнах, RSI и MACD. Целевая переменная задавалась строго как будущая доходность $y_t = r_{t+1}$, в коде target = log_ret.shift(-1). Нормализация признаков выполнялась только на train-части, затем те же параметры применялись к validation и test; пропуски обрабатывались forward fill и, при необходимости, масками. Последовательность преобразовывалась в пары «история – горизонт»: X_τ содержит окно encoder, Y_τ – будущий горизонт decoder. Для TFT признаки дополнительно разделялись на static, past observed и future known.

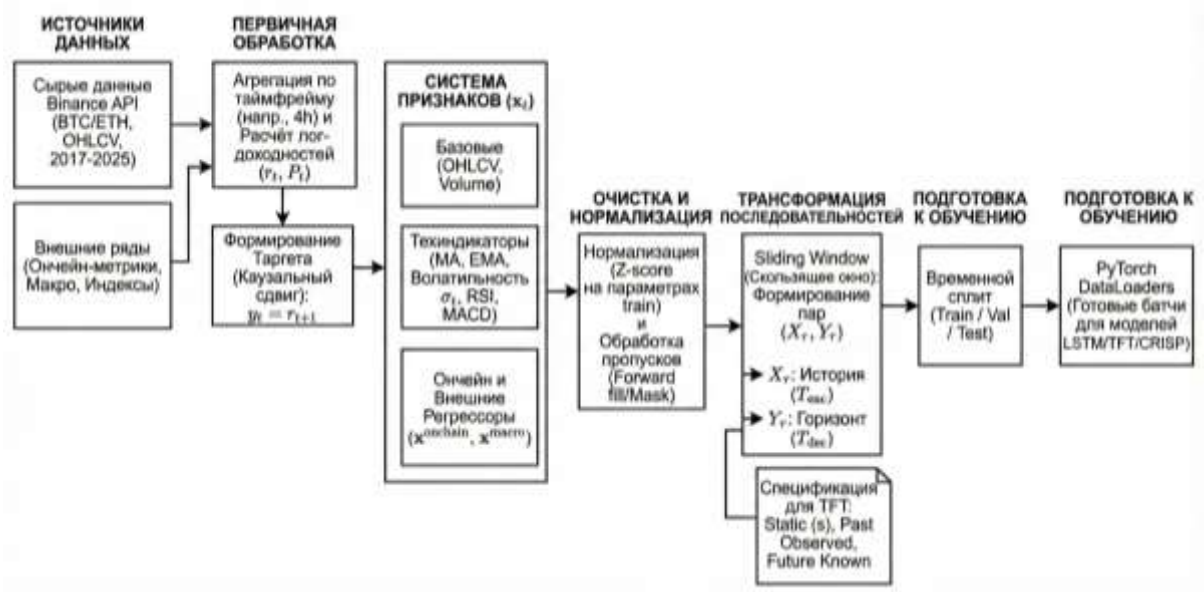


Рис. 1. Пайплайн сбора данных, формирования признаков и подготовки выборок

Архитектура и протокол проверки

Базовый TFT состоит из трех ключевых частей. Во-первых, Variable Selection Network через GRN и softmax вычисляет веса признаков в каждый момент времени и тем самым отбирает информативные переменные. Во-вторых, LSTM encoder-decoder моделирует краткосрочную динамику окна истории и горизонта прогноза. В-третьих, static enrichment и interpretable attention связывают локальные представления с долгосрочными зависимостями. Поэтому TFT выбран не как слабый baseline, а как сильная точка сравнения: если режимная модификация действительно полезна, она должна улучшить именно его.

В CRISP-TFT режимный модуль вычисляет z_t по текущему скользящему окну и подает его в TFT как дополнительный контекст. Ранние версии использовали сигнатуры путей глубины 2, которые кодируют геометрию траектории через итерированные интегралы [3], и MMD-подобную дистанцию для сравнения текущего окна с эталонными режимами [4]. Далее проверялись K-Means-кластеры, эвристика по RSI/тренду и гибридный байесовский детектор с априорными поправками; близкие задачи онлайн-детектирования режимов финансового рынка рассматриваются в [5]. Интуиция модификации проста: в режиме роста, падения и флэта модель должна по-разному взвешивать признаки и по-разному интерпретировать неопределенность.

Выходная голова CRISP-TFT формировала параметры прогноза μ_t и σ_t . Торговый слой переводил их в непрерывную позицию $p_t = tanh(k \cdot \frac{\mu_t}{\sigma_t})$ в диапазоне от −1 до 1; большая

σ_t уменьшает размер позиции. Доходность стратегии считалась как $r_t^p = p_{t-1} \cdot r_t - cost \cdot |p_t - p_{t-1}|$. Прогнозная часть обучалась по Gaussian NLL, а риск-ориентированная часть включала дифференцируемую Sharpe-loss и регуляризацию инвариантности между режимами: $L = \alpha L_{pred} + \beta L_{Sharpe} + \gamma L_{inv}$. Важно, что это исследовательский торговый слой, а не готовая инвестиционная стратегия.

Для сравнения были реализованы LSTM, GRU, Transformer+GRU и базовый TFT. Оценка строилась по последовательному разделению времени: обучение использовало только прошлое, validation применялась для выбора параметров, а test рассматривался как будущий интервал. Метрики делились на прогнозные и торговые. В таблице приведены торговые показатели; стрелка в заголовке показывает направление улучшения. Sharpe показывает, насколько стабильную и неслучайную доходность выдает модель [6], MDD – наибольшую просадку капитала [7], Calmar – отношение годовой доходности к MDD.

Результаты и обсуждение

На ранних версиях CRISP-TFT были получены аномально высокие значения: validation Sharpe = 15,7, test Sharpe = 8,2, Information Coefficient = 0,47. Для 4-часовых криптовалютных рядов такие числа нельзя интерпретировать как превосходство модели: они стали диагностическим сигналом ошибки постановки. Проверка показала несколько источников ложноположительного результата: в раннем target использовалась текущая доходность, видимая в encoder; нормализация местами зависела от будущих данных; tanh(signal·10) в Sharpe-loss попадал в насыщение; формула доходности стратегии была несогласована с временным сдвигом позиции.

После исправлений был зафиксирован каузальный target = log_ret.shift(-1), параметры нормализации стали обучаться только на train, коэффициент в tanh был уменьшен, а торговый слой пересчитан с задержкой позиции. В контрольной сборке явная утечка не обнаружена, но сигнал стал слабым: исходная гипотеза о превосходстве CRISP-TFT над базовым TFT в текущей реализации не подтвердилась.

Таблица 1

Сравнение моделей по торговым метрикам

Модель	Sharpe ↑	Ann. Return ↑	|MDD| ↓	Calmar ↑
LSTM	-0,89	-43,1%	52,8%	-0,81
GRU	0,65	12,4%	49,1%	0,27
Transformer+GRU	1,36	80,4%	36,0%	2,19
TFT	1,81	121,6%	30,0%	4,04
CRISP-TFT	-0,013	-5,2%	0,002%	-0,339

Лучший воспроизводимый результат в текущей сводке показал базовый TFT: он дал максимальные Sharpe и Calmar при умеренной просадке. Transformer+GRU оказался сильнее простых рекуррентных моделей, что указывает на пользу attention-механизма, но все же уступил TFT. CRISP-TFT в очищенном протоколе не показал преимущества: малая просадка здесь не является самостоятельным успехом, потому что из-за низкого масштаба прогнозов и осторожного риск-скейлинга стратегия почти не входила в позицию, а итоговая доходность осталась отрицательной.

Полученный результат не обесценивает режимную идею. Он показывает, что явный режимный сигнал может стать источником дополнительного шума, если детектор запаздывает, часто переключается или не согласован с торговым слоем. Базовый TFT,

вероятно, уже частично захватывает режимные эффекты через блоки VSN и attention без отдельной дисперсии детектора. Поэтому вклад работы состоит не в демонстрации завышенной метрики, а в строгой проверке гипотезы и фиксации условий, при которых режимная надстройка пока не улучшает сильную модель.

Заключение

В работе разработана и проверена CRISP-TFT – режимно-адаптивная модификация TFT для криптовалютных временных рядов. Реализованы воспроизводимый сбор OHLCV-данных, система признаков, обучающий, валидационный и тестовый пайплайн, набор бенчмарков и торговый слой с учетом издержек. После устранения утечек текущая версия CRISP-TFT не превзошла базовый TFT, однако этот результат является методологически значимым: он показывает, какие элементы финансового ML-протокола способны создавать ложную прогностическую силу.

Дальнейшая работа должна включать проверки отдельных компонентов CRISP-TFT, отдельный расчет торгового слоя по BTCUSDT и ETHUSDT, более устойчивый детектор режимов, последовательную проверку на нескольких рыночных циклах и расширение набора метрик устойчивости. Подготовленная инфраструктура позволяет повторить эксперимент, заменить детектор или торговое правило и проверить, улучшает ли новая версия риск-метрики относительно зафиксированного TFT-бенчмарка.

Список литературы

1. Lim B., Arik S.O., Loeff N., Pfister T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting // International Journal of Forecasting. 2021. Vol. 37, No. 4. P. 1748–1764. DOI: 10.1016/j.ijforecast.2021.03.012.
2. Binance Historical Market Data // Binance Data. URL: https://data.binance.vision/ (дата обращения: 04.05.2026).
3. Chevyrev I., Kormilitzin A. A primer on the signature method in machine learning // arXiv preprint. 2016. arXiv:1603.03788. DOI: 10.48550/arXiv.1603.03788.
4. Gretton A., Borgwardt K.M., Rasch M.J., Scholkopf B., Smola A. A kernel two-sample test // Journal of Machine Learning Research. 2012. Vol. 13. P. 723–773.
5. Issa Z., Horvath B. Non-parametric online market regime detection and regime clustering for multidimensional and path-dependent data structures // arXiv preprint. 2023. arXiv:2306.15835. DOI: 10.48550/arXiv.2306.15835.
6. Sharpe W.F. The Sharpe Ratio // The Journal of Portfolio Management. 1994. Vol. 21, No. 1. P. 49–58.
7. Magdon-Ismail M., Atiya A.F. Maximum drawdown // Risk. 2004. Vol. 17, No. 10. P. 99–102.

ГЕНЕРАЦИЯ ДИАГРАММ С ПРИМЕНЕНИЕМ КРИТИЧЕСКОГО АГЕНТА И КОЛИЧЕСТВЕННОЙ ОЦЕНКИ ВЛИЯНИЯ ПО ОБРАТНОЙ СВЯЗИ

СТАЦЕНКО[1] Г. Э., ШУРАНДИН[1] А. В., ТРУСОВ[1] И. А., ТРОФИМОВ[1] Ю. В., АВЕРКИН[1] А. Н.

2. Федеральное государственное бюджетное образовательное учреждение высшего образования «Университет «Дубна»

Аннотация. Рассматривается агентная система на основе большой языковой модели для автоматической генерации диаграмм из текстовых описаний. Система реализует конвейер: генерация черновика, критика, применение плана исправлений и верификация. Для оценки вклада критики применяется метод Шепли и А/Б-контрфактический прогон. Поддерживаются четыре типа диаграмм с тремя средствами визуализации. Апробация проведена на задаче воспроизведения архитектуры U-Net по текстовому заданию.

Ключевые слова: большая языковая модель, генерация диаграмм, критический агент, метод Шепли, цикл самокритики, визуализация архитектур.

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема № 124112200072-2).

Введение

Современные инструменты визуализации архитектур программных систем и нейронных сетей требуют от специалиста значительных трудозатрат: ручного составления описаний узлов и рёбер, выбора средства визуализации, отладки компоновки. Появление больших языковых моделей (БЯМ) открывает возможность автоматизации этого процесса: пользователь формулирует задачу на естественном языке, а система самостоятельно генерирует структурированное описание диаграммы и визуализирует его. Однако системы на основе БЯМ при однопроходной генерации нередко допускают ошибки в семантике схемы: пропускают элементы, неверно интерпретируют тип связи, нарушают компоновку [1]. Для преодоления этих ограничений перспективным направлением является применение агентных архитектур с механизмом итеративного самоулучшения [2].

В данной работе представлен граф-агент - агент на основе БЯМ для автоматической генерации диаграмм, в основе которого лежит конвейер из пяти этапов: генерация черновика, критика языковой моделью, применение плана исправлений, верификация и визуализация. Дополнительно система включает анализ значимости критики на основе метода Шепли и А/Б-контрфактический прогон для количественной оценки вклада критики в качество итоговой диаграммы.

Основная часть

Архитектура системы построена как последовательный управляемый конвейер. На первом этапе языковая модель получает запрос пользователя на естественном языке и формирует черновой граф диаграммы в структурированном формате, содержащий узлы, рёбра, тип средства визуализации и метаданные компоновки. На втором этапе отдельный модуль-критик оценивает соответствие черновика исходному заданию по нескольким осям (оценка соответствия задаче, визуальная оценка, общая оценка), выявляет пропущенные требования, неверные интерпретации и визуальные проблемы, после чего формирует структурированный план исправлений с приоритизированными правками.

На третьем этапе модуль улучшения применяет план исправлений к черновику и возвращает исправленный граф. Затем модуль верификации проверяет, что каждый пункт критики действительно учтён (статусы: исправлено, частично исправлено, проигнорировано). Если после первого прохода улучшения остаются проигнорированные или частично применённые правки, автоматически запускается дополнительный корректирующий проход только по нерешённым пунктам. Такой цикл самокритики позволяет системе исправлять собственные ошибки без участия пользователя.

Четвёртый этап - анализ значимости критики. Для каждого запуска агент вычисляет метрики изменения диаграммы (показатель изменений, прирост числа узлов и рёбер) и метрики следования критике (показатель согласованности с критикой, доля проигнорированных замечаний). Применение метода Шепли к суррогатной регрессионной модели, обученной на накопленной истории запусков, позволяет количественно оценить, какие аспекты критики в наибольшей степени определяют структурные изменения финальной диаграммы. Дополнительно реализован А/Б-контрфактический прогон: параллельно запускается проход улучшения с нейтральной критикой (общая оценка 1,0, пустые списки проблем), что позволяет изолировать эффект реальной критики от базового поведения модели [3].

На пятом этапе финальный граф маршрутизируется в одно из трёх средств визуализации в зависимости от типа диаграммы: Graphviz для схем общего и конвейерного типов, PlotNeuralNet для трёхмерных архитектур нейронных сетей, модуль инфографики

(векторная графика в растровый или PDF-формат) для инфографических схем. Поддержка нескольких средств визуализации в рамках единого агента устраняет необходимость переключения между инструментами при работе с разнородными типами диаграмм.

В качестве задачи апробации была выбрана генерация архитектуры U-Net - свёрточной сети для сегментации биомедицинских изображений, предложенной в работе [6]. Эталоном служила оригинальная схема архитектуры из первоисточника (рис. 1): система должна была воспроизвести её по текстовому описанию задачи, включающему 23 блока, пропускающие соединения и U-образную симметричную геометрию энкодера и декодера.

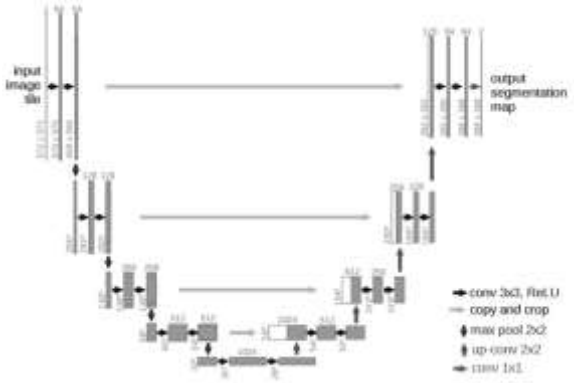


Рис. 1. Эталонная архитектура U-Net

Диаграмма, полученная в результате работы граф-агента (рис. 2), сравнивалась с эталоном визуально. В качественных экспериментах установлено, что цикл самокритики устраняет типичные ошибки однопроходной генерации: пропуски обязательных блоков, неверные направления рёбер, нарушения симметрии декодера.

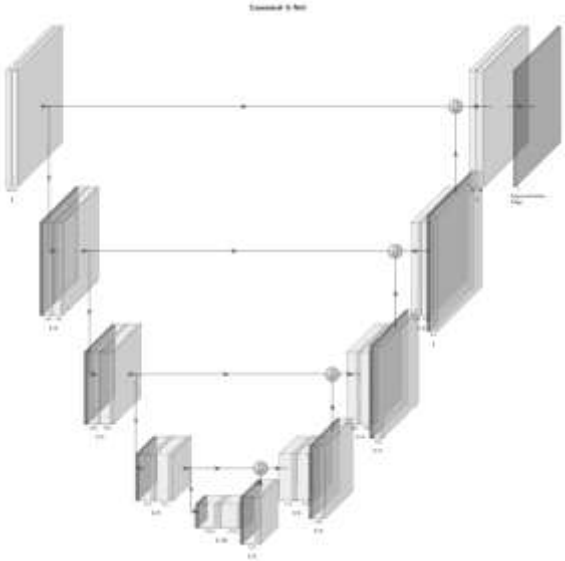


Рис. 2. Диаграмма U-Net, сгенерированная граф-агентом

Выводы

Представленный граф-агент демонстрирует, что итеративный цикл самокритики (генерация - критика - улучшение - верификация) позволяет системе на основе большой языковой модели существенно повышать качество структурированного вывода по сравнению с однопроходной генерацией. Интеграция анализа значимости критики и А/Б-

контрфактического прогона переводит оценку влияния критика из качественного наблюдения в измеримую количественную характеристику, что создаёт основу для сравнительного исследования различных стратегий критики. Поддержка четырёх типов диаграмм с тремя средствами визуализации в рамках единого конвейера делает систему применимой к широкому классу задач - от системных архитектур до нейросетевых моделей. В перспективе планируется расширение набора метрик оценки качества диаграмм и исследование влияния выбора модели-критика на сходимость цикла улучшения.

Список литературы

1. Wei J., Wang X., Schuurmans D., Bosma M., Ichter B., Xia F., Chi E., Le Q., Zhou D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models // Advances in Neural Information Processing Systems. — 2022. - Vol. 35. - P. 24824–24837.
2. Shinn N., Cassano F., Gopinath A., Narasimhan K., Yao S. Reflexion: Language Agents with Verbal Reinforcement Learning // Advances in Neural Information Processing Systems 36 (NeurIPS 2023). - Neural Information Processing Systems Foundation, 2023.
3. Lundberg S. M., Lee S.-I. A Unified Approach to Interpreting Model Predictions // Advances in Neural Information Processing Systems. - 2017. - Vol. 30.
4. Gansner E. R., North S. C. An open graph visualization system and its applications to software engineering // Software: Practice and Experience. - 2000. - Vol. 30. - P. 1203–1233.
5. Iqbal H., Aguerrebere C. PlotNeuralNet: Latex code for drawing neural networks for reports and presentation [Электронный ресурс]. - URL: https://github.com/HarisIqbal88/PlotNeuralNet (дата обращения: 18.04.2026).
6. Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation // Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015. Lecture Notes in Computer Science. - Springer, Cham, 2015. - Vol. 9351. - P. 234–241. - doi: 10.1007/978-3-319-24574-4_28.

МЕТОД ОБЪЯСНИМОЙ ДЕТЕКЦИИ ПАТОЛОГИЙ НА МАММОГРАФИЧЕСКИХ ИЗОБРАЖЕНИЯХ С ИСПОЛЬЗОВАНИЕМ GRADIENTSHAP И MEDSAM

Трусов[1] И. А., Ким Т. В., Кондрашова[1] Е. С., Стаценко[1] Г. Э., Трофимов[1] Ю. В., Аверкин[1] А. Н.

3. Федеральное государственное бюджетное образовательное учреждение высшего образования «Университет «Дубна»

Аннотация. В работе рассматривается задача повышения интерпретируемости нейросетевых методов анализа маммографических изображений. Предлагается подход, объединяющий градиентные атрибуции *SHAP* и сегментационную модель *MedSAM* для построения области патологического изменения, согласованной с решением модели. Показано, что преобразование карт значимости в пространственные подсказки позволяет получить более точную и клинически интерпретируемую локализацию.

Ключевые слова: объяснимый искусственный интеллект, маммография, локализация патологий, MedSAM, сегментация, метод Шепли

Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема № 124112200072-2).

Введение

Технологии искусственного интеллекта широко применяются в задачах медицинской диагностики и активно внедряются в системы поддержки принятия решений. В частности, методы компьютерного зрения позволяют автоматически обнаруживать патологические изменения на медицинских изображениях. Однако результаты работы таких моделей часто

воспринимаются как результат функционирования «чёрного ящика», что ограничивает их практическое применение в клинической среде.

Для медицинской практики недостаточно получения только итогового решения модели; врачу необходимо понимать, какие области изображения повлияли на результат и где именно локализовано патологическое изменение. В связи с этим активно развивается направление объяснимого искусственного интеллекта, направленное на повышение прозрачности и доверия к моделям [1].

В настоящей работе предлагается метод объяснимой локализации патологических изменений, основанный на использовании градиентных атрибуций *SHAP* [1] и уточняющей сегментации с применением модели *MedSAM* [2]. На первом этапе нейросетевая модель выполняет детекцию подозрительной области на маммографическом изображении. Далее для выделенного участка вычисляются карты *GradientSHAP*, отражающие вклад отдельных пикселей в итоговое решение модели.

Постановка задачи

В качестве исходных данных используется открытый набор маммографических изображений *VinDr-Mammo*, содержащий размеченные исследования и предназначенный для задач детекции и классификации патологий молочной железы [3]. В качестве архитектуры модели детекции выбрана *YOLOv8n*.

Пусть задано изображение:

$$x \in \square^{H \times W},$$

и нейросетевая модель $f(x)$, определяющая вероятность наличия патологического изменения.

Таким образом, задача сводится к построению преобразования:

$$x \to f(x) \to GS(x) \to MS(x),$$

где $GS(x)$ – карта значимости, а $MS(x)$ – уточнённая сегментация за детектированной области.

Метод объяснимой детекции

На первом этапе используется детектор патологических изменений, который выделяет на изображении область, потенциально содержащую новообразование или иное отклонение. Модель *YOLOv8n* обучается на наборе данных *VinDr-Mammo* [3]. После этапа детекции область интереса передаётся на этап интерпретации.

Для объяснения решения модели применяется метод *GradientSHAP* - градиентная аппроксимация значений Шепли [1]. В общем виде вклад пикселя может быть записан как математическое ожидание:

$$GS_i(x) = \mathrm{E}_{x',\alpha,\varepsilon}\left[\left(x_i - x_i'\right)\cdot\frac{\partial f\left(x' + \alpha\left(x - x'\right) + \varepsilon\right)}{\partial x_i}\right],\ \alpha \sim U\left(0,1\right)$$

Полученная карта значимости отражает вклад каждого пикселя в решение модели. Однако в случае медицинских изображений такие карты часто характеризуются размытостью, фрагментацией и несоответствием реальным анатомическим границам.

Следует отметить, что использование сегментационных моделей семейства *Segment Anything* требует адаптации к медицинским данным и специфике клинических задач [2, 4]. Для устранения указанных недостатков карта значимости используется как основа для формирования пространственной подсказки для сегментационной модели *MedSAM* [2]. В отличие от стандартного подхода, где подсказки задаются вручную или случайным

образом, в предлагаемом методе используется структурированная область, полученная на основе атрибуций *GradientSHAP*.

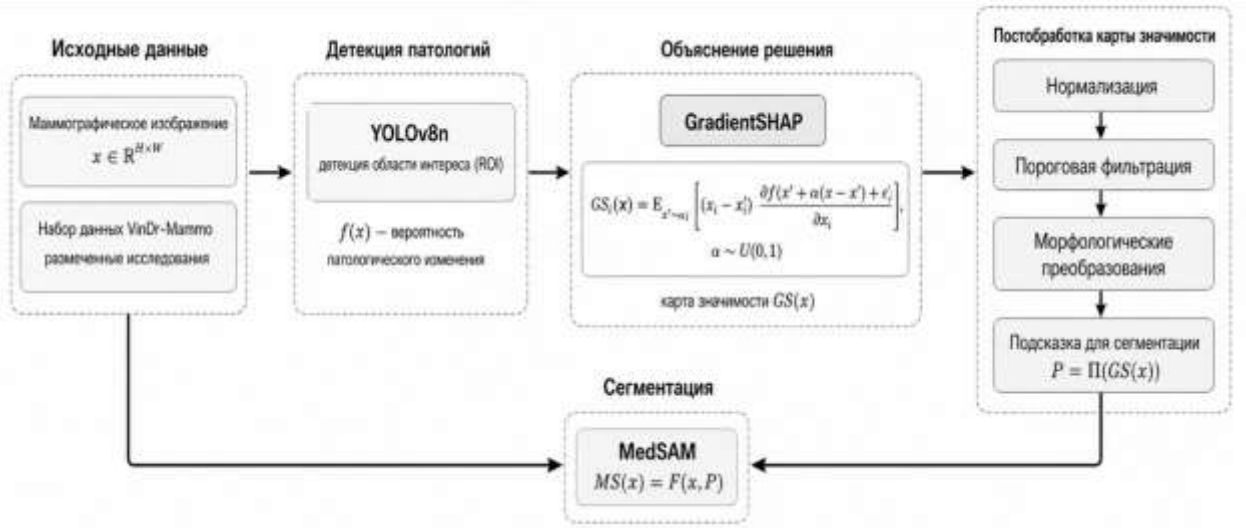


Рис. 1. Схема объяснимой детекции и сегментации патологий

Формально введём отображение преобразования карты значимости в подсказку:

$$\mathrm{P} = \Pi\left(GS\left(x\right)\right),$$

где $\Pi(\square)$ - оператор постобработки, включающий: нормализацию, пороговую фильтрацию и морфологический преобразования [5].

Тогда сегментационная модель задаётся отображением:

$$MS\left(x\right) = F\left(x, \mathrm{P}\right).$$

Метод объяснимой детекции в общем виде представлен на рисунке 1 и 2:

Выводы

В работе предложен метод объяснимой локализации патологических изменений на маммографических изображениях, основанный на совместном использовании *GradientSHAP* и сегментации. В отличие от стандартных подходов, карты значимости используются для визуальной интерпретации, как источник пространственной информации, направляющей процесс выделения области интереса.

Основной результат заключается в преобразовании размытых и фрагментированных атрибуций в связную геометрически осмысленную область, согласованную с решением модели. Это позволяет повысить устойчивость и интерпретируемость результатов локализации.

Предложенный подход может быть применён в медицинских системах анализа изображений, где важны прозрачность, воспроизводимость и доверие к результатам.

Список литературы

7. Lundberg S. M., Lee S.-I. A unified approach to interpreting model predictions // Advances in Neural Information Processing Systems. – 2017. – Vol. 30. – P. 4765–4774.
8. Ma J., He Y., Li F., et al. Segment anything in medical images // Nature Communications. – 2024. – Vol. 15. – Art. 654. – DOI: 10.1038/s41467-024-44824-z.
9. Nguyen H. Q., Lam K., Le L. T., et al. VinDr-Mammo: A large-scale benchmark dataset for computer-aided diagnosis in mammography // Scientific Data. – 2023. – Vol. 10. – Art. 277. – DOI: 10.1038/s41597-023-02084-1.

10. Kirillov A., Mintun E., Ravi N., et al. Segment Anything // Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). – 2023. – P. 4015–4026. – DOI: 10.1109/ICCV51070.2023.00371.

11. Универсальный метод локализаций объяснений SHAP / И. А. Трусов, Ю. В. Трофимов, Г. Э. Стаценко, А. Н. Аверкин // Инжиниринг предприятий и управление знаниями (ИП&УЗ-2025) : Сборник научных трудов XXVIII Российской научной конференции. В 2-х томах, Москва, 04–05 декабря 2025 года. – Москва: Российский экономический университет им. Г.В. Плеханова, 2025. – С. 344-350. – EDN GNWRZM.

ИССЛЕДОВАНИЕ МЕТОДОВ ПРЕДОБРАБОТКИ ДЛЯ СИГНАЛА ЭКГ ИЗ НАБОРА CASE В ЗАДАЧЕ ДЕТЕКТИРОВАНИЯ СТРЕССА С ИСПОЛЬЗОВАНИЕМ CNN

ЧЕГОДАЕВА Е.А.[1], ДОБРОХВАЛОВ М.О.[1], ЛИСС А.А.[1]

[1]Каф. МОЭВМ, СПБГЭТУ «ЛЭТИ», Санкт-Петербург, Россия

Аннотация. Данная работа заключается в сравнительном анализе значений accuracy и F1 при использовании чистых данных и данных с применением методов предобработки в рамках выявления 2 и 3 классов психофизического состояния посредством ResNet и FCN по сигналу ЭКГ. Проведен обзор смежной литературы на предмет самых применимых подходов к предобработке, в рамках которого определены размеры окон сигнала (10 и 60 сек.) и ряд методов: масштабирование, фильтры частотной области и изменение частоты дискретизации. Сигнал был извлечён из датасета CASE. По результату определено, что методы снижение ЧД позволяют получить наибольший прирост значений, полосовые фильтры оказывают наибольшее негативное воздействие, при применении методов масштабирования — влияние неустойчиво.

Ключевые слова: Методы предобработки, Электрокардиограмма (ЭКГ), Сверточные нейронные сети (CNN), Классификация психофизического состояния, Стресс, Датасет CASE.

Введение

Применение сигналов ЭКГ широко распространено для выявления стрессового состояния посредством моделей CNN. Так как биомедицинские сигналы сильно подвержены различному шуму (физиологические особенности субъектов, особенности устройств записи сигналов и др.) [1] [2] [3], существует широкий набор подходов к предобработке. Оценка влияния методов предварительной обработки сигнала ЭКГ позволит выявить наиболее обобщённые и применяемые подходы, позволяющие повысить точность определения состояний при использовании глубокого обучения.

Структура исследования

Для использования в текущем исследовании был определен ряд методов, которые наиболее часто применялись непосредственно к сигналам ЭКГ в работах, направленных на классификацию психоэмоциональных состояний посредством моделей сверточных нейронных сетей: MinMaxScaler [4] [5], StandardScaler [6] [7] [8], Полосовой фильтр в диапазоне 1 - 50 Гц [9], Полосовой фильтр в диапазоне 0.5 - 45 Гц [10], Фильтр низких частот на 40 Гц [11], Фильтр низких частот на 60 Гц [8], Режекторный фильтр на 50 Гц [4] [6], Снижение частоты дискретизации (ЧД) до 128 Гц [8] [5] и Снижение ЧД до 256 Гц [4] [7]. Список был сокращен таким образом, чтобы исключить вариативность алгоритмов реализации методов. Также в рамках данной работы не рассматриваются методы, направленные на выявление метрик вариабельности сердечного ритма и построения спектрограмм для дальнейшего использования в качестве входных данных моделей. Вместе с этим, следует отметить то, что все изученные работы включают сегментацию сигнала с применением метода скользящего окна: наиболее распространенные размеры окон

составляют 10 секунд [4] [6] [7] [9] с перекрытием 5 (10_5) и 60 секунд [4] [7] [8] с перекрытием 15 (60_15).

В качестве источника данных выбран датасет CASE [12], содержащий записи биомедицинских сигналов в синхронизации с разметкой психоэмоционального состояния на основе значений валентности и возбудимости. В эксперименте приняло участие 30 человек, которым были показаны видеоролики с различным содержимым для вызова тех или иных состояний. ЧД исходных сигналов составляет 1000 Гц. Сигналы ЭКГ всех субъектов были предварительно извлечены и разделены на окна указанного размера. Для анализа выбраны следующие состояния: радостное состояние (I четверть согласно модели оценки состояния), состояние стресса (II четверть) и нейтральное состояние (IV четверть). Окна разделены на обучающую и тестовую выборки в отношении 80/20 соответственно, для определения стресса в рамках двух и трех классов. Следует отметить сильный дисбаланс классов: 72,5% выборки составляет нейтральное состояние, 12,5% стрессовое и 15% радостное.

Чтобы сохранить основное направление исследования выбраны классические архитектуры сверточных нейронных сетей с низкой структурной сложностью: FCN и ResNet, реализация которых приведена в работе [13]. Использованы следующие параметры обучения: оптимизатор Adam, функция потерь CrossEntropyLoss, размер пакета 16, скорость обучения: 0,001, число эпох: 100, планировщик скорости обучения ReduceLROnPlateau.

Проведение серии изолированных запусков моделей осуществлялось посредством программного кода, приведённого в GitHub-репозитории[1]. Итоговое сравнение заключается в сопоставлении метрики accuracy и F1 при тестировании моделей, обученных на чистых данных и данных, после применения установленных методов, в рамках бинарной и трехклассовой задач классификации. Запуск и оценка моделей осуществлялись в среде Google Colab.

Исследование

На рис.1 и рис. 2 приведены значения accuracy и F1 соответственно, полученные при тестировании моделей. Результаты бинарной задачи отмечены «2» перед наименованием примененных методов предобработки, трехклассовой задачи — «3».

[1] https://github.com/lizavetaChe/Research_CNN_preprocessing

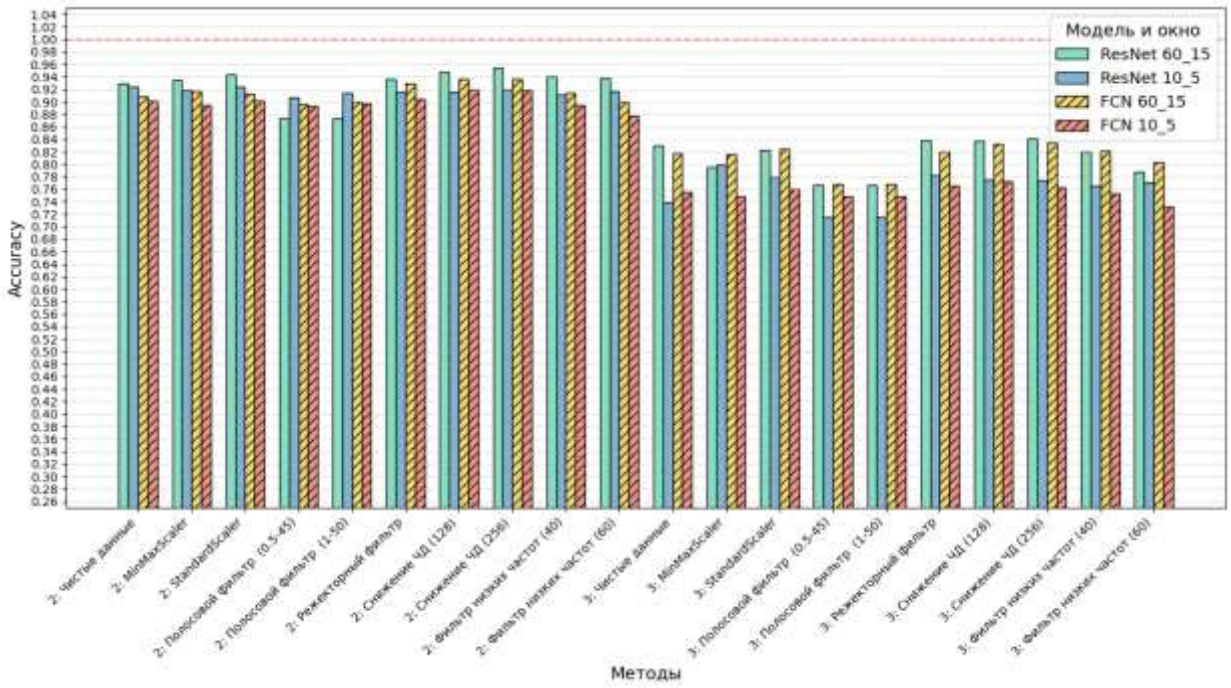


Рис. 1. Значения метрики accuracy для классификации состояний по 2-м и 3-м классам

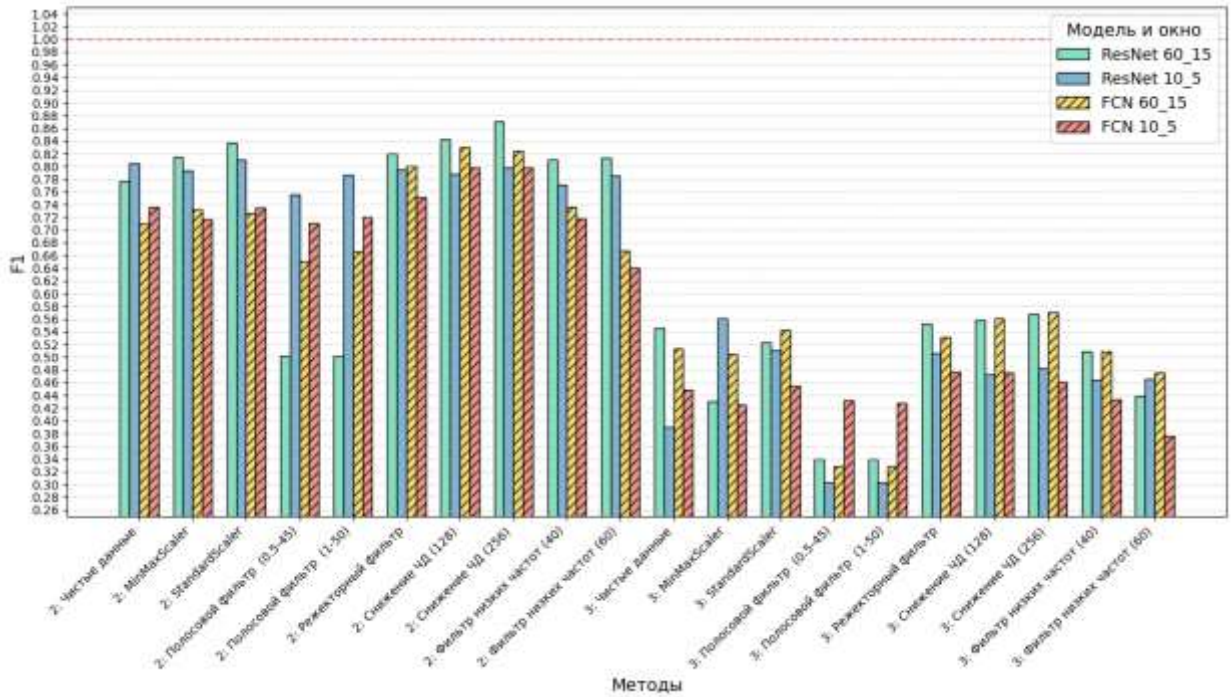


Рис. 2. Значения метрики F1 для классификации состояний по 2-м и 3-м классам

Результаты исследования

Первостепенно следует отметить критическую разницу между значениями accuracy и F1 для трёхклассовой задачи, что говорит об общем низком качестве классификации, однако при определении двух классов — разница менее значительна, что указывает на сильную уязвимость решений при дисбалансе классов. В рамках обеих задач можно отметить следующие положения по поводу влияния методов предобработки на результат:

Применение StandardScaler преимущественно демонстрирует рост метрик: accuracy в среднем возрастает на 0,48%-1,56%, F1 на 2,64%-8,36%. Использование MinMaxScaler

менее стабильно влияет на результат: в случае бинарной задачи и окна 60_15: рост accuracy составил 0,49%-0,84%, F1 — 3,11%-4,85%, для бинарной 10_5: accuracy падает на 0,60%-0,83%, F1 на 1,41%-2,71%. Для трехклассовой задачи — метод демонстрирует спад значений в среднем на 0,71% для accuracy и на 3,96% для F1.

Полосовые фильтры демонстрируют спад значений: accuracy на 2,13%-4,44%, F1 на 11,48%-25,12%. Фильтр низких частот на 60 Гц проявляет отрицательный эффект: accuracy падает на 0,9%-1,35%, F1 на 4,19%-5,97%. Фильтр на 40 Гц незначительно повышает метрики для окна 60_15 при бинарной задаче, однако понижает в остальных случаях, в среднем влияние отрицательное: accuracy уменьшается на 0,09%-0,90%, F1 на 0,79%-6,66%.

Режекторный фильтр демонстрирует прирост значений: accuracy в среднем на 0,60%-2,2%, F1 на 4,76%-10,17%.

Метод понижения ЧД до 128 Гц позволяет увеличить метрики: accuracy на 1,51%-2,43%, F1 на 7,92%-9,66%. При понижении ЧД до 256 также преимущественно можно отметить рост метрик: accuracy на 1,75%-2,30%, F1 на 8,95%-10,38%.

При обзоре архитектур следует отметить то, что ResNet в сравнении с FCN демонстрирует значения выше. Также наиболее эффективным оказалось окно 60_15.

Анализ пульсовой волны объема крови (BVP)

Помимо ЭКГ, аналогичное исследование было проведено для сигнала BVP, включая иные диапазоны применения методов. Полученные результаты не продемонстрировали процесса обучения при текущих подходах, исходя из этого обзор влияния методов предобработки сигнала BVP по данным CASE провести не предоставляется возможным.

Заключение

Таким образом, было выявлено влияние методов предварительной обработки сигнала ЭКГ на точность CNN в задаче определения стрессового состояния в рамках бинарной и трехклассовой задач. В качестве источника данных использован набор CASE, содержащий ряд биомедицинских сигналов и разметку состояний по значениям валентности и возбудимости. Определены подходы к предобработке данных для текущего исследования. Посредством реализованного программного кода проведена серия изолированных запусков архитектур ResNet и FCN, на данных без применения методов и после применения.

Необходимо подчеркнуть, что изначальный дисбаланс классов сильно повлиял на результат применения методов предобработки.

В соответствии со значениями accuracy и F1, наибольший прирост значений получен при применении снижения ЧД до 256 Гц: accuracy более 0,91%, F1 более 2,74%, при снижении ЧД до 128 Гц также отмечена положительная динамика. Чуть менее значительный рост получен при применении режекторного фильтра: более 0,36% по accuracy и более 1,92% по F1. Иные частотные фильтры демонстрируют спад значений: accuracy в среднем на 1,41%-2,37%, и F1 на 7,17%-13,50%. StandardScaler повышает метрики более, чем на 0,48% по accuracy и более 0,64% согласно F1, MinMaxScaler демонстрирует нестабильный результат.

Также следует отметить, что при использовании окна 60_15 значения выше, чем при 10_5: accuracy на 1,36% - 6,48%, F1 — на 1,87% - 8,30%. Вместе с этим, применение методов чаще оказывало положительный эффект при окне размером 60_15.

Дальнейшие исследования будут направлены на анализ иных методов предобработки и расширение числа задействованных архитектур.

Список литературы

1. Kang M. et al. Classification of Mental Stress Using CNN-LSTM Algorithms with Electrocardiogram Signals //Journal of Healthcare Engineering. – 2021. – Т. 2021. – №. 1. – С. 9951905.
2. Bhateja V. et al. A composite wavelets and morphology approach for ECG noise filtering //Pattern Recognition and Machine Intelligence: 5th International Conference, PReMI 2013, Kolkata, India, December 10-14, 2013. Proceedings 5. – Springer Berlin Heidelberg, 2013. – С. 361-366.
3. Kuttala R., Subramanian R., Oruganti V. R. M. Multimodal hierarchical CNN feature fusion for stress detection //IEEE Access. – 2023. – Т. 11. – С. 6867-6878.
4. Prajod P., André E. On the generalizability of ecg-based stress detection models //2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA). – IEEE, 2022. – С. 549-554.
5. Subhiyakto E. R. et al. Evaluation of Resampling Techniques in CNN-Based Heartbeat Classification //Igénierie des Systèmes d'Information. – 2024. – Т. 29. – №. 4.
6. Zhang P. et al. Real-time psychological stress detection according to ECG using deep learning //Applied Sciences. – 2021. – Т. 11. – №. 9. – С. 3838.
7. Cho H. M. et al. Ambulatory and laboratory stress detection based on raw electrocardiogram signals using a convolutional neural network //Sensors. – 2019. – Т. 19. – №. 20. – С. 4408.
8. Pidgeon M. et al. End-to-End Emotion Recognition using Peripheral Phys-iological Signals //35th International BCS Human-Computer Interaction Conference. – BCS Learning & Development, 2022. – С. 1-10.
9. He J. et al. Real-time detection of acute cognitive stress using a convolutional neural network from electrocardiographic signal //IEEE Access. – 2019. – Т. 7. – С. 42710-42717.
10. Awan A. W. et al. Advancing emotional health assessments: a hybrid deep learning approach using physiological signals for robust emotion recognition //IEEE Access. – 2024.
11. Hu Y. et al. Detection of Paroxysmal Atrial Fibrillation from Dynamic ECG Recordings Based on a Deep Learning Model. 2022, SSRN 4098696 [Электронный ресурс].
12. Sharma K. et al. A dataset of continuous affect annotations and physiological signals for emotion analysis //Scientific data. – 2019. – Т. 6. – №. 1. – С. 196.
13. Wang Z., Yan W., Oates T. Time series classification from scratch with deep neural networks: A strong baseline //2017 International joint conference on neural networks (IJCNN). – IEEE, 2017. – С. 1578-1585.

ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ СИСТЕМЫ ПОИСКА ПО ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ

ШЕЛЕПУГИН И.М., МАЛЮТИН Е.В.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. В статье выполнен обзор и анализ различных подходов к поиску по тексту: поиск по схожести, полнотекстовый поиск, семантический поиск, гибридный поиск. Выделены преимущества и недостатки рассмотренных подходов. С учетом полученных результатов, спроектирована и реализована распределенная система поиска по технической документации, позволяющая извлекать наиболее релевантные документы и фрагменты документов, применяя различные методы и параметры поиска.

Ключевые слова: системы искусственного интеллекта, обработка естественных языков, методы информационного поиска, распределенные системы.

Введение

Методы поиска текстовой информации применяются для решения широкого спектра задач. Они используются поисковыми движками для выдачи релевантных страниц, в корпоративных базах знаний и документообороте, в сфере разработки ПО, в юриспруденции, в медицине, в научных исследованиях и во многих других отраслях. Актуальность поиска текстовых данных растет с постоянным увеличением хранимых объемов информации, а также по мере развития технологий ее обработки. Так, методы извлечения текстовой информации нашли применение в системах искусственного

интеллекта: RAG (Retrieval-Augmented Generation), рекомендательных технологиях, ИИ-агентах [1].

В работе рассмотрены наиболее распространенные методы извлечения текстовой информации: поиск по схожести, полнотекстовый поиск, семантический поиск, гибридный поиск — в контексте системы поиска по технической документации.

Задача поиска текстовой информации состоит в нахождении записей, релевантных некоторому запросу [2]. Поскольку возможно существование нескольких релевантных записей, методы поиска ранжируют все записи на основе некоторой характеристики и возвращают первые N результатов. Чтобы избежать ложных результатов (например, при малом числе записей), на значение релевантности может быть наложено ограничение снизу. В этом случае возвращаются не более N результатов со значением релевантности выше данного ограничения.

Один из рассматриваемых методов поиска основан на N-граммах — подпоследовательностях символов или слов в исходном тексте. Записи ранжируются по количеству N-грамм, совпадающих с N-граммами запроса. В контексте поиска могут использоваться как последовательности слов, так и последовательности символов.

Наиболее распространенным методом поиска является полнотекстовый поиск (Full-text search, FTS). При использовании полнотекстового поиска записи преобразуются в структурированный формат с помощью токенизации, стемминга и фильтрации [3]. На этапе токенизации текст разбивается на отдельные слова. Стемминг — процесс приведения этих слов к их основе. Нормализованные слова фильтруются, убираются частые незначащие слова, например, артикли “a”, “the”. Аналогичные преобразования применяются к поисковому запросу, после чего система ищет записи с совпадающими словами, наборами и последовательностями слов.

Семантический поиск — современный метод извлечения информации, который учитывает не только синтаксический состав текста, но и его смысл [4]. Для исходного текста (или его частей) и запроса с помощью специальной модели генерируются эмбеддинги — семантические векторы. Векторы записей ранжируются по схожести с вектором запроса. В случае с текстовыми эмбеддингами, как правило, используется косинусное сходство.

Системы могут использовать несколько методов поиска для одного и того же запроса и комбинировать полученные результаты. Сочетание полнотекстового и семантического поиска получило название гибридный поиск. Релевантность результатов определяется как функция от значений релевантности, полученных двумя методами. Например, может использоваться среднее арифметическое взвешенное этих величин или взаимное ранговое слияние (Reverse reciprocal fusion, RRF) [5].

Неотъемлемой частью текстового поиска является индексация данных. Индексы позволяют существенно сократить время обработки запроса, поэтому их использование необходимо для поиска по большому объему информации. Широко применяется инвертированный индекс. Он представляет собой ассоциативный массив ссылок на записи, в которых содержится данное слово [6].

Предлагаемый подход

Система поиска по технической документации представляет собой программный комплекс, применяющий методы извлечения текстовой информации для поиска по документации приложений, фреймворков и библиотек, справочникам API, спецификациям и стандартам. Она реализует два основных сценария использования: управление

документами (создание, изменение, удаление) и поиск по ним. Было принято решение применить поиск с использованием триграмм (поиск по схожести), полнотекстовый, семантический и гибридный поиск по отдельным фрагментам документов (чанкам). Связанные документы объединяются в рабочие пространства.

При загрузке документа его необходимо разделить на чанки, для каждого чанка сформировать эмбеддинг и сохранить чанки в базу данных. Протокол HTTP подразумевает отправку ответа на запрос, т.е. системе необходимо дождаться полной индексации документа и только после этого отправить заголовки и тело ответа. Однако документ может иметь большой размер, кроме того, задача генерации эмбеддингов для каждого чанка является трудоемкой. Поэтому необходимо наличие очереди на индексацию документов. Ответ на HTTP-запрос в такой модели сообщает о статусе документа.

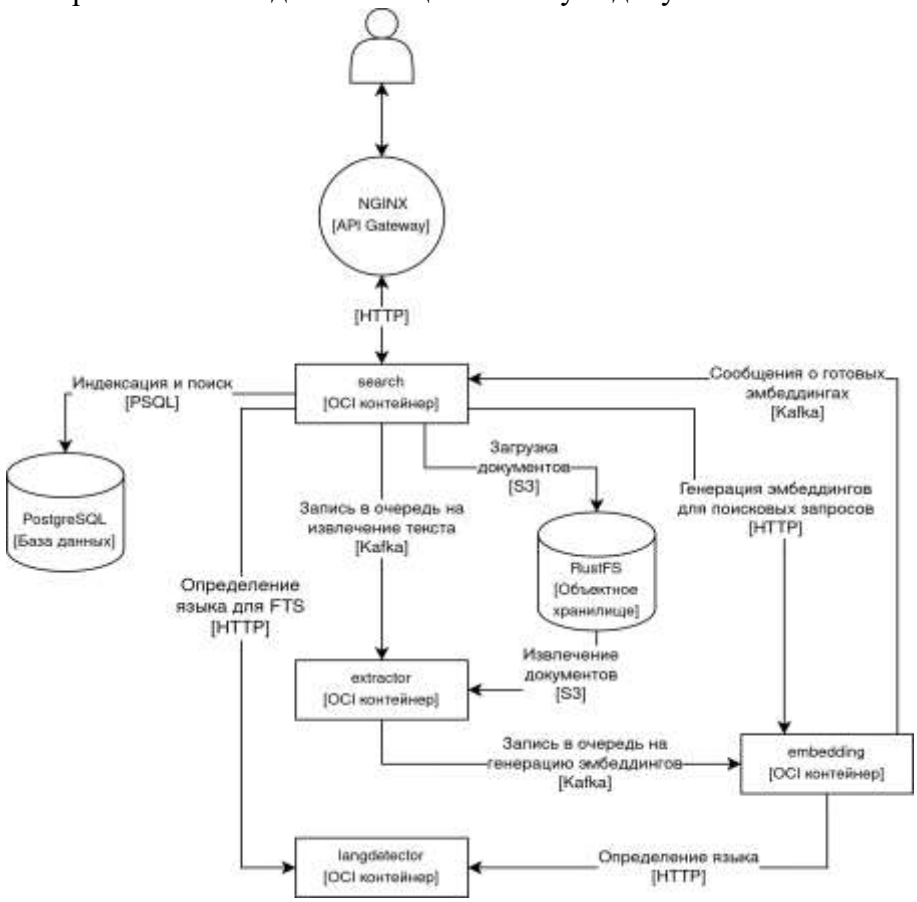


Рис. 1. Распределенная архитектура системы поиска по документации

Различные этапы обработки документа подразумевают разный вид нагрузки. Генерация эмбеддингов использует CPU или GPU и является вычислительной задачей. Обработка запросов и запись в базу данных являются задачами ввода-вывода. Разбиение документа на чанки — комбинированная задача. Кроме того, данные задачи используются неравномерно, поскольку более частым сценарием использования является поиск, в котором разбиение текста не задействуется.

Учитывая данные особенности, было принято спроектировать распределенную систему. Для поиска микросервисы взаимодействуют синхронно посредством REST API, для индексации — асинхронно, с помощью брокера сообщений Apache Kafka. Высокоуровневая схема архитектуры системы представлена на рисунке 1.

Поскольку документы могут иметь большой размер, прямая передача через брокер сообщений неэффективна. Система использует средство хранения документов в виде объектного хранилища RustFS, совместимого с AWS S3. При постановке документа в очередь в сообщении передается лишь идентификатор соответствующего объекта.

Методы поиска реализованы с помощью встроенной и дополнительной функциональности СУБД PostgreSQL. Для поиска по схожести используется расширение pg_trgm, релевантность определяется по соответствию триграмм отдельных слов. Полнотекстовый поиск основан на встроенном типе данных tsvector, для улучшения точности в таблице хранится информация о языке чанка. Семантический поиск реализован с помощью расширения pgvector и использует косинусное сходство между эмбеддингами для ранжирования. В качестве индексов для поиска по сходству и полнотекстового поиска используется GIN (Generalized Inverted Index), семантический поиск оптимизирован с помощью алгоритма HNSW (Hierarchical navigable small world) [7].

Генерация текстовых эмбеддингов реализована с помощью sentence-transformers [8]. Реализована поддержка мультиязычных моделей, а также возможность выбора модели в зависимости от языка входного текста.

Разработанная система реализует REST API для управления документами и поиска по ним. На рисунке 2 представлен пример использования API для осуществления гибридного поиска. Для демонстрации используется интерфейс, автоматически сгенерированный на основе спецификации OpenAPI.

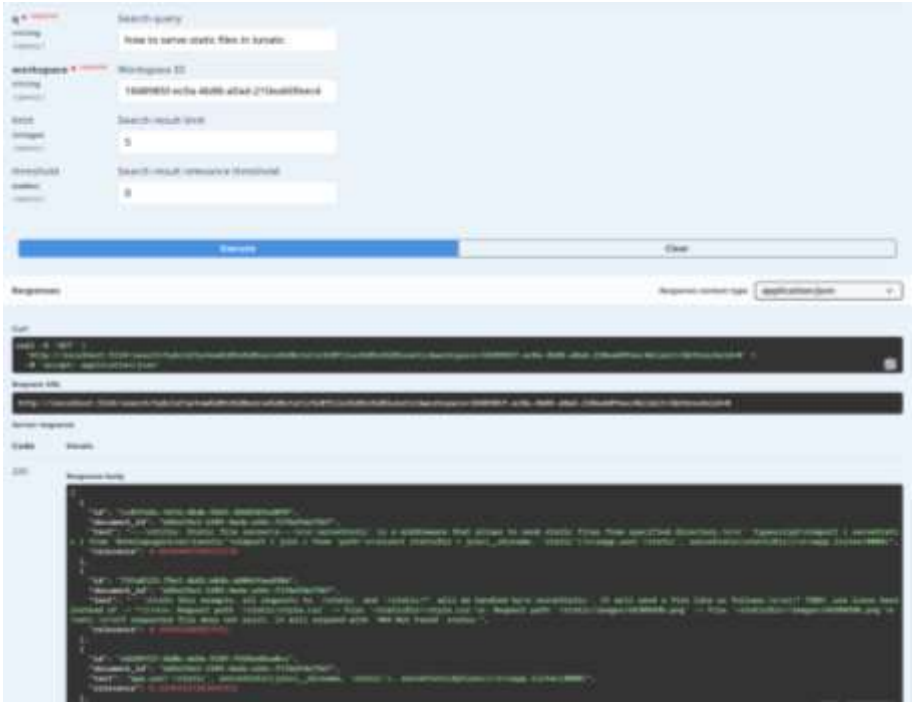


Рис. 2. Работа REST API разработанной системы

Доступность кода

Исходный код системы доступен по ссылке: https://github.com/shelepuginivan/carmen.

Заключение

Рассмотрены основные методы поиска текстовой информации. Спроектирована и реализована распределенная система поиска по технической документации, использующая приведенные методы.

Направлениями дальнейшей разработки системы станут: внедрение предобработки пользовательских запросов, использование моделей класса cross-encoder для дополнительного ранжирования отобранных чанков, интеграция системы мониторинга, реализация RAG на основе разработанного программного комплекса.

Список литературы

1. Patel B., Belli D., Jalalirad A., Arnold M., Ermolov A., Major B. Dynamic Tool Dependency Retrieval for Efficient Function Calling // arXiv preprint arXiv:2512.17052. – 2025.
2. Курейчик, В. В. Основные подходы к извлечению текстовой информации (обзор) / В. В. Курейчик, П. С. Герасименко. — Текст : непосредственный // Известия ЮФУ. Технические науки. — Таганрог : Южный Федеральный Ун-т, 2024. — С. 6-14.
3. Full-text search explained. — Текст : электронный // Google Cloud : [сайт]. — URL: https://cloud.google.com/discover/what-is-full-text-search (дата обращения: 07.04.2026).
4. What is semantic search?. — Текст : электронный // Elastic : [сайт]. — URL: https://www.elastic.co/what-is/semantic-search (дата обращения: 07.04.2026).
5. Katz, Jonathan Hybrid Search With PostgreSQL and Pgvector / Jonathan Katz. — Текст : электронный — URL: https://jkatz05.com/post/postgres/hybrid-search-postgres-pgvector/ (дата обращения: 07.04.2026).
6. Baeza-Yates, Ricardo Modern Information Retrieval / Ricardo Baeza-Yates, Ribeiro-Neto. — New York : ACM Press, 1999. — 499 с. — Текст : непосредственный.
7. Malkov Y. A., Yashunin D. A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs //IEEE transactions on pattern analysis and machine intelligence. – 2018. – Т. 42. – №. 4. – С. 824-836.
8. Reimers, Nils Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks / Nils Reimers, Iryna Gurevych. — Association for Computational Linguistics, 2019. — Текст : непосредственный.

НЕЙРОСЕТЕВАЯ РЕКОМЕНДАЦИЯ ОБРАЗА ДНЯ ПО ПОСЛЕДНИМ МУЗЫКАЛЬНЫМ ПРЕДПОЧТЕНИЯМ

Ширяева М.Д.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» имени В.И. Ульянова (Ленина)

Аннотация. На сегодняшний день существует множество систем практически во всех отраслях, которые на основе искусственного интеллекта подбирают персонализированную информацию. Однако не так много моделей, объединяющих различные сферы, которые потенциально могут иметь хорошую совместимость. В данной статье предлагается рассмотреть нейросеть, которая способна сочетать музыку и моду. На основе последних прослушиваемых пользователем композиций она может подобрать наиболее подходящий наряд на день. Представлена архитектура сети на основе векторов, используемые метрики и механизмы объяснимости, а также указаны трудности и актуальность данной системы. В результате анализа и сравнения с существующими технологиями установлено, что аналогичной нейросети до сих пор не разработано, а это напрямую указывает на своевременность ее внедрения.

Ключевые слова. Рекомендательные системы, музыкальные предпочтения, объяснимый искусственный интеллект, нейронные сети, динамическая персонализация.

С геометрической прогрессией растет количество людей, использующих искусственный интеллект в повседневной жизни. Теперь это не только инструмент для профессиональной и образовательной деятельностей, но и неотъемлемая часть нашей рутины. С помощью него мы получаем советы на любую волнующую нас тему: что лучше съесть на завтрак, как провести выходные, куда лучше съездить отдохнуть, что прочитать в свободный вечер или что послушать, когда грустно или весело. Все больше компаний внедряют эти алгоритмы в свои продукты, следуя трендам на рынке. Такая тенденция

коснулась и сферы музыки, где передовой технологией являются глубокие нейронные сети, адаптирующие активность слушателей в непрерывный поток треков, подобранных по их предпочтениям. Это внедрение, например, в сервисе «Яндекс Музыка», привело к увеличению количества подписчиков на 32% по сравнению с годом ранее. Тем самым, своевременное внедрение передовых технологий напрямую влияет на успешное продвижение компании. Аналогично происходит и в индустрии моды, где искусственный интеллект помогает с генерацией дизайнов, примеркой одежды и персональным подбором образов. В связи с тем, что во всех ведущих областях эти технологии активно применяются, следующим этапом является объединение отраслей. Такие инновации могут послужить созданию новых рынков и открыть еще больше возможностей для расширения. В настоящее время развитие сфер музыки и моды довольно статичны, не хватает персонализированного, творческого подбора для целевой аудитории, которая чувствует окружающий мир более образно, метафорически. Отталкиваясь от потребностей таких людей, можно сделать вывод о том, что им не хватает инструмента, который объединял бы эти два способа самовыражения. Эта технология позволила бы расширить границы их восприятия, а также помочь с поддержкой творческих успехов в периоды усталости, грусти или отсутствия времени. Таким инструментом может быть нейросеть, подбирающая образ дня по последним прослушиваемым трекам.

Изучив существующие системы, похожие на это предложение, можно сделать вывод о том, что они либо очень медленно развиваются, либо не развиваются вовсе. Так, приложение FITS, выпущенное в 2018 году, ограниченно ассортиментом одного бренда, а также оно дает рекомендации только на основе всего профиля, а не актуальной на последние сутки информации, что является ключевым моментом. Другое приложение под названием AI StyleTwin, релиз которого состоялся уже в 2025 году, уже ближе к задуманной технологии, но оно зависит от другого сервиса Qloo API, откуда и берет информацию, сопоставляющую образы фанатов разных жанров. Это тоже не является оптимальным решением, так как условия стороннего приложения в любой момент могут измениться, что повлечет изменение всей структуры. Помимо существующих платформ активно проводятся исследования на эту тему. К примеру, статья «Music-triggered fashion design: from songs to the metaverse» предстает наиболее близким аналогом данной разработки. В ней авторы как раз таки сопоставляют музыку и моду с помощью динамики, они предлагают учитывать все звуковые эффекты для дальнейшего преобразования их в визуальные образы. Однако это исследование находится на начальном этапе и не представляет собой конкретных технологий, а скорее дает лишь рекомендации. На основе опыта всех этих аналогов можно составить более четкий и современный образ о том, как это реализовать на существующих платформах.

Переходя к точному описанию данного алгоритма, для начала стоит выделить входные и выходные данные. Ими могут послужить все треки, прослушиваемые пользователем за последние 24 часа, чтобы рекомендация стала наиболее актуальной. Упомянутая область для многих пользователей неразрывно связана с их настроением и ощущением в пространстве, что также напрямую влияет и на предпочитаемый стиль в одежде. Обработав всю эту информацию, нейросеть сможет выдать визуальные образы подходящих нарядов, причем избегая эффекта «черного ящика», с обоснованием данного выбора по определенной формуле.

Чтобы обеспечить такую технологию необходимой базой данных, потребуется большой объем фотографий образов с тегами, которые можно будет взять из различных открытых

источников, таких как, например, DeepFashion2 или iMaterialist Fashion Attribute. Далее их нужно будет сопоставить с треками, внедряя данную нейросеть в существующий музыкальный центр, хорошим примером могла бы послужить «Яндекс Музыка», успешно использовавшая эту технологию для подбора персонализированных треков, этот процесс удобно было бы расширить с помощью дополнительной функции. Следующим важнейшим этапом является разметка данных, к чему могли бы быть подключены специалисты, способные при прослушивании 10-секундного отрывка выбрать наиболее подходящий образ. Эта база в дальнейшем послужила бы для обучения нейросети. Другим способом является действие от обратного, заключающееся в описании образа с картинки с помощью перечня жанров или известных треков. В итоге будет собрана база сопоставимых с музыкой образов, чей объем может постоянно увеличиваться.

Следующим этапом является техническая обработка полученных результатов. Для экономии ресурсов можно использовать существующую нейросеть, изучающую музыку и преобразующую ее в вектор размерностью 512, который включает в себя тембр, ритм, голос, эмоциональную окраску и т.д. Другой актуальный вектор такой же размерности может быть задействован для исследования одежды, однако его следует сделать обучаемым в отличие от предыдущего, чтобы он мог сопоставлять данные компоненты друг с другом. Для объединения этих векторов следует привлечь многослойный перцептрон, который может преобразовать их в единое пространство для дальнейшего сравнения напрямую. С помощью функции потерь InfoNCE возможно обучение модели на позитивных и негативных парах, основной задачей которой является сближение векторов наиболее подходящих образов для дальнейшего запоминания. Но база данных скорее всего будет довольно большая, учитывая сколько композиций пользователь может прослушать дальше, чтобы учесть их все вместе их также стоит свести в один вектор с учетом времени прослушивания: чем позднее трек был прослушан, тем он более актуальный. Вводятся веса и на основе них производится оценка среднего. И в конце концов происходит поиск в базе данных по объединенному вектору.

Проведя разработку следует перейти к ее контролю, который покажет, работает ли на практике данная инновационная идея и не является ли она просто генератором случайных подборок. Чтобы достичь этого стоит пропорционально разделить собранную базу данных, большую часть оставив на обучение, а другие две на валидацию и тестирование. После обучения модели стоит проверить, не заучила ли она готовые решения, при необходимости изменяя настройки, а также на абсолютно новых данных проверить, насколько строго она следует алгоритму. Для конкретной оценки работоспособности нейросети необходимо использовать метрики, в данном случае могут подойти такие как Recall@5, помогающая оценить, входит ли образ в первые 5 рекомендаций, и NDCG@K, которая уже учитывает позицию рекомендации по порядку, оценивая при этом ее место в списке. Оба этих способа помогают прийти модели к более близкому решению. Однако данные метрики используются на заранее размеченных данных, что может быть субъективно, поэтому также следует прибегнуть к пользовательскому исследованию, во время которого приглашенные люди без дополнительной информации должны выбрать между решением данной нейросети и простой системы, работающей по аналогии с жанром или образом артиста. При значительном перевесе в предпочтениях в сторону модели стоит считать ее успешной.

Немаловажной стадией является обоснование выбранного образа, необходимо показать пользователю, почему данная рекомендация является аргументированной. Это можно сделать с помощью уже ранее использованной базы данных, словесно описывающей образы

и стили, а также с помощью модели, описывающей музыкальную составляющую. Вывод может выглядеть так: «Сегодня Вы прослушивали энергичную музыку с элементами ирландского фолк-метала, в связи с чем предлагаем Вам сегодня надеть образ, сочетающий нео и традиционные наряды» На изображении может быть тканевый корсет, юбка с клетчатым принтом, неоновые легинсы и грубые берцы, дополнив образ металлическими кольцами и серьгами болотного цвета в виде кельтских крестов.

В связи со сложностью данной модели нельзя избежать и слабых мест, среди которых длительность и трудозатратность сбора необходимых данных и их разметки, риск закрепления стереотипов, важно отслеживать непредвзятость выборки, а также среди них можно указать отсутствие учета личных предпочтений, который в дальнейшем также могут быть внедрены. Несмотря на все это реализация данной нейросети имеет высокие шансы увенчаться успехом, так как пересечение моды и музыки всегда будет актуальной в творческой среде.

Список литературы

1. Приложение мужского бренда одежды подбирает вещи на основе вкусов в Spotify // Marketing Dive. URL: https://www.marketingdive.com/news/menswear-startup-tailors-app-to-pick-clothes-based-on-spotify-tastes/528978/ (дата обращения: 02.05.2026).
2. AI-стилист-двойник // Devpost. URL: https://devpost.com/software/ai-styletwin#updates (дата обращения: 02.05.2026).
3. Музыкально-обусловленный дизайн одежды: от песен до метавселенной // arXiv.org. URL: https://www.researchgate.net/publication/384699808_Music-triggered_fashion_design_from_songs_to_the_metaverse (дата обращения: 02.05.2026).
4. Многослойный персептрон // Вики Loginom. URL: https://wiki.loginom.ru/articles/multilayered-perceptron.html (дата обращения: 02.05.2026).
5. Метрики классификации и регрессии // Яндекс Образование. URL: https://education.yandex.ru/handbook/ml/article/metriki-klassifikacii-i-regressii (дата обращения: 02.05.2026).

ВЕБ-СЕРВИС АДАПТИВНОГО СЖАТИЯ КОНТЕКСТА УЧЕБНОЙ ЛИТЕРАТУРЫ ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА ОТВЕТОВ БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

Щедрин А.А., Чепасов Д.В.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. В работе рассматривается задача применения больших языковых моделей для работы с учебной литературой, объём которой превышает размер контекстного окна. Выполнен сравнительный анализ пяти методов сжатия контекста в RAG-системах и предложена архитектура веб-сервиса, комбинирующего семантическую фильтрацию, контекстно-зависимое сжатие и сохранение структурированных образовательных элементов.
Ключевые слова: Большие языковые модели, Сжатие контекста, RAG-системы, Образовательные приложения, LLMLingua, Векторный поиск.

Введение

Применение больших языковых моделей (LLM) в образовании ограничено двумя фундаментальными проблемами. Во-первых, модели подвержены галлюцинациям — генерации правдоподобной, но фактически неверной информации, что недопустимо при обучении. Во-вторых, при работе с длинным контекстом возникает эффект «потерянности

в середине», когда релевантная информация, расположенная не в начале и не в конце промпта, фактически игнорируется. Эмпирические исследования показывают, что эффективная длина контекста большинства моделей составляет 4 000–32 000 токенов, тогда как объём типичного учебника достигает 100 000–250 000 токенов.

Частичным решением является подход RAG (Retrieval-Augmented Generation), при котором ответ LLM формируется на основе извлечённых фрагментов проверенного источника. Это повышает фактологическую точность, но не устраняет ограничение контекстного окна. Для решения применяются методы сжатия контекста — алгоритмы, сокращающие объём подаваемого модели текста при сохранении его ключевой семантики. Целью работы является разработка веб-сервиса, реализующего адаптивное сжатие учебных текстов для повышения качества ответов LLM.

Сравнительный анализ методов сжатия контекста

Проведён сравнительный анализ пяти open-source методов сжатия контекста, отобранных по критериям доступности кода и совместимости с LangChain и LlamaIndex: LLMLingua-2, LongLLMLingua, RECOMP (extractive), Selective Context и EmbeddingsFilter. Сравнение выполнялось по трём критериям: степени сжатия, сохранению семантики и скорости обработки. LLMLingua-2 обеспечивает сжатие до 20-кратного при точности 95 % по ROUGE-L, однако не учитывает запрос пользователя. LongLLMLingua использует оценку перплексии с учётом запроса и даёт прирост точности на 21,4 % на датасете NaturalQuestions при 4-кратном сжатии. RECOMP достигает 17-кратного сжатия, но требует дообучения, а готовые веса доступны только для английских корпусов. Selective Context не требует обучения, но теряет качество при коэффициенте свыше 3-кратного, что критично для задач логического вывода. EmbeddingsFilter работает на CPU за 5–10 мс на документ, но не способен удалить избыточную информацию внутри отдельного чанка.

Обоснование выбранного решения

Ни один из рассмотренных методов не решает задачу сжатия образовательного контента полностью: все они тестировались на англоязычных датасетах и не предоставляют механизмов сохранения математических формул, блоков кода и определений как атомарных единиц. В разрабатываемом сервисе применяется комбинация трёх компонентов: EmbeddingsFilter для быстрой первичной фильтрации, LLMLingua-2 на базе мультиязычной модели mDeBERTa-v3-base для основного контекстно-зависимого сжатия с поддержкой русского языка и пользовательская логика, при которой фрагменты с формулами, блоками кода и определениями размечаются на этапе индексирования и исключаются из сжатия.

Архитектура сервиса

Сервис реализован на Python 3.11 по модульному принципу и состоит из пяти компонентов: модуля загрузки документов, векторного хранилища, модуля сжатия, LLM-клиента и оркестратора. При индексировании документ разбивается на фрагменты по 512 символов с перекрытием 50; текст извлекается PyMuPDF, а для отсканированных страниц применяется OCR через Apple Vision. Каждый чанк кодируется моделью multilingual-e5-small в вектор размерностью 384, нормируется по L2-норме и добавляется в индекс HNSW (hnswlib). Для фрагментов со структурированным содержимым применяется буст +0,1 к оценке релевантности.

Пайплайн обработки запроса поддерживает три режима: baseline (без контекста), full_rag (top-10 чанков без сжатия) и compressed (сжатие до 40 % объёма через LLMLingua-2). Качество ответов оценивается метриками Keyword Hit Rate и ROUGE-L. Сервис предоставляет веб-интерфейс на Streamlit и REST API на FastAPI с автоматической документацией OpenAPI.

Результаты

Разработанная архитектура обрабатывает учебные материалы объёмом до 500 страниц с задержкой менее 5 секунд, сохраняя ключевую образовательную информацию при сокращении объёма контекста в 3–6 раз. Практическая значимость заключается в возможности применения предложенной архитектуры при создании образовательных AI-ассистентов и в использовании методологии сравнения при выборе методов оптимизации RAG-систем для русскоязычных учебных материалов.

Список использованной литературы:

1. Liu N. F., Lin K., Hewitt J. et al. Lost in the Middle: How Language Models Use Long Contexts // Transactions of the Association for Computational Linguistics. — 2024. — Vol. 12. — PP. 157–173.
2. Pan Z., Wu Q., Jiang H. et al. LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression // Findings of ACL. — 2024. — PP. 963–981.
3. Jiang H., Wu Q., Luo X. et al. LongLLMLingua: Accelerating and Enhancing LLMs in Long Context Scenarios via Prompt Compression // Proceedings of ACL. — 2024. — PP. 1658–1677.
4. Xu F., Shi W., Choi E. RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation // Proceedings of ICLR. — 2024.
5. Li Y. Unlocking Context Constraints of LLMs: Enhancing Context Efficiency of LLMs with Self-Information-Based Content Filtering. — 2023. — arXiv:2304.12102.

ПРОГРАММНАЯ ИНЖЕНЕРИЯ И АВТОНОМНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ СИСТЕМЫ

АЛГОРИТМ ИНТЕЛЛЕКТУАЛЬНОГО АГЕНТА-ПЕШЕХОДА В СРЕДЕ UNREAL ENGINE 5 ДЛЯ ТРЕНАЖЕРНОЙ ПОДГОТОВКИ ОПЕРАТОРОВ БПЛА

АКСЕНОВ В.В.

Муромский институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования «Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых»

Аннотация. В статье представлен подход к созданию интеллектуального бота-пешехода в симуляторе БПЛА, имитирующего поведение человека в стрессовых ситуациях. Бот использует стандартные модули Unreal Engine 5 без применения машинного обучения. Описан алгоритм оценки угрозы на основе расстояния, скорости сближения и уровня шума с учетом препятствий. Реализована многоуровневая система реагирования: от состояния внимания до панического бегства. Все параметры поведения гибко настраиваются, что позволяет создавать реалистичные сценарии тренировки операторов.

Ключевые слова: симулятор БПЛА, Unreal Engine 5, интеллектуальный агент, уклонение от бпла, поведение пешехода, оценка угрозы, тренировка операторов

Введение

С развитием и массовым внедрением квадрокоптеров всё большее значение приобретает качественная подготовка операторов. Однако существующие тренажёры, как правило, опираются на статичные сценарии и не могут в полной мере имитировать непредсказуемое поведение живых людей, которые могут внезапно замереть, искать укрытие или хаотично бежать. Особенно важно отрабатывать действия в нестандартных и стрессовых ситуациях, которых сложно добиться на статичных тренажёрах. Это создаёт разрыв между тренировочной средой и реальностью, снижая готовность оператора к быстрым изменениям обстановки.

Для решения этой проблемы был разработан интеллектуальный агент-пешеход – искусственный интеллект внутри симулятора БПЛА, который ведёт себя как живой человек.

В работе описывается алгоритм работы бота-пешехода, работающий как непрерывный цикл восприятия и оценки угрозы, а также многоуровневая стратегия реагирования – от состояния настороженности до панического бегства. Ключевым элементом является вычисление числового показателя угрозы на основе расстояния до дрона, скорости его сближения, уровня шума и наличия укрытий. Все параметры модели вынесены в настройки, что позволяет инструктору гибко адаптировать поведение ботов под конкретные сценарии тренировок оператора.

Вся работа выполнена в среде Unreal Engine 5 с использованием её стандартных систем ИИ, что позволило обойтись без дорогостоящего сбора данных и машинного обучения. Unreal Engine 5 предоставляет готовые модули для восприятия (Perception System), поиска укрытий (Environment Query System) и управления логикой (Behavior Tree).

Алгоритм работы бота

Опишем алгоритм работы бота. В основе построения алгоритма заложен постулат о том, что бот сам «слышит» и «видит» дрон, оценивает опасность и выбирает подходящую стратегию: замереть на месте, поискать укрытие или броситься наутёк.

Работа бота построена как непрерывный цикл, который удобно представить в виде блок-схемы (рисунок 1). В начале каждого цикла бот находится в состоянии патрулирования — он спокойно движется по сцене, не ожидая угрозы. Его сенсоры (зрение и слух) постоянно опрашивают окружение. Зрительный канал имеет ограниченный угол обзора (примерно 140 градусов) и максимальную дальность (до 80 метров). Он имитирует человеческий глаз: бот не видит, что происходит у него за спиной. Звуковой канал, наоборот, круговой: он слышит звуки со всех направлений, но громкость зависит от расстояния и наличия препятствий. Как только появляется стимул, например шум моторов (даже если дрон ещё не виден) или прямое визуальное обнаружение дрона, система переходит к вычислению числового показателя опасности T (от 0 — безопасно, до 1 – критично).

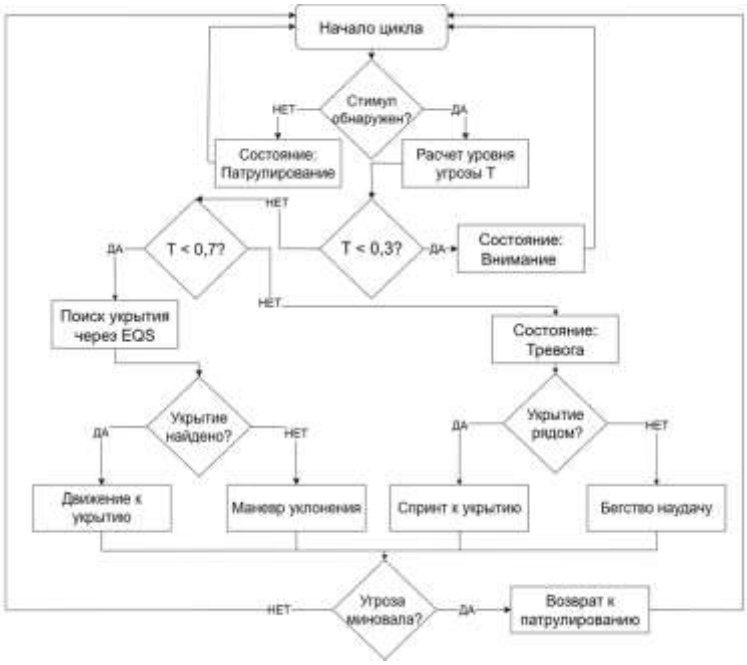


Рис. 1. Схема работы бота

В расчёт идут три фактора: расстояние до дрона, скорость его сближения и уровень шума. Кроме того, если между ботом и дроном есть непрозрачное препятствие (стена, дерево, забор), опасность снижается – бот чувствует себя защищённым. Поэтому в случае обнаружения дрона, бот начинает искать такие препятствия и использует их в качестве укрытия. Если таких в радиусе поиска несколько, то выбирается ближайшее укрытие. Если доступных укрытий нет, то бот начинает двигаться хаотично, что создаёт непредсказуемую траекторию, и дрону сложнее удержать бота в прицеле. Устаревшие стимулы (старше двух секунд) отбрасываются, чтобы бот не боялся давно улетевшего дрона. Это важно, потому что иначе бот мог бы, например, зависнуть в панике на несколько секунд после того, как дрон уже скрылся за горизонтом.

Для поиска ближайшего укрытия используется система пространственных запросов EQS, которая ищет лучшее укрытие в радиусе до 30 метров - EQS генерирует сетку точек вокруг бота (150 точек), для каждой точки вычисляет рейтинг по нескольким критериям: как далеко точка от дрона (чем дальше, тем лучше), есть ли прямая видимость от дрона до точки (если нет — хорошо), и как быстро бот может до точки добраться по навигационной сетке NavMesh. В итоге выбирается точка с наивысшим рейтингом. Если EQS нашла такое укрытие, то бот ускоренным шагом движется к нему. Если же ни одна точка не удовлетворяет минимальным требованиям (например, всё простреливается), то бот выполняет манёвр уклонения: он начинает двигаться зигзагом, делая резкие повороты на 45–90 градусов перпендикулярно текущему направлению на дрон. Это затрудняет прогнозирование траектории и снижает вероятность «попадания» в случае, если дрон имитирует атаку.

Вычисление уровня угрозы T осуществляется по формуле взвешенной суммы:

$$T = \alpha * f_D(D) + \beta * f_V(V_{rel}) + \gamma * f_N(N)$$

где α,β,γ — коэффициенты важности каждой компоненты. По умолчанию они равны 0,5; 0,3 и 0,2 соответственно, то есть расстояние вносит основной вклад в ощущение опасности

(ровно половину), скорость сближения – чуть меньший (30%), а звук служит ранним предупреждением и даёт 20% вклада.

Все эти коэффициенты легко менять в конфигурационном файле, подстраивая бота под разные сценарии. Например, для тренировки в шумном городском окружении можно уменьшить вес звука, а для ночных сценариев, где дрон хуже виден, увеличить вес слуха

Функция близости $f_D(D)$ зависит от расстояния $D = \|P_{bot} + P_{drone}\|$. Она определяется по формуле:

$$f_D(D) = \max(0{,}1 - \frac{D}{D_{max}})$$

где D_{max} — максимальная дистанция, на которой бот вообще реагирует (обычно 50–80 метров). Если дрон находится прямо над головой ($D = 0$), функция равна 1; если на пределе дальности – 0; на промежуточных расстояниях угроза нарастает линейно, так как именно такая функция даёт плавное и предсказуемое изменение угрозы, что близко к поведению человека. К тому же линейную функцию проще отлаживать и настраивать.

Радиальная скорость сближения $V_{rel} = \frac{\partial D}{\partial t}$. Положительное значение означает, что дрон приближается, отрицательное — удаляется. Опасность возникает только при сближении, поэтому:

$$f_V(V_{rel}) = \begin{cases} 0, V_{rel} \le 0, \\ \min\left(1, \frac{V_{rel}}{V_{max}}\right), V_{rel} > 0. \end{cases}$$

Параметр V_{max} – порог, выше которого скорость считается максимально угрожающей (например, 10 м/с для типичного квадрокоптера). Если дрон медленно приближается, вклад пропорционален скорости; если быстро – функция равна единице; а если улетает – угроза не увеличивается. Благодаря этому бот не паникует, когда дрон просто пролетает мимо. Если дрон летит параллельно на дистанции 30 метров, не меняя расстояния, то $V_{rel} = 0$, вклад скорости равен нулю, и бот реагирует только на близость. Если при этом дрон внезапно развернётся и полетит прямо на бота, V_{rel} станет положительным, и угроза возрастёт дополнительно.

Громкость N (в условных единицах, получаемая от слухового сенсора) сравнивается с эталонным уровнем N_{ref} :

$$f_N(N) = \min(1, \frac{N}{N_{ref}})$$

Это позволяет боту услышать дрон из-за угла или спиной, ещё до того, как он появится в поле зрения, как это и происходит в реальной жизни. Например, дрон приближается сзади: бот его не видит, но слышит нарастающий гул. Уровень T увеличивается за счёт звуковой компоненты, и бот может повернуться или начать уклоняться, даже не видя угрозы. Это резко повышает реалистичность.

После расчёта базового T, алгоритм применяет поправку на препятствия. Выстраивается луч (LineTrace) от головы бота до центра дрона. Если луч пересекается с любой статической непрозрачной геометрией, значит, между ними есть препятствие.

В этом случае итоговая угроза равна:

$$T_{final} = k_{occ} \cdot T.$$

Коэффициент $k_{occ} = 0{,}3$. Эксперименты показали, что 0,3 даёт хороший баланс: остаточная настороженность сохраняется, но паника не возникает. Если полностью обнулить угрозу ($k_{occ} = 0$), бот будет стоять за тонкой стеной и вообще игнорировать дрона,

что нереалистично. Если взять слишком высокий коэффициент (близкий к 1), например, k_{occ} = 0,8, бот будет паниковать даже за бетонной стеной.

Дальше поведение бота подчиняется следующей последовательности действий:

1) Проверяется условие $T_{final} < 0,7$

- если нет, то бот проверяет есть ли укрытие в радиусе 10 метров
- укрытие есть, бот увеличивает скорость в 2 раза
- если укрытий несколько, выбирается ближайшее
- укрытия нет, бот впадает в хаотичное бегство. Выбирается вектор, направленный прямо от дрона, к нему добавляется случайное ортогональное смещение и бот двигается в этом направлении 0,5 секунды, затем снова пересчитывает направление.
- если да, то проверяется условие $T_{final} < 0,3$:
- если да бот переходит в состояние «Внимание»: поворачивается лицом в сторону источника (чтобы смотреть на дрон), скорость уменьшается в 2 раза, но продолжает двигаться по своему маршруту патрулирования;
- иначе бот начинает активное уклонение:

a. запускается система пространственных запросов EQS, которая ищет лучшее укрытие в радиусе до 30 метров,

b. если укрытие обнаружено, то выбирается укрытие с наивысшим рейтингом, направление бота меняется на кратчайшее до этого укрытия,

c. скорость бота увеличивается в 1,5 раза.

d. если укрытия не обнаружено, то бот начинает двигаться, делая резкие повороты на 45–90 градусов перпендикулярно текущему направлению на дрон каждые 0,5 с.

Подобный алгоритм подразумевает, что любое текущее действие может быть немедленно прервано, если угроза резко возрастёт.

Заключение

Разработанный алгоритм и реализованный бот позволяет создавать вариативные, реалистичные и легко настраиваемые сценарии тренировки операторов БПЛА. Инструктор может быстро сгенерировать множество ситуаций, меняя только параметры, а не переписывая код. Разработанное решение уже интегрировано в симуляционную среду и может рассматриваться как практический инструмент повышения качества подготовки кадров в сфере беспилотной авиации.

Список литературы

1. Андриевский Б. Р., Попов А. М., Михайлов В. А., Попов Ф. А. Применение методов искусственного интеллекта для управления полетом беспилотных летательных аппаратов // Аэрокосмическая техника и технологии. 2023. №2. URL: https://cyberleninka.ru/article/n/primenenie-metodov-iskusstvennogo-intellekta-dlya-upravleniya-poletom-bespilotnyh-letatelnyh-apparatov (дата обращения: 02.04.2026).

2. Галкин Д.В., Петухов И.В., Танрывердиев И.О., Стешина Л.А., Стешин И.С., Курасов П.А. Разработка симулятора для обучения операторов беспилотных летательных аппаратов // Современные наукоемкие технологии, № 10, 2024. С. 27-31. URL: https://s.top-technologies.ru/pdf/2024/10/40167.pdf (дата обращения: 01.04.2026).

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЗАИМОДЕЙСТВИЯ С ИИ-АГЕНТАМИ С ПОДДЕРЖКОЙ МНОГОПОЛЬЗОВАТЕЛЬСКОГО АСИНХРОННОГО ДИАЛОГА

ГАРАНИН Р.А.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. Рассматривается проблема организации многопользовательского асинхронного взаимодействия с ИИ-агентами через единый Telegram-интерфейс. Предложена трёхкомпонентная архитектура: Telegram-бот (aiogram 3.x), backend управления сессиями (FastAPI, PostgreSQL, arq/Redis) и слой ИИ-агентов с трёхступенчатым AgentRouter. Экспериментально подтверждено 15,5-кратное ускорение обработки запросов по сравнению с синхронной схемой (93 RPS против 6 RPS) при нагрузке 20 параллельных пользователей. Все нефункциональные требования выполнены.

Ключевые слова: Telegram-бот, ИИ-агенты, асинхронная обработка, управление сессиями, маршрутизация запросов, FastAPI, arq/Redis.

1. Актуальность и постановка задачи

Быстрый рост аудитории мессенджеров и распространение больших языковых моделей (LLM) формируют спрос на диалоговые системы, способные обслуживать множество пользователей одновременно. По данным Statista, ежемесячная аудитория Telegram в 2024 году превысила 900 млн человек, а согласно отчёту McKinsey 2023 года, более 80% компаний планируют внедрить ИИ-ассистентов к 2026 году. Вместе с тем существующие open source-решения — Rasa, Botpress, Dialogflow ES, TeleChat — не обеспечивают одновременно: изоляцию диалогов нескольких пользователей, асинхронную обработку LLM-запросов и гибкую маршрутизацию между специализированными ИИ-агентами через единый Telegram-интерфейс.

Цель работы — обеспечение многопользовательского асинхронного взаимодействия пользователей с ИИ-агентами. Для её достижения последовательно решены пять задач: анализ аналогов; формулировка требований; разработка архитектуры; реализация прототипа; экспериментальная проверка.

2. Анализ аналогов и требования к системе

Оценивались пять решений по трём критериям (шкала 0–5): К1 — поддержка многопользовательских асинхронных диалогов, К2 — вариативность интеграции с ИИ-агентами, К3 — масштабируемость и отказоустойчивость. Результаты представлены в таблице 1.

Таблица 1

Сравнение аналогов

Аналог	К1	К2	К3	Итог	Тип
Microsoft BF + Azure	5	4	5	**14**	Облако
Google Dialogflow ES	4	3	4	**11**	NLU

Rasa Open Source	4	5	3	**12**	OpenSrc
Botpress OSS/cloud	4	4	4	**12**	Плагины
TeleChat (ChatGPT)	4	2	2	**8**	Простой

Ни одно решение не удовлетворяет всем трём критериям одновременно — это обосновывает собственную разработку.

На основе анализа сформулированы шесть требований: Р.1 — многопользовательский режим (≥20 одновременных пользователей, user_id-изоляция); Р.2 — сохранение контекста в ACID-совместимой СУБД; Р.3 — асинхронная обработка LLM-вызовов через очередь задач; Р.4 — автоматическая маршрутизация к одному из четырёх специализированных агентов; Р.5 — восстановление сессий после перезапуска компонентов; Р.6 — расширяемость без изменения бизнес-логики.

3. Архитектура и реализация

Система построена по трёхкомпонентной архитектуре. Транспортный уровень реализован на основе aiogram 3.x и работает в режиме polling или webhook, не выполняя тяжёлых вычислений: Telegram-бот извлекает user_id и chat_id и направляет сообщение в backend POST /messages. Backend управления сессиями построен на FastAPI 0.115, хранит состояние сессий в PostgreSQL 16 с ACID-гарантиями через SQLAlchemy 2.x + asyncpg, а тяжёлые задачи передаёт в асинхронную очередь arq/Redis 7. Слой ИИ-агентов содержит AgentRouter и четыре специализированных агента: Chat (свободный диалог), Programmer (код и алгоритмы), Economist (финансовые расчёты), Copywriter (тексты и контент).

Ключевым элементом является трёхступенчатый AgentRouter. Этап 1 — Keyword Scoring: лексический фильтр подсчитывает совпадения ключевых слов; при ≥2 совпадений агент выбирается без обращения к LLM. Этап 2 — LLM Classifier: при неоднозначности делается запрос к GPT-4o-mini (max_tokens=5), возвращающий метку агента. Этап 3 — Continuation Detector: если LLM вернул «chat», но запрос ≤8 слов, система сохраняет предыдущего агента для поддержания семантической связности. Развёртывание осуществляется через Docker Compose (пять сервисов).

4. Экспериментальная проверка

Проверка системы проводилась по пяти показателям: время ответа (среднее и p95), пропускная способность (RPS), корректность изоляции контекста, точность маршрутизации и устойчивость к сбоям. Асинхронная архитектура обеспечивает 93 RPS при 20 параллельных пользователях против 6 RPS при синхронной схеме — ускорение в 15,5 раза. Значение p95-задержки составило 42,3 мс при пороге 500 мс. Изоляция сессий — 100% тестовых сценариев. Точность маршрутизации — 12/12. Восстановление сессий после перезапуска — 100%. Все шесть требований выполнены.

Заключение

В работе решена задача организации многопользовательского асинхронного взаимодействия с ИИ-агентами через Telegram-интерфейс. Предложена и реализована трёхкомпонентная архитектура с трёхступенчатым AgentRouter. Экспериментально подтверждено 15,5-кратное ускорение обработки запросов по сравнению с синхронным подходом. Система обладает модульной структурой и допускает подключение новых агентов и транспортных адаптеров без изменения бизнес-логики.

Список литературы

1. Shanahan M. Talking about large language models // Communications of the ACM. 2024. Vol. 67, No. 2. P. 68–79.
2. Statista. Number of monthly active Telegram users worldwide from 2013 to 2024. URL: https://www.statista.com (дата обращения: 15.03.2025).
3. McKinsey & Company. The state of AI in 2023. URL: https://www.mckinsey.com (дата обращения: 15.03.2025).
4. Microsoft. Bot Framework Documentation. URL: https://docs.microsoft.com/en-us/azure/bot-service/ (дата обращения: 20.03.2025).
5. Microsoft. Azure Bot Service Overview. URL: https://learn.microsoft.com/en-us/azure/bot-service/ (дата обращения: 20.03.2025).
6. Google. Dialogflow ES Documentation. URL: https://cloud.google.com/dialogflow/es/docs (дата обращения: 20.03.2025).
7. Bocklisch T. et al. Rasa: Open source language understanding and dialogue management // arXiv:1712.05181. 2017.
8. Rasa Technologies. Rasa Open Source Documentation. URL: https://rasa.com/docs/rasa/ (дата обращения: 22.03.2025).
9. Botpress Inc. Botpress Documentation. URL: https://botpress.com/docs (дата обращения: 22.03.2025).
10. TeleChat. ChatGPT Telegram Bot. URL: https://github.com/chatgpt-telegram-bot/chatgpt-telegram-bot (дата обращения: 25.03.2025).

ML-МОДЕЛЬ МНОГОКЛАССОВОЙ КЛАССИФИКАЦИИ ДЛЯ ПРОДУКЦИОННОЙ ЭКСПЕРТНОЙ СИСТЕМЫ

Дудин Н.О.[1]

[1]СПБГЭТУ «ЛЭТИ» им. В. И. Ульянова (Ленина)

Аннотация. В статье рассматривается задача разработки ML-модели для продукционной экспертной системы по выбору программы магистратуры для абитуриентов. Предлагается подход, основанный на многоклассовой классификации с использованием алгоритма Random Forest. Описывается структура синтетического датасета, сгенерированного на основе логики продукционной экспертной системы Drools. Приводятся признаки модели (баллы, ответы на вопросы) и целевая переменная – одна из 17 программ. Обсуждается возможность интеграции модели в гибридную систему для формирования персонализированных рекомендаций.

Ключевые слова: паттерны, синтетический датасет, Random Forest, многоклассовая классификация, гибридная система, Drools.

Введение

Выбор магистерской программы представляет собой сложную задачу для абитуриента в силу большого количества направлений подготовки и отсутствия прозрачных критериев сопоставления. Традиционные подходы к консультированию часто предлагают усреднённые решения, не учитывающие уникальную комбинацию компетенций и предпочтений конкретного кандидата. Кроме того, статичность заложенных правил не позволяет системе адаптироваться к изменяющимся условиям приёма и появлению новых образовательных программ. Альтернативным подходом является применение методов машинного обучения, способных на основе исторических данных выявлять скрытые паттерны – устойчивые сочетания ответов и баллов, которые с высокой вероятностью приводят к выбору той или иной программы. Настоящая работа посвящена разработке ML-модели многоклассовой классификации, которая по ответам абитуриента и его

конкурсному баллу предсказывает наиболее подходящие направления обучения. В перспективе модель может быть интегрирована с существующей экспертной системой на базе Drools, образуя гибридную систему, сочетающую логическую прозрачность правил и прогностическую силу машинного обучения.

Разработка ML-модели

В рамках данного исследования ставится задача предсказания предпочтительной магистерской программы на основе данных, собираемых в ходе диалога с абитуриентом. Каждый абитуриент характеризуется вектором признаков, включающим конкурсный балл admission_score (целое число от 30 до 100), ответы на вопросы choice_1 … choice_8 (каждый ответ принимает значения 1 или 2, либо 0, если вопрос не был достигнут в цепочке), а также количество отвеченных вопросов choice_count. Целевой переменной является одна из 17 магистерских программ, обозначенных уникальным кодом, например CAD-01 (Системы автоматизированного проектирования микроэлектроники), SE-01 (Разработка распределенных программных систем) или DS-02 (Распределенные интеллектуальные системы и технологии). Таким образом, задача формулируется как многоклассовая классификация с 17 классами.

Для решения этой задачи предлагается использовать алгоритм Random Forest, который представляет собой ансамбль решающих деревьев, каждое из которых обучается на случайной подвыборке данных и случайном подмножестве признаков. Данный алгоритм обладает устойчивостью к переобучению, способностью работать с категориальными признаками без интенсивной предобработки и позволяет оценивать важность каждого признака для итогового предсказания. Итоговое предсказание модели формируется голосованием деревьев, а для многоклассовой классификации Random Forest выдаёт вектор вероятностей принадлежности образца к каждому из 17 классов.

Ввиду отсутствия реальных исторических данных на начальном этапе разработки был создан генератор синтетического датасета, моделирующий прохождение абитуриентом диалога, реализованного на базе продукционной системы Drools. Логика переходов между вопросами образует дерево решений, каждый лист которого соответствует одной из 17 программ. Параметры генератора включают размер датасета 5000 записей, распределение баллов с корректировкой под условную «сложность» программы, уровень шума 10% (невалидные пути, моделирующие нелогичный выбор абитуриента) и дополнение ответов нулями до фиксированной длины в 8 позиций. Распределение программ в датасете выполнено неравномерно с весами, что отражает реальную разницу в популярности направлений подготовки.

На основе вектора вероятностей, полученного от модели Random Forest, формируется персонализированная рекомендация из трёх программ с наибольшими значениями. Эти значения интерпретируются как процент совместимости профиля абитуриента с программой, а также как оценка шанса успешного поступления. Например, абитуриент с баллом 85 и цепочкой ответов [1, 2, 1] может получить рекомендацию:

1. SE-01 «Разработка распределенных программных систем» — 87%;
2. ACS-01 «Управление и информационные технологии в технических системах» — 76%;
3. CAD-01 «Системы автоматизированного проектирования микроэлектроники» — 54%.

Абитуриенту представляются три программы с указанием процента совместимости/шанса, что позволяет ему осознанно сравнить варианты.

Заключение

В работе предложен подход к разработке ML-модели для поддержки выбора магистерской программы на основе алгоритма Random Forest. Сгенерирован синтетический датасет, имитирующий логику диалога абитуриента с экспертной системой Drools. Модель решает задачу многоклассовой классификации (17 классов) и выдаёт топ-3 программы с вероятностной оценкой, которая интерпретируется как процент совместимости и шанс поступления. Дальнейшие направления развития включают обучение модели на реальных данных о поступлениях, интеграцию модели в существующую продукционную систему Drools через REST API для формирования гибридных рекомендаций, а также применение интерпретируемых методов (например, SHAP) для объяснения причин выбора той или иной программы.

Список литературы

1. Breiman L. Random Forests. Machine Learning, 2001, vol. 45, no. 1, pp. 5–32.
2. Никольский И.М. Оптимизация размера классификатора при сегментации трехмерных точечных образов древесной растительности. Компьютерные исследования и моделирование, 2025, т. 17, № 4, с. 665–675.
3. Hornung R., Hapfelmeier A. Multi forests: Variable importance for multi-class outcomes. arXiv preprint, 2024, arXiv:2409.08925.
4. Конопопець М., Туровський О., Аксамітний О., Матійко А. Порівняльний аналіз моделей машинного навчання Random Forest та XGBoost у задачі класифікації інцидентів безпеки. Herald of Khmelnytskyi National University. Technical sciences, 2025, № 357, с. 29.
5. Документация по платформе Drools: введение в технологию Drools. Электронный ресурс. Red Hat, 2026. Режим доступа: https://docs.drools.org/8.44.0.Final/drools-docs/drools/introduction/index.html. Дата обращения: 04.05.2026.

РАЗРАБОТКА МИКРОСЕРВИСНОЙ СИСТЕМЫ ИДЕНТИФИКАЦИИ ЛЮДЕЙ В ВИДЕОПОТОКЕ С РАЗДЕЛЕНИЕМ РЕЛЯЦИОННОГО И ВЕКТОРНОГО ХРАНЕНИЯ ДАННЫХ

В.С. Ипатов[1]

[1]*Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»*

Аннотация. В работе представлена микросервисная система идентификации людей в видеопотоке, объединяющая детекцию, трекинг и биометрический поиск по признакам лица и походки. Предложена архитектура с разделением реляционного и векторного хранения: PostgreSQL используется для событий и метаданных, Milvus — для векторных представлений и поиска ближайших соседей. Реализован асинхронный обмен между компонентами через RabbitMQ, обеспечивающий масштабируемость и отказоустойчивость. Показано, что выбранная архитектура сокращает время операторского анализа архивов и повышает оперативность поиска появлений человека по множеству камер.

Ключевые слова: видеонаблюдение, идентификация личности, компьютерное зрение, векторный поиск, микросервисная архитектура, Milvus, PostgreSQL.

1. Введение

Современные распределенные системы видеонаблюдения формируют непрерывный поток видеоданных, анализ которого в ручном режиме становится практически невозможным при увеличении количества камер. В реальных сценариях эксплуатации

оператору необходимо не только фиксировать текущие события, но и быстро находить похожие эпизоды в архиве, восстанавливать маршрут перемещений человека и проверять результаты автоматического распознавания.

Классические архитектуры, где все данные и поисковые операции сосредоточены в одной реляционной базе, плохо масштабируются при росте объема биометрических признаков. Поиск по многомерным векторам требует специализированных индексов и алгоритмов ближайших соседей, что делает целесообразным использование отдельного векторного хранилища [4][5].

Целью работы является разработка микросервисной системы идентификации людей в видеопотоке с разделением реляционного и векторного хранения данных. Для достижения цели решаются задачи организации конвейера обработки видеопотока, хранения событийной информации, извлечения биометрических признаков и интеграции поисковых сервисов в единый прикладной контур.

2. Архитектура и принципы построения системы

Архитектура построена по микросервисному принципу и включает следующие компоненты: сервис получения и агрегации видеопотоков, алгоритмический модуль детекции и трекинга, очередь сообщений, серверную часть с REST/WebSocket-интерфейсами, реляционную базу данных PostgreSQL, векторную базу Milvus и веб-клиент оператора [5][6][7]. Такой подход позволяет масштабировать вычислительные и серверные компоненты независимо и упрощает сопровождение системы.

Обработка видеопотока начинается с детекции людей на кадрах и ассоциации обнаружений в треки [1][3]. Для каждого трека фиксируются временные характеристики, координаты ограничивающих рамок и дополнительные метаданные. После завершения трека выполняется отбор ключевых кадров, на основе которых рассчитываются векторы лица и походки. Полученные признаки используются для поиска совпадений и дальнейшей идентификации.

Обмен результатами между вычислительным модулем и серверной частью организован асинхронно через RabbitMQ [7]. Очередь сообщений компенсирует пики нагрузки и предотвращает потерю данных при временной недоступности одного из сервисов. Такой механизм особенно важен в многокамерных системах, где нагрузка на вычислительные узлы и API может меняться неравномерно.

3. Разделение реляционного и векторного хранения

Реляционный контур хранения предназначен для событийной модели и бизнес-данных [6]. В PostgreSQL сохраняются записи о событиях наблюдения (PersonLog), идентификаторы камер, интервалы времени, траектории движения, ссылки на медиаматериалы и служебные статусы обработки. Также в реляционной БД находятся пользовательские сущности, роли и правила доступа к функциональности системы.

Векторный контур реализован на базе Milvus и используется для хранения биометрических признаков лица и походки, а также для выполнения операций поиска ближайших соседей [5]. В системе применяются отдельные коллекции для текущих (live) и эталонных (known) векторов, что позволяет разделить задачи оперативного поиска и накопления подтвержденных эталонов. Для ускорения поиска используются специализированные индексы и метрики расстояния в многомерном пространстве признаков [5].

Связь между двумя контурами организована через идентификаторы векторов, которые сохраняются в реляционной записи события. При пользовательском запросе серверная часть выполняет векторный поиск в Milvus, затем объединяет результаты с карточками событий из PostgreSQL и возвращает оператору агрегированное представление: событие, список совпадений, источники и временной контекст.

4. Практическая реализация и оценка применимости

Реализованная система поддерживает сценарии контроля доступа и постфактум-аналитики: поиск всех появлений человека по образцу, проверку пересечений маршрутов, фильтрацию событий по времени и камерам. Наличие двух специализированных хранилищ дает возможность одновременно поддерживать быстрый векторный поиск и гибкую работу со структурированными запросами по метаданным.

С практической точки зрения предложенный подход снижает время реакции оператора при расследовании инцидентов, так как основной объем ручного поиска по архиву заменяется автоматизированной выдачей релевантных совпадений. Микросервисная архитектура также упрощает масштабирование: при росте числа камер можно отдельно увеличивать вычислительные ресурсы алгоритмического контура и серверного API.

Дополнительным преимуществом является технологическая независимость компонентов: изменения в моделях детекции и векторизации не требуют переработки реляционной схемы, а оптимизация параметров индексов Milvus выполняется без изменения бизнес-логики клиентских приложений. Это повышает устойчивость решения к эволюции алгоритмов и требований заказчика.

5. Заключение

В работе разработана микросервисная система идентификации людей в видеопотоке, основанная на разделении реляционного и векторного хранения данных. Показано, что такое разделение позволяет эффективно сочетать надежное хранение событийной информации с высокоскоростным биометрическим поиском. Реализованный подход ориентирован на промышленное применение и может быть использован в масштабируемых системах интеллектуального видеонаблюдения.

Дальнейшее развитие предполагает проведение расширенной экспериментальной оценки точности идентификации в разных условиях съемки, исследование влияния параметров векторных индексов на латентность поиска и интеграцию дополнительных модальностей признаков для повышения устойчивости распознавания.

Список литературы

1. Redmon J., Farhadi A. YOLOv3: An Incremental Improvement [Электронный ресурс]. URL: https://arxiv.org/abs/1804.02767 (дата обращения: 30.04.2026).
2. Wojke N., Bewley A., Paulus D. Simple Online and Realtime Tracking with a Deep Association Metric [Электронный ресурс] // 2017 IEEE International Conference on Image Processing (ICIP). URL: https://ieeexplore.ieee.org/document/8296962 (дата обращения: 30.04.2026).
3. Ge Z., Liu S., Wang F., Li Z., Sun J. YOLOX: Exceeding YOLO Series in 2021 [Электронный ресурс]. URL: https://arxiv.org/abs/2107.08430 (дата обращения: 30.04.2026).
4. InsightFace: 2D and 3D Face Analysis Project [Электронный ресурс]. URL: https://github.com/deepinsight/insightface (дата обращения: 30.04.2026).
5. Milvus Documentation [Электронный ресурс]. URL: https://milvus.io/docs (дата обращения: 30.04.2026).
6. PostgreSQL Documentation [Электронный ресурс]. URL: https://www.postgresql.org/docs/ (дата обращения: 30.04.2026).
7. RabbitMQ Documentation [Электронный ресурс]. URL: https://www.rabbitmq.com/docs (дата обращения: 30.04.2026).

АРХИТЕКТУРА ИНФОРМАЦИОННОЙ СИСТЕМЫ НА БАЗЕ DROOLS

Литвинов Е.Н.[1]

[1]*СПБГЭТУ «ЛЭТИ» им. В. И. Ульянова (Ленина)*

Аннотация. В статье рассматривается архитектура распределённой информационной системы с встроенной экспертной системой на основе правил, предназначенной для абитуриентов магистратуры. Проведён сравнительный анализ движков CLIPS, JESS и Drools, обоснован выбор последнего как наиболее современного решения. Описана интеграция с KIE-Server, REST API, PostgreSQL и Docker. Сделан вывод о пригодности предложенной архитектуры для построения масштабируемых систем поддержки принятия решений.

Ключевые слова: Экспертные системы, движок правил, Drools, KIE-Server, распределенная информационная система, архитектура программного обеспечения, правила.

Введение

В современном мире большое внимание уделяется проблемам технологии искусственного интеллекта, которые прочно вошел во многие сферы жизни. В области искусственного интеллекта на данный момент есть несколько направлений развития, одно из которых – экспертные системы. Экспертные системы можно реализовать несколькими способами: большие языковые модели на основе нейронных сетей или экспертная система на основе правил. На сегодняшний день экспертные системы на основе правил используются как вспомогательные программы для создания больших информационных систем. Однако возможно использование экспертных систем как часть большой распределенной системы. Далее рассмотрим архитектуру распределенной информационной системы с включенной в нее экспертной системы на основе правил. Данная система должна помогать абитуриентам магистратуры с выбором программы обучения на основании конкурсных баллов и ответов на вопросы системы.

Архитектура и инфраструктура

В данном случае ядро нашей системы – экспертная система на основе правил. Это означает, что в первую очередь требуется сделать выбор движка правил, который будет находить ответ по входным данным с помощью правил в базе знаний. Есть несколько возможных вариантов движков, предлагается рассмотреть каждый отдельно:

1. CLIPS - инструмент для создания экспертных систем, изначально разработанный Отделом программных технологий (STB) Космического центра НАСА имени Линдона Б. Джонсона. С момента первого выпуска в 1986 году CLIPS постоянно дорабатывался и совершенствовался. CLIPS предназначен для упрощения разработки программного обеспечения, моделирующего человеческие знания или экспертный опыт [1]. На данный момент CLIPS является устаревшим и фактически не используется.
2. JESS (Java Expert System Shell) - движок правил и среда сценариев, написанные полностью на языке Java. Он предназначен для разработки экспертных систем, основанных на правилах, и может быть тесно интегрирован с Java-кодом. JESS был разработан Эрнестом Фридманом-Хиллом. Изначально JESS создавался как клон системы CLIPS, но со временем превратился в самостоятельную среду с собственными уникальными возможностями [2]. На сегодняшний день, разработка на JESS затруднена, так как прекращена поддержка данного инструмента.
3. Drools – движок бизнес-правил, предназначенный прежде всего для разработки программных продуктов с изменяемыми требованиями. Написан на языке Java и

является одним из модулей Java [3][4]. Данный инструмент можно использовать и для встраивания экспертной системы как часть распределенной информационной системы. На сегодняшний день это лучшее решение для задачи, описанной выше.

Для удобного встраивания экспертной системы у разработчика Drools – JBoss – также присутствуют другие продукты, которые можно использовать при разработке распределенной информационной системы [4].

Во-первых, KIE-Server – сервер для обработки правил и хранения базы знаний. Данное решение помогает выделить в архитектуре отдельный от основной логики компонент, связанный с обработкой правил. Данный подход может быть полезен при разработке крупной распределенной системы с несколькими параллельно работающими серверами. Для динамического управления правилами предусмотрена упаковка правил в специализированный файл KJAR, который собирается на KIE-Server после какого-либо изменений правил.

Во-вторых, Business Central – система управления разработкой от JBoss, которая может быть использована для управления и администрирования правил и знаний в системе, к которой подключен Business Central. Данный подход не удобен, если экспертная система является функциональным модулем информационной системы, поскольку функционал предоставляемый Business Central является вспомогательным в разработке ПО. В данном инструменте также есть функциональность системы контроля версий. Business Central предназначен для полного контроля проектом.

Далее стоит отметить, что вариант с подключением Business Central не подходит для решения задачи администрирования и управления правилами пользователями из-за сложного интерфейса и полного доступа к исходным кодам разрабатываемой системы.

Что касается интеграций сервисов системы, в том числе обработчик правил и основная логика приложения, JBoss предоставляет возможность взаимодействия при помощи REST API, которое является удобной современной технологией для передачи данных между сервисами.

Хранение данных должно происходить в базе данных. В качестве системы управления базой данных используем PostgreSQL, из ее преимуществ можно отметить удобный синтаксис и большое сообщество пользователей. Также данная СУБД является импортозамещённой. Для удобного развёртывания системы можно использовать контейнеры Docker, в которых будет KIE-Server и база данных.

Таким образом присутствуют все необходимые элементы для построения целостной архитектуры распределенной информационной системы с встроенной в нее экспертной системы. Рассмотрим архитектуру, представленную на диаграмме (Рис. 1).

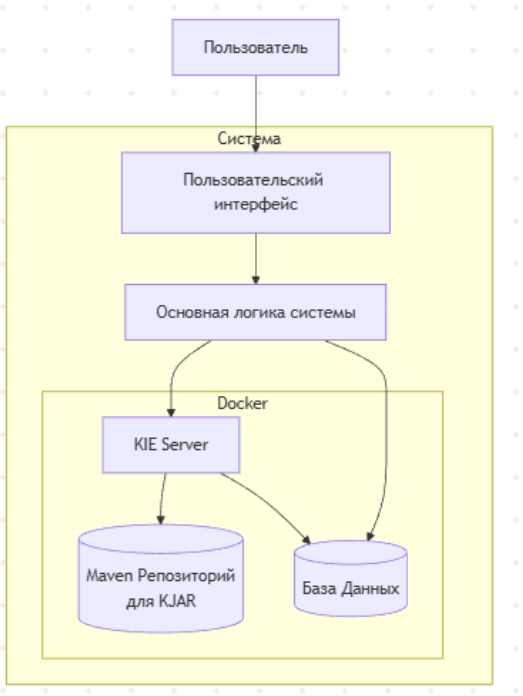


Рис. 1. Архитектура системы

На диаграмме представлены все взаимосвязи сервисов, которые обеспечиваются с помощью API.

На диаграмме нет сервиса Business Central, так как данный инструмент не подходит по функционалу для задачи, описанной выше.

Заключение

Данная архитектура подходит для распределенных информационных систем содержащих экспертную систему с движком правил Drools, так как здесь используется технологии и инструменты, разработанные компанией, создавшей Drools – JBoss. Они полностью совместимы и удовлетворяют все потребности разработчиков информационных систем.

Список литературы

1. Giarratano J.C. CLIPS User's guide.
2. Ernest J. Friedman-Hill. Jess, The Rule Engine for the Java Platform.
3. Narendra Kumar, Dipti D Patil, Dr. Vijay M.Wadha. Rule based programming with Drools.
4. Документация по платформе Drools: введение в технологию Drools. Электронный ресурс. Red Hat, 2026. Режим доступа: https://docs.drools.org/8.44.0.Final/drools-docs/drools/introduction/index.html. Дата обращения: 04.04.2026.

УЧЕБНАЯ АНАЛИТИКА КАК ИНСТРУМЕНТ РАННЕЙ ДИАГНОСТИКИ АКАДЕМИЧЕСКОЙ НЕУСПЕШНОСТИ

РАЗЖИВИНА Н.М.

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)

Аннотация. Запоздалое обнаружение академической неуспеваемости приводит к потери момента, когда педагогические и управленческие меры были бы эффективными. В статье рассматривается возможность раннего выявления риска отчисления как задача предиктивной аналитики с управляемой интерпретацией результата. Представлена подсистема предупреждения, в которой риск оценивается как прогноз неблагоприятного академического исхода на основе данных о посещаемости, успеваемости, наличия академических задолженностей и цифрового следа активности в LMS. Описаны принципы расчета риск-скоринга, механизм факторной декомпозиции и перевод сигнала риска в список приоритетных кейсов для сопровождения.

Ключевые слова: учебная аналитика, риск отчисления, раннее предупреждение, цифровой след, интерпретируемая предиктивная модель.

Введение

В условиях цифровизации управление учебным процессом опирается на данные и предиктивную аналитику [1], но академический риск нередко выявляется постфактум, после накопления задолженностей, когда восстановить траекторию сложнее. В работе описана подсистема раннего выявления риска отчисления в ИС поддержки учебного процесса: она формирует риск-профиль студента и приоритетный перечень кейсов для кураторов и заместителя декана. Цель статьи – обосновать методику раннего скоринга и показать, как измеримые факторы, их динамический пересчёт и интерпретируемость переводят разрозненные данные в управленческие решения.

В международной практике системы раннего предупреждения академического риска развиваются более десяти лет: от институциональных решений (например, Purdue Signals) [2] до открытых инициатив уровня OAAI [3] и теоретико-методологических рамок learning analytics [4, 5]. В этих исследованиях показана эффективность раннего выявления риска по цифровому следу, однако в ряде реализаций акцент делается либо на LMS-активности, либо на итоговой классификации без явной управленческой декомпозиции сигнала. Предлагаемый в статье подход ориентирован на практику вуза и сочетает совместный анализ административных данных и LMS-метрик, интерпретируемый риск-скоринг с факторной расшифровкой для куратора, а также динамический недельный пересчёт риска с механизмом ограничения доминирования, снижающим влияние кратковременных всплесков.

Исследовательская постановка задачи раннего реагирования

Риск отчисления в системе раннего предупреждения определяется как вероятность неблагоприятного академического исхода (отчисление или устойчивые задолженности) по текущей динамике учебных и поведенческих признаков; такой подход соответствует практикам предиктивной аналитики в вузах [6]. Гипотеза состоит в том, что цифровой образовательный след и его изменения в течение семестра позволяют выявлять неблагоприятную траекторию до наступления критического статуса.

Для проверки сопоставлены недельные траектории среднего риска в группах «впоследствии отчисленные» и «не отчисленные»: с 9-10-й недели наблюдается устойчивое расхождение (~45–50% против ~15–21%; рис. 1). В той же схеме темпоральной валидации ROC-анализ показывает рост AUC-ROC от ~0,63 на 4-й неделе до ~0,83 к 10-й (рис. 2), что подтверждает формирование устойчивой различимости классов уже в первые месяцы семестра.

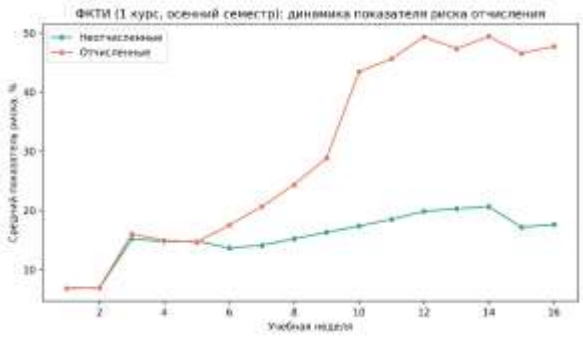


Рис. 1. Динамика показателя риска отчисления

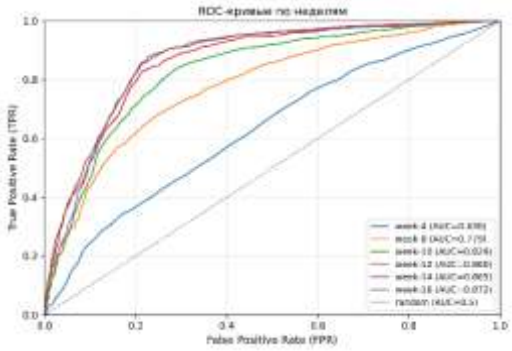


Рис. 2. ROC-кривые

Выявление риска: что и как считается

Риск в системе пересчитывается на протяжении семестра, поскольку значимость факторов и состояние студента изменяются во времени. Для студента i на шаге t формируется риск-скоринг $R_i(t)$ в диапазоне от 0 до 100, а также категория риска (низкий/средний/высокий) и факторная расшифровка.

Для расчета используются четыре класса факторов, каждый из которых связан с наблюдаемыми и измеримыми метриками учебного процесса:

1. Посещаемость: $f_{att}(i,t) = clip(1 - A_i(t) / \tau_{att}(g,t), 0, 1)$, где $A_i(t)$ – текущая доля посещенных занятий, $\tau_{att}(g,t)$ – норматив для соответствующей когорты g (факультет/программа/курс).

2. Успеваемость: $f_{grade}(i,t) = clip((z_{crit} - z_i) / z_{span}, 0, 1)$; $z_i = (G_{prev\_i} - \mu_g) / \sigma_g$, где G_{prev_i} – средний балл студента по итогам последней сессии, μ_g и σ_g – среднее и стандартное отклонение по когорте g, z_{crit} – критический уровень, z_{span} – ширина переходной зоны.

3. Академические задолженности: $f_{debt}(i,t) = 1 - \exp(-\alpha \cdot D_i(t) - \beta \cdot O_i(t))$, где $D_i(t)$ – число задолженностей за предыдущий семестр, $O_i(t)$ – число задолженностей за позапрошлый семестр, α и β – веса задолженностей (задаются на основе исторических данных, как правило, $\beta > \alpha$) .

4. Онлайн-активность: $f_{lms}(i,t) = clip(1 - LMS_i(t) / \tau_{lms}(g,t), 0, 1)$, где $LMS_i(t)$ – агрегированный индекс цифрового следа (просмотр видео, изучение материалов, выполнение заданий/тестов, загрузка отчетов), $\tau_{lms}(g,t)$ – калибруемый норматив.

Здесь clip(x,0,1) – ограничение значения интервалом [0,1].

Адаптация $\tau_{att}(g,t)$ и $\tau_{lms}(g,t)$ выполняется по когорто-недельным распределениям ретроспективных данных: для каждой когорты g на неделе t рассчитываются устойчивые базовые уровни (медиана и межквартильный размах), после чего норматив задается как $\tau(g,t) = Q50_{g,t} - \lambda \cdot IQR_{g,t}$ с ограничением допустимого диапазона по экспертным правилам факультета/программы. Параметр λ задает степень смещения норматива ниже медианы когорты и выбирается по ретроспективной калибровке как компромисс между чувствительностью раннего выявления и допустимой долей ложных срабатываний. Далее нормативы сглаживаются по неделям (экспоненциальное сглаживание), чтобы исключить влияние разовых выбросов.

При расчете риска учитывается неоднородность условий. Текущие показатели интерпретируются относительно персональной предыстории студента: оценивается, насколько его показатели отклоняются от уровней предыдущих семестров. Дополнительно учитывается индекс лояльности преподавателя, рассчитанный по историческим данным (доля студентов, не закрывших дисциплину). Данный контекст используется как

корректирующий фактор, но не отменяет базовую логику модели: при устойчивом росте задолженностей вклад риск-компонент усиливается, отражая ухудшение академической траектории.

Итоговый риск для студента i в момент t задается аддитивной композицией:

$$R_i(t) = 100 \cdot \sum w_k \cdot f_k(i,t), \sum w_k = 1.$$

Здесь w_k – веса вкладов факторов, которые определяются двухэтапно: сначала задаются базовые экспертные веса $w_{k,base}$ (стартовые приоритеты), затем для каждой недели по историческим данным оценивается информативность факторов и формируются $w_{k,data}(t)$. После этого выполняется обновление $w_{k,new}(t) = (1 - \theta) \cdot w_{k,base} + \theta \cdot w_{k,data}(t)$, где $\theta \in [0,1]$ задает баланс доверия данным относительно экспертной инициализации и ограничение скорости изменения относительно предыдущей недели $w_{k,(t-1)}$ с порогом Δ, чтобы избежать резких скачков. На завершающем шаге веса повторно нормируются ($\sum w_k(t)=1$). Параметры θ и Δ подбираются по ретроспективной валидации по критерию лучшего качества прогноза при устойчивом недельном профиле весов, после чего фиксируются на семестр и пересматриваются по итогам нового цикла данных. На рис. 3 показана эволюция $w_k(t)$: на ранних неделях сильнее LMS и успеваемость прошлых периодов, на поздних – задолженности, что согласуется с накоплением административной информации.

Рис. 3. Динамика весов по неделям

Чтобы кратковременный всплеск одного признака не давал ложной тревоги, компоненты сглаживаются по времени, превышение порога учитывается только при длительности не менее двух недель подряд; при отсутствии подтверждения другими факторами вклад компоненты ограничивается. Пороги групп риска (низкий/средний/высокий) задаются по факультету, программе и курсу и по ретроспективным данным когорты так, чтобы при ограничениях на precision и долю студентов в зоне риска (нагрузка на куратора) максимизировать recall. Таким образом обеспечивается баланс между полнотой раннего выявления и управляемостью сопровождения.

По результатам темпоральной валидации система демонстрирует формирование раннего сигнала риска уже к 10-й неделе (ROC-AUC ~0.83), а к 12-й неделе достигает устойчивого качества разделения (ROC-AUC ~0.86). Дальнейшая донастройка модели предполагает повышение precision и снижение ложных включений при сохранении recall; в пилоте принят компромисс: расширенный охват при ручной верификации куратором.

Таблица 1

Динамика метрик качества прогноза по неделям семестра

Неделя	Precision	Recall	F1	ROC-AUC
4	0.299	0.120	0.171	0.630
8	0.424	0.496	0.457	0.779
10	0.456	0.636	0.509	0.829
12	0.458	0.638	0.532	0.860
16	0.454	0.726	0.559	0.872

На рис. 4 показан кумулятивный охват отчислений при ранжировании студентов по риск-скорингу: для выбранной калибровке система к 10–12 неделе при выборе примерно топ-20% студентов дает более 60% охвата отчисленных, что подтверждает практическую возможность раннего предупреждения.

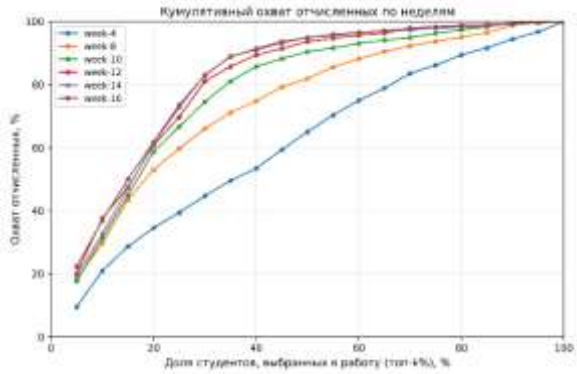


Рис. 4. Кумулятивный охват

По сравнению с используемыми в литературе бенчмарками раннего предупреждения (простые правила по одной задолженности/посещаемости и базовая логистическая регрессия на тех же признаках), предложенный риск-скоринг показывает более устойчивое качество ранжирования на средних неделях семестра. Для 8-12 неделей наблюдается ROC-AUC 0,78–0,85 при сохранении управляемой нагрузки на сопровождение, что сопоставимо с опубликованными результатами для EWS в e-learning и институциональных системах [2-3,7], но при этом дополняется факторной интерпретацией сигнала и недельным пересчётом риска.

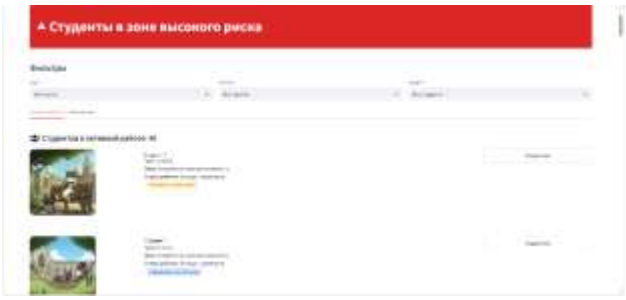


Рис.5. Страница риск-кейсов

Сигналы риска используются именно для адресного сопровождения и ручной верификации, а не для автоматических санкций. Интерпретируемость обеспечивается факторной декомпозицией. Пользователь видит, какие компоненты f_k внесли наибольший вклад и описанную причину: например, в приоритизированном списке риск-кейсов (рис. 5) пользователь видит указание основного фактора и маркер повторного попадания в группу

риска. Также указан текущий статус работы, как правило, если у студента есть академические долги, то с ним работает заместитель декана, в случае низкой активности студента сопровождает куратор.

Качество риск-оценки зависит от полноты данных посещаемости и цифрового следа (на данный момент они частично неполны). Соответственно, полученные метрики следует трактовать как реалистичную пилотную оценку, требующую настройки весов и формализации регламента заполнения данных об учебном процессе.

Реализация подсистемы раннего выявления факторов риска

Технически подсистема реализована на Python/Streamlit как модуль комплекса учебной аналитики. Сквозной процесс данных задаётся цепочкой: данные → ETL и нормализация → расчёт метрик → риск-компоненты → агрегированный риск-скоринг → ролевые дашборды. Вычислительный слой построен на pandas/numpy: это обеспечивает воспроизводимую обработку табличных образовательных данных (временные окна, нормирование, векторизованные агрегаты) и удобное уточнение правил без смены стека. Персистентное хранение – PostgreSQL; доступ к витринам – SQLAlchemy/psycopg2, что поддерживает устойчивые запросы и переносимость при росте нагрузки.

Наиболее трудоёмкий этап – объединение разнородных источников (журналы посещаемости, ведомости успеваемости, данные LMS) с нормализацией идентификаторов студента, группы и дисциплины. Пропуски в данных обрабатываются по типам. Если в недельном ряду временно отсутствует значение, используется последнее доступное значение (в пределах заданного окна наблюдения). Для агрегированных показателей по когорте пропуски заполняются медианой соответствующей группы и недели. Если LMS-данные структурно отсутствуют (например, по дисциплине не ведется активность в LMS), такой признак помечается как ненаблюдаемый, а веса пересчитываются между доступными компонентами. Для каждого студента также рассчитывается индикатор полноты данных; при высокой доле пропусков риск-оценка помечается как требующая ручной проверки куратором.

После формирования витрин рассчитываются риск-компоненты и итоговый скоринг, который в Streamlit выводится в ролевых интерфейсах: куратор видит закреплённые группы и персональные кейсы, заместитель декана – агрегированную аналитику по курсам/группам/дисциплинам. Доступ к профилям ограничен ролями, выгрузки для анализа обезличиваются, действия пользователей логируются.

Заключение

Подсистема раннего выявления риска отчисления в составе ИС учебного процесса рассматривается как предиктивный инструмент, позволяющий перейти от фиксации последствий к превентивному сопровождению студентов. На основе проведенного анализа показано, что динамика факторов цифровой образовательной истории содержит ранние признаки академической деградации и применима для формирования интерпретируемого риск-профиля до накопления задолженностей. Эти выводы согласуются с современными исследованиями в области учебной аналитики, подтверждающими возможность раннего предупреждения академического риска. Внедрение подсистемы повышает обоснованность и своевременность решений куратора и заместителя декана, способствует снижению доли запоздалых вмешательств и повышению сохранности контингента. В дальнейшем планируется расширение модели с учетом факторов психологического состояния, текущей

успеваемости и внеучебного контекста для формирования более полной и точной картины академического риска.

Список литературы

1. Ведомственная программа цифровой трансформации Министерства науки и высшего образования Российской Федерации на 2022–2024 годы (утв. Минобрнауки России). [Электронный ресурс]. URL: https://legalacts.ru/doc/vedomstvennaja-programma-tsifrovoi-transformatsii-ministerstva-nauki-i-vysshego-obrazovanija_1.
2. Arnold K.E., Pistilli M.D. Course Signals at Purdue: Using Learning Analytics to Increase Student Success // Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (LAK '12). 2012. P. 267–270. DOI: 10.1145/2330601.2330666.
3. Open Academic Analytics Initiative (OAAI). Predictive Analytics Reporting (PAR) Framework. [Электронный ресурс]. URL: https://www.openacademicanalytics.org/.
4. Siemens G. Learning Analytics: The Emergence of a Discipline // American Behavioral Scientist. 2013. Vol. 57. No. 10. P. 1380–1400. DOI: 10.1177/0002764213498851.
5. Pardo A., Siemens G. Ethical and Privacy Principles for Learning Analytics // British Journal of Educational Technology. 2014. Vol. 45. No. 3. P. 438–450. DOI: 10.1111/bjet.12152.
6. Аликина Е.В., Мальцев Д.В. Управление образовательным процессом университета на основе прогнозирования успеваемости обучающихся // Высшее образование в России. 2024. Т. 33. № 11.
7. Boudjehem R., Lafifi Y. An early warning system to predict dropouts inside e-learning environments // Education and Information Technologies. 2024. Vol. 29. P. 16365–16385. DOI: 10.1007/s10639-024-12498-1.

Сборник материалов

XIV Всероссийской Научно-практической конференции
с международным участием

«НАУКА НАСТОЯЩЕГО И БУДУЩЕГО»

для студентов, аспирантов и молодых ученых

состоявшейся 16-18 мая 2026 г. в г.Санкт-Петербурге

Том IV