
**ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ,
РОБОТОТЕХНИЧЕСКИЕ СИСТЕМЫ,
МАШИННОЕ ОБУЧЕНИЕ,
КОМПЬЮТЕРНАЯ ЛИНГВИСТИКА И АЛГОРИТМЫ**

DEVELOPMENT OF METHODS OF INTELLECTUAL ANALYSIS OF AUDIO DATA

M. Y. Likhachev, K. A. Likhacheva

Voronezh State University

Annotation. This paper focuses on the ANN approach for Note recognition by frequency and amplitude involves extracting relevant acoustic features. Precedent data is used to design a solution for recognizing melody using a neural network model with custom architecture and metrics assess the quality of the result.

Keywords: ANN, melody, note, recognition, machine learning.

Introduction

Music is an important part of any person's life, one of the ancient art forms. Nowadays it continues to develop, including with the help of modern technologies. The Internet hosts a large number of online applications and platforms designed to work with digitized sound information (music). This group includes such types of systems as music players (iTunes, YouTube Music, Yandex.Music, Spotify), music recognition and search systems (Shazam), as well as platforms for commercial distribution of user authored music (SoundCloud, Bandcamp) [1].

One of the well-known research problems in this area is the melody recognition problem. Some of the most well-known and widespread machine learning systems that solve such problems related to audio analysis (virtual assistants Alexa, Google Home, Siri) are systems based on data processing models that extract various features from audio signals. Depending on the data extraction methods (optical or auditory), there are different solutions with their own characteristic approaches (Optical character recognition (OCR), Optical music recognition (OMR), Note recognition, and Speech recognition). This paper will focus on the study of ANN approach for Note recognition task as the most common one in such tasks. [2]

The purpose of the article is to present the results of the development of an ANN model for the melody recognition solution. [1, 2].

1. Problem statement

Audio analysis, a field that includes automatic speech recognition (ASR), digital signal processing, and music classification, tagging, and generation, is an emerging subdomain of deep learning applications. Sound is represented in the form of an audio signal with parameters such as frequency, bandwidth, decibel, etc. A typical audio signal can be expressed as a function of amplitude and time. Some devices can capture these sounds and represent them in a machine-readable format. The audio processing process involves extracting acoustic features relevant to the task at hand, followed by decision making schemes that include detection, classification and knowledge fusion. There are various characteristics by which an audio signal can be evaluated.

Spectrogram is a visual way of representing the level or loudness of a signal over time at different frequencies present in a waveform. The data is converted into a short-time Fourier transform. It can be used to determine the amplitude of the different frequencies being played at a given time of an audio signal. Spectral (frequency) features are obtained by transforming the time signal into the frequency domain using the Fourier transform. These include fundamental tone frequency, frequency components, spectral centroid, spectral flux, spectral density, spectral rolloff, etc. [3]

Spectral Centroid (Spectral Centroid) indicates at which frequency the energy of the spectrum is concentrated, or in other words, indicates where the "center of mass" for sound is located.

Chromaticity (feature or color vector, Chromagram) (Chroma) is usually represented by a feature vector of 12 elements that indicates the amount of energy of each pitch class (Pitch class) C, C#, D, D#, E, F, F#, G, G#, A, A#, B in the signal. It is used to describe a measure of similarity between musical pieces.

This problem belongs to the class of identification problems. A problem belonging to the class of time series analysis was also used as a subproblem to be solved. A finite set of precedents is given, for each of which some data are collected. The data about a precedent is also its description. The aggregate of all available precedent descriptions is a training sample. It is required to identify general dependencies, patterns, interrelationships inherent not only in this particular sample, but in general all precedents, including those that have not yet been observed, and on the basis of the identified dependencies to design the architecture of the solution to the problem of recognizing the melody. [4]

2. Development of neural network model

For the design and realization of the neural network model for solving the problem 4 indicators (audio features) were recorded — spectral rolloff, spectral centroid, mel-frequency characteristics, and chromagram.

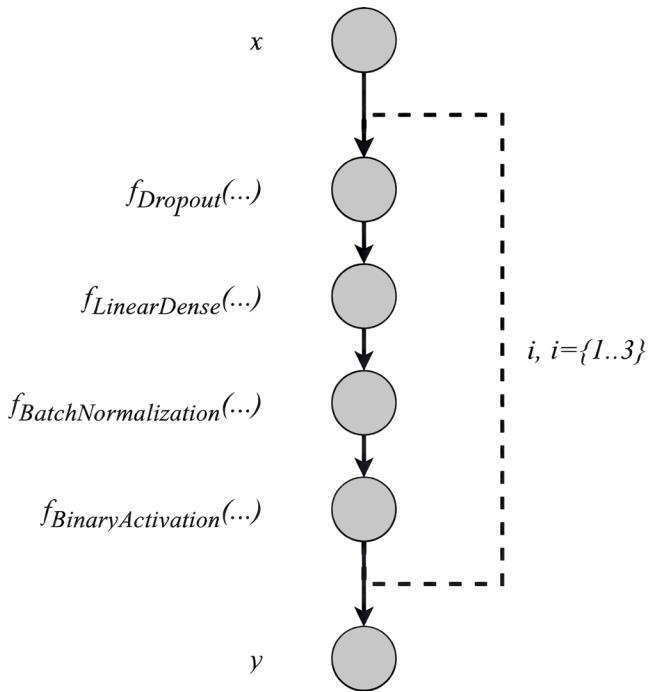


Fig. 1. Architecture of ANN model

To build an ANN model for processing the indicators, an architecture was designed based on a MLP ANN model. The choice of this model as the core of the current solution was made within the framework of the idea of dividing the problem to be solved into 2 regression tasks: the task of predicting frequency heights and the task of predicting durations and intervals. The peculiarity of the model in the considered solution is the recognition of both values (duration and pitch) in parallel with the subsequent superimposition of filters and masks and combination into a familiar for linear regression one-dimensional array of values. A chromagram, spectral and mel-frequency characteristics are processed and passed as predicted parameters of the each of 3 MLP hidden layers of the ANN model (ANN model's architecture is presented on Figure 1). In the course of this solution, the Adam learning algorithm for ANN-model was selected as the most optimal

one, and the Accuracy and Hinge was chosen as the most optimal metric for quality assessment and target function of ANN-model learning.

2.1. Mathematical Description

The following variables and the sets described by them were introduced for the mathematical description of the problem and computation algorithms.

Batch normalization function for scaling vectors of values into the range from 0 to 1:

$$f_{BN}(x) = W_k \frac{x - E(x)}{\sqrt{D(x)}} + b_k \quad \forall k \in 1, \dots, d, \quad f_{BN}(x) \in [0, \dots, 1], \quad x = (x_1, \dots, x_d),$$

where x — vector of dimensionality d , $E(x)$ — mathematical expectation, $D(x)$ — variance, W, b — coefficients adjusted as a result of ANN-model operation (W — matrix of ANN-model weights, b — interference vector). It is used for processing of spectral and mel-frequency characteristics.

Vector of processed spectral rolloff:

$$x'_{sr} = f_{BN}(x_{sr}), \quad x'_{sr} = [0, \dots, 1],$$

where x_{sr} — vector of initial spectral rolloff.

Vector of processed spectral centroid:

$$x'_{sc} = f_{BN}(x_{sc}), \quad x'_{sc} = [0, \dots, 1],$$

where x_{sc} — vector of initial spectral centroid.

Vector of processed mel-frequency characteristics:

$$x'_{mfcc} = f_{BN}(x_{mfcc}), \quad x'_{mfcc} = [0, \dots, 1],$$

where x_{mfcc} — vector of initial mel-frequency characteristics.

Processed chromaticity vector, chromagram:

$$\begin{aligned} x'_{chroma} &= t', \\ x'_{ct} &= \max(x'_{chroma}), \\ \forall t \in [1, \dots, 12], \quad x'_{chroma} &\in [1, \dots, 12], \end{aligned}$$

where x_{chroma} — initial chromagram.

Function of the Binary Activation layer of the ANN model:

$$f_{BA}(x) = f_{BinaryActivation}(x) = \begin{cases} 0, & x < 0,5 \\ 1, & x > 0,5 \end{cases}, \quad x = [0, 1].$$

The main function of the ANN-model operation:

$$y = f_{ANN}(x'_{sr}, x'_{sc}, x'_{mfcc}, x'_{chroma}) = f_i(f_{BA}(f_{BN}(f_{LD}(f_{DO}(x'_{sr}, x'_{sc}, x'_{mfcc}, x'_{chroma}))))).$$

3. Testing the decision model

For the experiment, a sample of precedents was generated and trained from 1277 audio files and containing an average duration of 200 to 1000 seconds. The sample of precedents for training and test samples are divided into a percentage of 70 % and 30 % respectively. In processing the raw data, all vectors were reduced to values of 0 to 1. Training of the obtained ANN model of 20 epochs with 1750 training steps in each epoch using training sample of precedents was carried out. The obtained results were evaluated using the Accuracy and Hinge metrics. Testing of the trained ANN model was carried

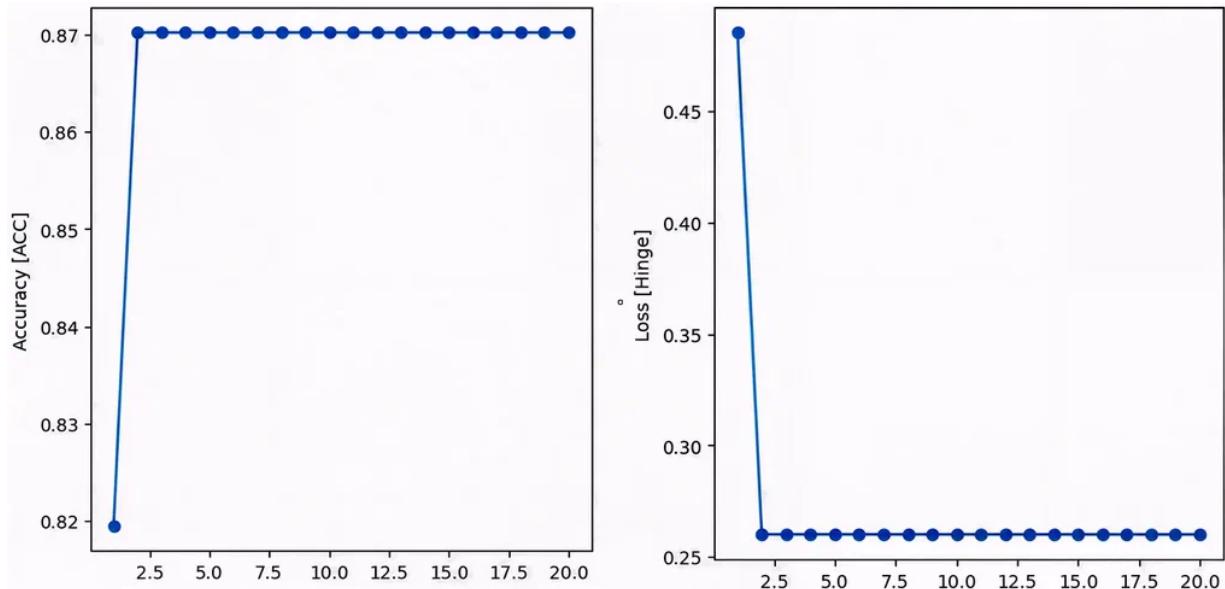


Fig. 2. Graphs of loss and accuracy on training

out using the test sample of precedents. The obtained test results were obtained by evaluating the obtained values by quality assessment metrics based on the edit string distance.

Illustration of the results of ANN-model training performance are given according to 2. An illustration of the performance results of the ANN model prediction are given according to 3.

As a result of testing the decision model on the automatically generated data, it was found that the average comparison metric of the obtained note sets varies between 82 %–88 %. This result indicates a good predictive ability of the obtained data processing model, but not perfect.

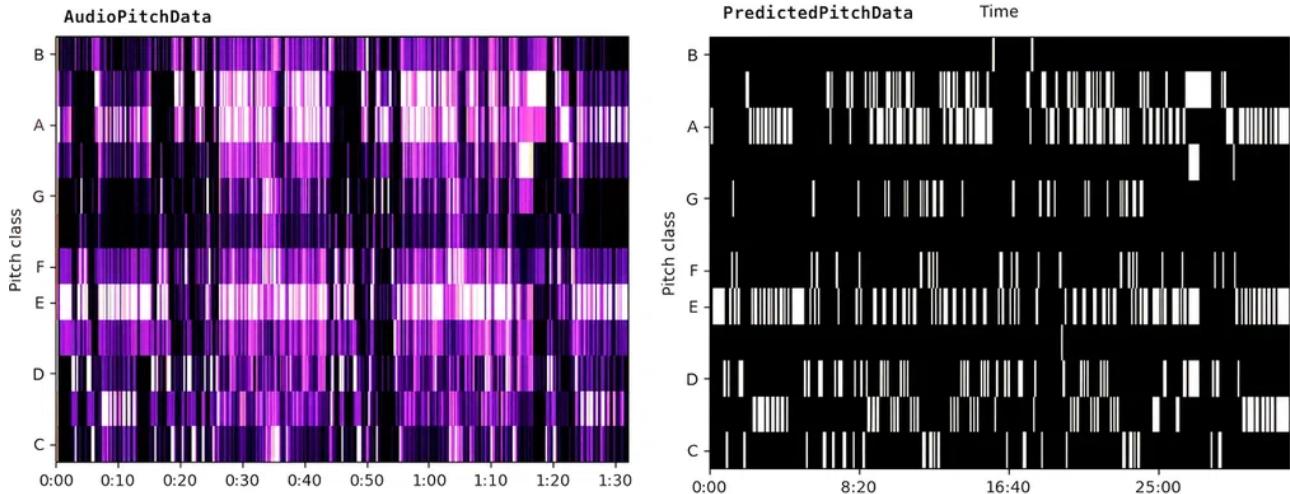


Fig. 3. Example illustration of the comparison graph of prediction and original melody data on training
Conclusion

The result of this article was the analysis of audio data and the implementation of a test ANN model for the developed solution to the problem of melody recognition. On the basis of this analysis, the methods, features, advantages and problems of the obtained solution of the above problem on the basis of the realized ANN-model are studied, the problems of the obtained implementation are formulated and the goals and objectives for further research are defined.

References

1. Azoff E. M. Neural Network Time Series Forecasting of Financial Markets // John Wiley & Sons Inc. – 1994.
2. Balas V. E., Motoc I. M., Barbulescu A. Combined Haar-Hilbert and Log Gabor Based Iris Encoders // New Concepts and Applications in Soft Computing, Studies in Computational Intelligence Series. – P. 1–26, Springer-Verlag Berlin Heidelberg (in press).
3. Bhattacharjee D., Basu D. K., Nasipuri M., Kundu M. Human Face Recognition Using Fuzzy Multilayer Perceptron // Journal of Soft Computing – A Fusion of Foundations, Methodologies and Applications archive. – January 2010. – Vol. 14. – P. 559–570.
4. Copeland B. J. What is Artificial Intelligence? // The Turing Archive for the History of Computing. – <http://www.alanturing.net/>, retrieved on May 29, 2012.
5. Jain L. C., Martin N. M. Fusion of Neural Networks // Fuzzy Systems and Genetic Algorithms : Industrial Applications. – CRC Press, 1998.
6. Kasabov N. K. Foundations of Neural Networks // Fuzzy Systems and Knowledge Engineering. – MIT Press, 1998.
7. Kohonen T. The Self-Organizing Map. // Proc. of the IEEE. – September 1990. – Vol. 78, no. 9 – P. 1464–1480.
8. McCulloch W., Pitts W. A Logical Calculus of Ideas Immanent in Nervous Activity. In: Bulletin of Mathematical Biophysics. – 1943. – Vol. 5. – P. 115–133.

РАСПОЗНАВАНИЕ РУКОПИСНОГО ТЕКСТА ТЕХНОЛОГИЯМИ МАШИННОГО ОБУЧЕНИЯ

Е. Э. Абдрахманова

Воронежский государственный университет

Аннотация. Данная статья посвящена исследованию методов распознавания рукописного текста технологиями машинного обучения, также разработке и анализу модели, способной распознать рукописный текст на изображении. Рассмотрены ключевые моменты этой задачи, связанные с формированием подходящего набора данных, учетом стилистических особенностей различных языков и алфавитов, создана модель для обработки и распознавания, учитывая раннее описанные способности, а также обучена на тестовом наборе данных.

Ключевые слова: распознавание рукописного текста, машинное обучение, CNN, RNN, LSTM, СТС, градиент, обучение, функция потерь, генерация тренировочных данных, обработка последовательностей, нейросетевые модели.

Введение

В современном мире, особенно в последние годы актуальной задачей становится автоматизация обработки информации, объем которой увеличивается в геометрической прогрессии, и значительная часть этой информации хранится на бумажных носителях — в большинстве случаев, написанная от руки, в качестве рукописных документов.

Проблема автоматического распознавания рукописного текста заключается в том, что человеческий почерк очень разнообразен: помимо того, что одна и та же буква может быть написана тысячами вариаций, даже одна вариация может быть разделена еще на несколько, в зависимости от толщины и цвета ручки, качества бумаги и так далее. При этом невозможно составить один набор правил, по которому определяется конкретная буква, а сами они, особенно в русском языке (кириллице) могут быть очень схожи.

Перспективным и наилучшим направлением является применение методов машинного обучения, в частности, глубоких нейронных сетей. Алгоритм машинного обучения анализирует тысячи и миллионы примеров и сам для себя составляет набор очень сложных правил, которые мог бы не заметить обычный человек. Сам этот алгоритм может быть схож с обучением ребенка чтению: сначала он видит букву в букваре, потом в словах, потом в книжках, потом учит в школе в прописях и так на протяжении всей жизни, и может различать буквы, в каком виде они бы ни были, в любом шрифте, а не только в том, который видел раньше.

Цель данной работы заключается в изучении и исследовании существующих методов и алгоритмов машинного обучения, а также разработка и анализ модели, способной эффективно обрабатывать и распознавать рукописный текст с помощью данных методов и алгоритмов.

1. Формирование многоязычного датасета

Обучение модели, как и человека, не обходится без достаточно большого количества данных, в основном, это тысячи или десятки тысяч размеченных примеров. В случае с распознаванием рукописного текста, их объем увеличивается в связи с количеством различных языков. Необходимо сделать модель, умеющую распознавать текст на четырех из шести официальных языках ООН: английский, французский, испанский, русский. Все эти языки имеют свои осо-

бенности, так как они имеют различные алфавиты: латиница и кириллица, и разные особенности каллиграфии, и для каждого из них нужно большое количество данных для обучения.

Основная проблема — отсутствие готовых, подходящих датасетов для всех языков. В качестве решения, предлагается обращение к генеративной нейросети, синтезирующей подходящие датасеты для необходимого объема размеченных данных. Под разметкой подразумевается добавление к исходным данным, в случае данной задачи — к изображениям рукописного текста, специальных меток, которые объясняют модели, что именно находится на картинке, «эталонное представление».

2. Архитектурное решение

В качестве архитектурного решения наиболее выгодна архитектура, соединяющая CNN для извлечения признаков и RNN для анализа последовательностей.

Общая схема модели представлена на рис. 1.



Рис. 1. Общая архитектура модели для распознавания рукописного текста

2.1. Сверточная часть (CNN)

CNN — сверточные сети — используют свертки для возможности выделения форм без участия человека и эффективно работают с изображениями для выделения важных признаков. В данной задаче важен такой подход, так как, несмотря на различие почерков, каждая буква имеет свои признаки, по которым сам человек может отличить одну букву от другой: в букве «А» (заглавной) всегда будет две наклонных и линия между ними, независимо от почерка, и к ней может еще что-то прибавляться, изменяться угол наклона, но это те признаки, которые есть только у заглавной «А». Из этого можно сделать вывод, что принцип работы CNN вдохновлен и схож с организацией зрительной коры головного мозга:

CNN состоит из нескольких слоев, и каждый из этих слоев отвечает за определенные действия:

- **Сверточные слои:** применяют набор фильтров к изображению (маленькая матрица размером 3×3 или 5×5), которые скользят по входному изображению, умножают значение пикселя под окном на соответствующие веса, в итоге сумма дает одно значение карты признаков на фильтр, тогда несколько фильтров дают набор карт признаков. При этом ранние слои выделяют низкоуровневые признаки (углы и контуры), тогда как глубокие слои выделяют все более сложные структурные элементы и паттерны.

- **Слой активации** добавляет нелинейность через функции типа ReLU, что ускоряет обучение, пропуская части с отсутствующими важными признаками, и позволяет модели обучится сложным зависимостям.

- **Слои пулинга:** уменьшают размерность карт признаков, сохраняя наиболее важную информацию, выделяя доминирующие значения, при этом повышает инвариантность к малым искажениям, снижая чувствительность к дефектам и шуму.

- **Полносвязный слой** преобразует двумерные карты признаков в одномерные векторы. Теперь они будут использоваться для финального предсказания в классическом понимании нейронных сетей, где каждый нейрон связан с нейронами предыдущего слоя.

В контексте задачи распознавания рукописного текста CNN способна выявлять инвариантные признаки символов, независимо от вариативности почерка.

2.2. Рекуррентная часть (LSTM)

RNN — рекуррентные нейронные сети — специализируются на обработке последовательности данных, они используют одну и ту же функцию на каждом элементе последовательности, при этом не забывая о результатах предыдущих элементов. В отличие от традиционных нейронных сетей, RNN способны обработать последовательность произвольной длины, при этом сохраняя связь между элементами — что важно для рекуррентных сетей. Также существуют двунаправленные RNN, которые расширяют возможности до обработки последовательностей в обе стороны.

В традиционных RNN существует проблема, связанная с тем, что на действительно больших последовательностях происходит затухание градиента (вектор частных производных функции потерь по всем параметрам модели), что снижает эффективность. Тогда в качестве решения этой проблемы была создана LSTM — сеть с долговременной и кратковременной памятью, основная идея которой заключалась во внедрении структур («ворот»), регулирующих поток информации внутри сети, что обеспечивает более стабильный градиент.

Для модели распознавания текста в итоге была выбрана, более подходящая данной задаче, LSTM.

2.3. Процесс обучения

Для обучения модели использована функция потерь Connectionist Temporal Classification (CTC), специально разработанная для случаев, когда соответствие между входными и выходными последовательностями неизвестно. И для задачи распознавания текста это особенно актуально, так как отсутствует явное выравнивание между временными шагами и выходными символами.

Математическая основа СТС использует алгоритм forward-backward для расчёта вероятности всех возможных выравниваний между входной и выходной последовательностями. Этот алгоритм эффективно вычисляет сумму вероятностей всех допустимых путей, которые сопоставляют входную последовательность с выходной, включая специальный токен для отсутствующего символа, на рис. 2 приведена схема процесса обучения, включающая в себя этапы, описанные ниже.

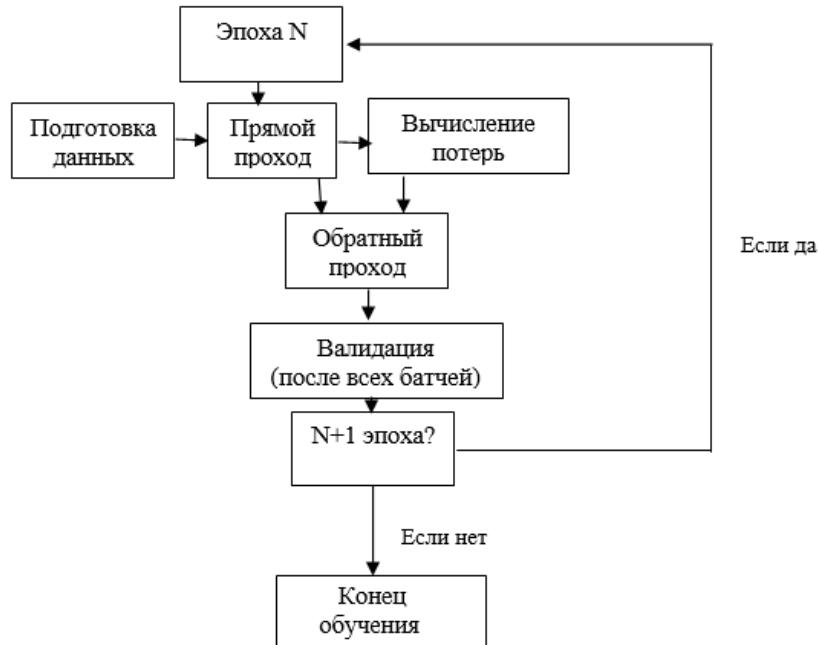


Рис. 2. Схема процесса обучения модели

Процесс обучения включает в себя 3 основных этапа:

– **Прямой проход (Forward Pass).** На данном этапе происходит предобработка изображений, включающая в себя преобразование в градации серого, ресайзинг до фиксированного размера и нормализация значений пикселей в диапазон $[-1, 1]$. Далее идет извлечение пространственных признаков с помощью CNN и анализ последовательностей через LSTM, и на выходе получаем значения вероятностей символов для каждого временного шага.

– **Вычисление функции потерь.** Здесь CTC Loss сравнивает предсказанные последовательности с истинными метками, учитывая все возможные выравнивания.

– **Обратный проход и обновление весов.** Здесь происходит вычисление градиентов функции потерь относительно параметров модели и обновление весовых коэффициентов (числовых значений, которые присвоены соединениям между нейронами, по сути, само обучение — настройка этих соединений) с помощью оптимизатора Adam.

3. Анализ

Данная архитектура сочетает в себе сверточные и рекуррентные сети, что обеспечивает преимущества для задачи многоязычного распознавания рукописного текста. Инвариантность CNN позволяет устойчиво распознать символы независимо от вариаций почерка, масштаба, ориентации, наличия шумов, а обработка LSTM обеспечивает учет лингвистических зависимостей между символами, что важно для различных языков со своими грамматическими структурами.

Если сравнивать с традиционными подходами, то данная архитектура имеет существенные преимущества, так как традиционные OCR-системы имеют низкую адаптивность к вариациям почерка, а обычные CNN-классификаторы игнорируют контекстные зависимости, что в данной задаче недопустимо.

Но несмотря на преимущества, могут возникнуть проблемы, связанные с вычислительной сложностью, требующей значительных ресурсов, или с несбалансированностью датасетов из-за различных языков.

Заключение

В рамках данной работы была проделана значительная аналитическая часть по проектированию системы распознавания рукописного текста. В ходе исследования была рассмотрена архитектура CNN+RNN и обоснована ее эффективность, работающая в сочетании с CTC Loss для обучения модели. Также рассмотрены решения проблем связанных с многоязычностью поставленной задачи, и проанализированы преимущества и ограничения данного подхода.

С одной стороны, модель выигрывает у более традиционных подходов, но с другой стороны, все еще имеются проблемы, связанные с огромным количеством затрачиваемых ресурсов, возможным смещением модели из-за различия языков и схожестью некоторых символов в кириллице и на латинице, что может вызывать некоторые ошибки.

Литература

1. Коэльо Л. П. Построение систем машинного обучения на языке Python / Л. П. Коэльо, В. Ричарт ; пер. с англ. А. А. Слинкина – Москва : ДМК Пресс, 2019. – 302 с.
2. Николенко С. Глубокое обучение. Погружение в мир нейронных сетей / С. Николенко, А. Кадурин, Е. Архангельская. – Санкт-Петербург : Питер, 2018. – С. 81–91.
3. Постолит А. В. Основы искусственного интеллекта в примерах на Python / А. В. Постолит. – Санкт-Петербург : БХВ, 2021. – С. 381–387.

4. Рашика С. Python и машинное обучение / С. Рашика. – Москва : ДМК Пресс, 2015. – 116 с.
5. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / П. Флах. – Москва : ДМК Пресс, 2015. – 205 с.
6. PyTorch documentation – URL: <https://docs.pytorch.org/docs/stable/index.html> (дата обращения 15.10.2025).

**СРАВНИТЕЛЬНЫЙ АНАЛИЗ ФУНКЦИЙ ПОТЕРЬ
ДЛЯ ПРОГНОЗИРОВАНИЯ РИСКА СЕРДЕЧНО-СОСУДИСТЫХ ЗАБОЛЕВАНИЙ
С ИСПОЛЬЗОВАНИЕМ ГЛУБОКОГО ОБУЧЕНИЯ**

А. А. Алексеенко, Н. Н. Максимова

Амурский государственный университет

Аннотация. В работе исследуется применение полно связной нейронной сети для прогнозирования риска сердечно-сосудистых заболеваний на основе анамнестических данных из открытого набора «Personal Key Indicators of Heart Disease». Особое внимание уделено проблеме сильной несбалансированности классов, для решения которой использованы ручная балансировка классов ($class_weight = \{0: 1.0, 1: 20.0\}$) и сравнение функций потерь — Binary Cross-Entropy и Focal Loss. Показано, что модель с Binary Cross-Entropy обеспечивает высокий Recall (0.89), что делает её предпочтительной для медицинских скрипинговых задач, несмотря на низкую точность положительных предсказаний.

Ключевые слова: искусственный интеллект, машинное обучение, нейронные сети и глубокое обучение, прогнозирование сердечно-сосудистых заболеваний, бинарная классификация, несбалансированные данные, Focal Loss, Binary Cross-Entropy, Recall.

Введение

Искусственный интеллект все более активно используется в медицине для анализа больших массивов данных, диагностики и прогнозирования рисков различных заболеваний [1, 2]. Одним из наиболее перспективных направлений в медицине является применение нейронных сетей, которые способны находить корреляции между медицинскими показателями.

Заболевания системы кровообращения все еще являются ведущей причиной смерти в России [3]. Соответственно, одной из актуальных задач искусственного интеллекта и машинного обучения является раннее выявление рисков сердечных патологий. Это позволит эффективно распознавать группы риска и предупреждать развитие заболеваний.

Цель данной работы — построение и обучение полно связной нейронной сети для определения прогнозирования риска сердечных патологий на основе медицинских параметров человека.

1. Описание набора данных

В качестве обучающего набора был использован набор данных «Personal Key Indicators of Heart Disease» [4]. Данный набор является производной (обработанной) версией официальных данных, собранных в рамках Behavioral Risk Factor Surveillance System (BRFSS) — ежегодной системы опросов населения США, проводимой Центрами по контролю и профилактике заболеваний (CDC, Centers for Disease Control and Prevention).

Датасет содержит 17 признаков (возраст, пол, уровень холестерина, артериальное давление и другие показатели здоровья) и 1 целевую переменную (наличие или отсутствие сердечного заболевания). Размер исходного набора составляет 319795 строк.

В рамках предварительной обработки [5] были удалены дубликаты, осуществлена перекодировка категориальных переменных, выполнен анализ корреляционной зависимости между переменными. В дополнение к этому был проведен анализ числовых данных на выбросы. Для такого анализа существует множество визуальных (Box plot, гистограмма, Scatter plot, QQ-plot) и статистических (IQR, Z-оценка, модифицированная Z-оценка, метод трёх сигм) методов, а также методы на основе машинного обучения. На основе проведенного анализа выбросов по мето-

ду межквартильного размаха (IQR) можно сделать следующие выводы: большинство значений, формально определяемых как выбросы, не являются ошибками, а отражают реальные и клинически значимые состояния пациентов. Тем самым, удаление всех статистических выбросов в данном датасете нецелесообразно. Вместо этого следует применять гибкий, предметно-ориентированный подход: удалять только заведомо невозможные значения и сохранять клинически обоснованные экстремумы. Были проверены границы на физическую допустимость (BMI из промежутка [10, 60], SleepTime — [0, 24], Physical и MentalHealth — [0, 30]). Такой анализ показал, что только BMI содержит явные выбросы (492 записи), которые были удалены из набора.

Окончательно для построения моделей машинного обучения были использованы 301225 строк. Эти данные были разделены на тренировочную и тестовую выборки в соотношении 80/20 с параметром `random_state = 42`. Кроме того, при разделении учтена несбалансированность классов посредством настройки `stratify = y` (так гарантируется, что пропорции классов в целевой переменной будут одинаковыми как в обучающей, так и в тестовой выборках).

Кроме того, из-за несбалансированности («Yes» — 9 %, «No» — 91 %) при оценке качества обучения нельзя пользоваться метрикой Accuracy; в этом случае основным показателем качества будет выступать метрика Recall, поскольку модели будут предсказывать наличие заболевания, а также дополнительные метрики, такие как F1-мера и ROC-AUC. Кроме того, этот факт следует учесть при выборе архитектуры и настроек нейронной сети.

Все процедуры загрузки и обработки данных, проводились средствами библиотек языка Python в среде для разработки и выполнения программного кода Google Colaboratory [6].

2. Архитектура нейронной сети

Предложенная архитектура искусственной нейронной сети представляет собой полносвязную многослойную сеть последовательной структуры [7], предназначенную для решения задачи бинарной классификации. Сеть состоит из входного слоя, четырёх скрытых слоёв с убывающим количеством нейронов (256 / 128 / 64 / 32) и одного выходного нейрона.

Для повышения устойчивости обучения и снижения риска переобучения в архитектуру последовательно интегрированы следующие техники регуляризации и стабилизации:

- Batch Normalization (применяется после каждого скрытого слоя для нормализации распределения активаций, что ускоряет сходимость и повышает обобщающую способность модели) [8];
- Dropout с постепенно уменьшающимся коэффициентом (от 0.5 до 0.2) используется для стохастического отключения нейронов в процессе обучения, что снижает взаимную адаптацию нейронов и улучшает робастность модели [9].

В качестве функции активации в скрытых слоях выбрана ReLU (Rectified Linear Unit), обеспечивающая нелинейность преобразований при сохранении вычислительной эффективности. Выходной слой использует сигмоидальную функцию активации, что позволяет интерпретировать выход модели как вероятность принадлежности к положительному классу.

Такая архитектура сбалансирована с учётом объёма и структуры исходных данных, обеспечивая достаточную выразительную мощность для выявления сложных нелинейных зависимостей между факторами риска и целевым признаком, при одновременном контроле переобучения за счёт многоуровневой регуляризации. Общее количество параметров (весов) модели составляет 49793.

3. Настройки нейронной сети

В качестве функции потерь были использованы два варианта. Первый из них — стандартная для задач двухранговой классификации Binary Cross-Entropy Loss [10], которая определяется по формуле

$$L(y, \hat{y}) = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})].$$

Эта функция потерь хорошо работает с сигмоидальной активацией и чувствительна к ошибкам в обоих направлениях: ложноположительным и ложноотрицательным.

Вторая функция ошибки — Focal Loss, которая была предложена в работе [11] для решения проблемы сильного дисбаланса между классами. Математическая запись формулы имеет вид

$$L_{\text{focal}}(y, p) = -\alpha_t (1 - p_t)^\gamma \log(p_t),$$

где

$$p_t = \begin{cases} p, & \text{если } y = 1, \\ 1 - p, & \text{если } y = 0, \end{cases} \quad \alpha_t = \begin{cases} \alpha, & \text{если } y = 1, \\ 1 - \alpha, & \text{если } y = 0, \end{cases}$$

$y \in \{0, 1\}$ — истинная метка, $p \in (0, 1)$ — предсказанная моделью вероятность класса 1, $\alpha \in (0, 1)$ — вес редкого класса (по умолчанию $\alpha = 0.75$), $\gamma \geq 0$ — фокусирующий параметр (по умолчанию $\gamma = 2$).

Для минимизации функции потерь использовался оптимизатор Nadam, сочетающий в себе преимущество оптимизаторов Adam и Nesterov Accelerated Gradient [12]. Обычный метод Adam использует экспоненциально скользящие средние первого (градиент) и второго (квадрат градиента) моментов для адаптивного изменения шага обучения по каждому параметру. Однако он не учитывает «коррекцию по инерции», как в методе Nesterov, где обновление параметров происходит не по текущему градиенту, а по прогнозируемому (вперёд-смотрящему) направлению. Nadam модифицирует Adam, внедряя импульс по схеме Nesterov непосредственно в оценку первого момента, что позволяет ускорить сходимость, уменьшить «перелёты» через оптимум, улучшить стабильность обучения, особенно в начальных эпохах.

Для оценки качества модели в процессе обучения отслеживались две ключевые метрики — Precision (точность) и Recall (полнота), что позволяет комплексно контролировать баланс между ложноположительными и ложноотрицательными предсказаниями. Такой выбор метрик обусловлен спецификой медицинской задачи, где критически важно не только минимизировать количество ложных тревог, но и максимально полно выявлять пациентов с сердечно-сосудистыми заболеваниями.

Обучение проводилось в течении 50 эпох с применением ранней остановки EarlyStopping [13]. Обучение прерывалось, если в течение 10 последовательных эпох не наблюдалось улучшение значения функции потерь на валидационной выборке (val_loss). При этом сохранялись веса модели, соответствующие наилучшему значению контролируемой метрики (восстановление лучших весов — restore_best_weights=True). Такой подход позволил избежать избыточного обучения и повысить обобщающую способность полученной нейронной сети.

В связи с выраженной несбалансированностью целевой переменной при обучении нейронной сети была применена ручная настройка балансировки классов посредством функции class_weight_dict = {0: 1.0, 1: 20.0}. Это означает, что при вычислении функции потерь модель будет в 20 раз сильнее «наказываться» за ошибки на объектах класса 1 (положительный класс — «есть сердечно-сосудистое заболевание»), чем за ошибки на объектах класса 0 («нет заболевания»). Такой подход приводит к большей продолжительности обучения, но к лучшему качеству.

Обучение нейронной сети выполнялось с размером батча (batch size), равным 256, что обеспечивает компромисс между стабильностью градиентных обновлений и вычислительной эффективностью. Для оценки обобщающей способности модели и контроля за переобучением использовалось разделение выборки с выделением 20 % данных на валидационную подвыборку (validation_split = 0.2).

4. Анализ качества прогнозирования

Процесс обучения нейронной сети представлен на рис. 1-2. На рис. 1 отображена динамика изменения ключевых метрик, на рис. 2 — функции ошибки. После обучения была осущест-

влена проверка качества на тестовом наборе. В табл. 1 представлены значения всех метрик, на рис. 3 — графики ROC-кривых.

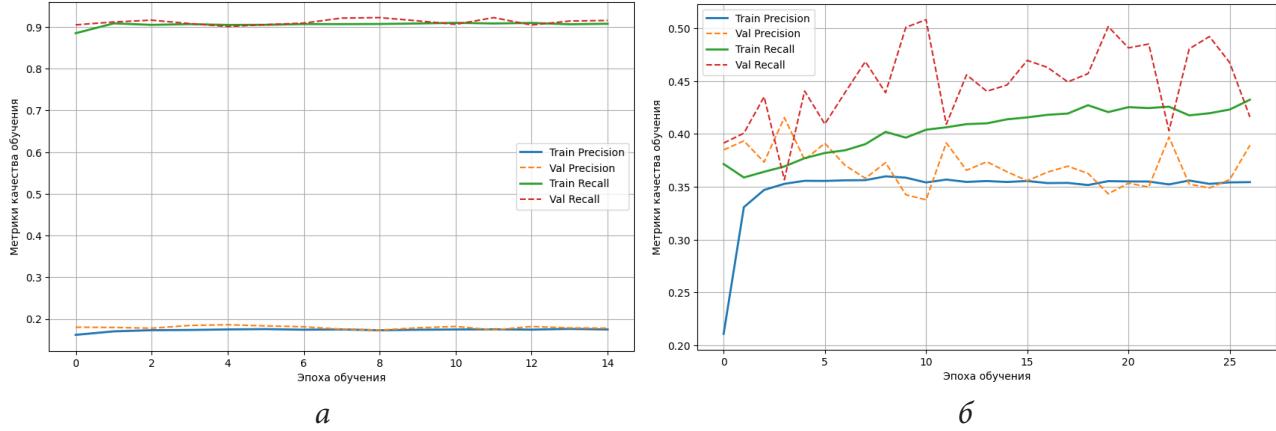


Рис. 1. Динамика изменения метрик *Precision* и *Recall* на тренировочной и валидационной выборках (а — с функцией *Binary Cross-Entropy Loss*, б — с функцией *Focal Loss*)

Рис. 1а показывает, что значение метрики Recall стабильно высокое (около 0.90) как на тренировочной, так и на валидационной выборке. Это говорит о том, что модель очень хорошо находит больных пациентов, что является главной целью задачи. Метрика Precision является крайне низкой (около 0.17–0.18); это означает, что из всех предсказаний «больной» лишь 17–18 % действительно являются больными, остальные — ложные срабатывания. Разница между Train и Val для обеих метрик минимальна — это указывает на отсутствие переобучения.

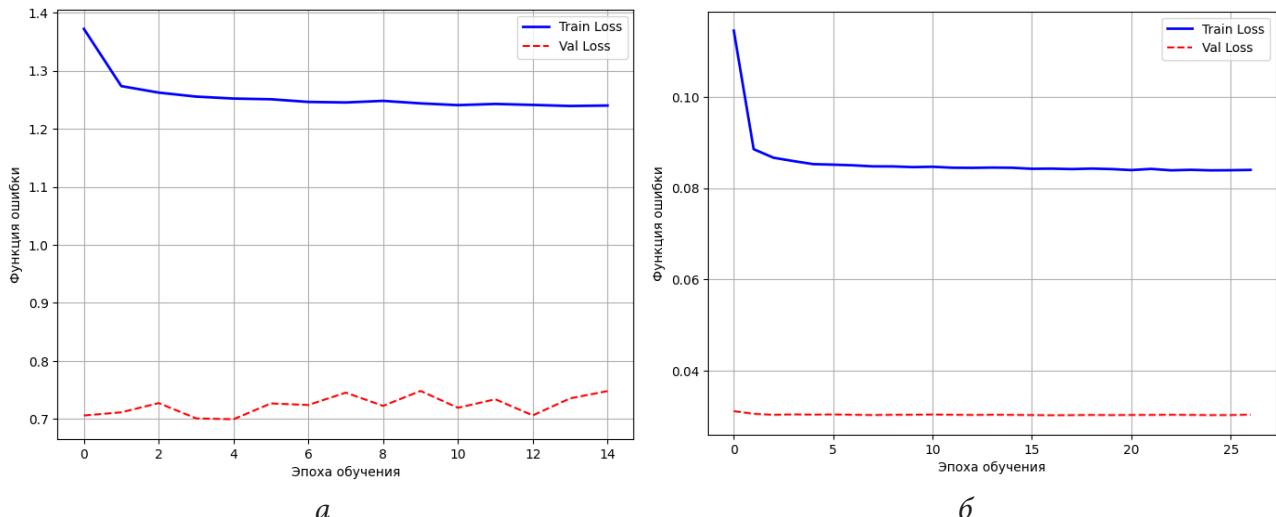


Рис. 2. Динамика изменения функции ошибки (а — для *Binary Cross-Entropy Loss*, б — для *Focal Loss*)

По рис. 1б видно, что метрика Recall значительно ниже (колеблется в пределах 0.40–0.5). Это означает, что модель находит только 40–50 % больных пациентов, что существенно хуже, чем в случае а). Метрика Precision заметно выше (около 0.35–0.4), т. е. из всех предсказаний «больной» теперь примерно 35–40 % действительно больны — точность прогноза повысилась. Наблюдается небольшая разница между Train и Val, особенно по Recall, что может указывать на начало переобучения. Динамика более шумная — особенно для Precision на валидации — что характерно для Focal Loss, который фокусируется на сложных примерах.

На рис. 2а наблюдается значительный разрыв между Train Loss и Val Loss — это типичный признак того, что модель обучается на «легких» примерах, которые преобладают в данных (здо-

ровые пациенты). Валидационная ошибка низкая, потому что модель почти всегда предсказывает 0 («здоров»), что верно для большинства случаев. Focal Loss значительно уменьшает общее значение потерь (рис. 26), особенно на тренировочной выборке. Это связано с тем, что она «фокусируется» на трудных примерах (в данном случае — на больных пациентах), и её формула подавляет вклад легко классифицируемых примеров. Однако, как показано в рис. 16, это привело к снижению Recall. Низкий Val Loss может быть обманчивым — он отражает хорошую работу модели на «легких» примерах, но не гарантирует хорошего качества на редком классе.

Таблица 1

Значение метрик качества на тестовом наборе

Функция ошибки	Класс	Метрика				
		precision	recall	f1-score	accuracy	AUC
Binary Cross-Entropy Loss	0	0.98	0.61	0.75	0.63	0.833
	1	0.18	0.89	0.30		
Focal Loss	0	0.94	0.93	0.93	0.87	0.834
	1	0.35	0.44	0.39		

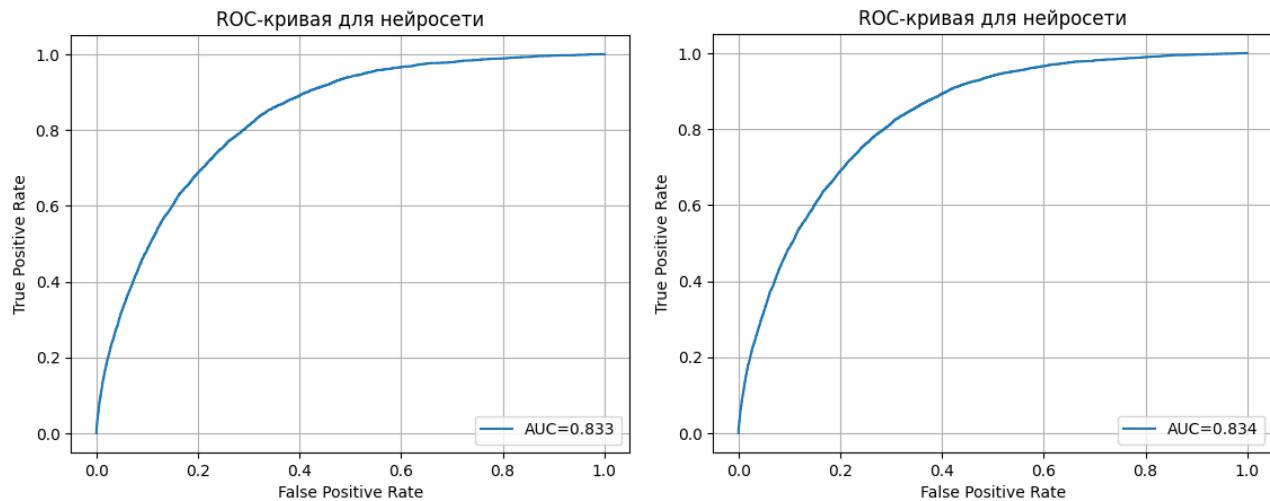


Рис. 3. ROC-кривые для обученных моделей на тестовом наборе
(а — для Binary Cross-Entropy Loss, б — для Focal Loss)

Результаты, представленные в табл. 1, демонстрируют фундаментальный компромисс между полнотой (Recall) и точностью (Precision) при прогнозировании сердечно-сосудистых заболеваний. Модель, обученная с использованием Binary Cross-Entropy Loss, обеспечивает высокий Recall (0.89) для положительного класса, что критически важно в медицинских приложениях. В то же время Focal Loss, несмотря на улучшение общей точности и баланса метрик, значительно снижает способность модели выявлять больных пациентов (Recall = 0.44), что делает её менее пригодной для практического использования в скрининговых системах.

Заключение

В рамках проведённой работы была построена и обучена полно связная нейронная сеть для прогнозирования риска сердечно-сосудистых заболеваний на основе анамнестических и демографических данных из набора «Personal Key Indicators of Heart Disease». Исходный датасет содержал более 319 тысяч записей, после предварительной обработки (удаления дубликатов

и клинически недопустимых выбросов) для обучения использовано 301 225 объектов. В ходе экспериментов была разработана архитектура нейронной сети с четырьмя скрытыми слоями ($256 \rightarrow 128 \rightarrow 64 \rightarrow 32$ нейронов), интегрированы методы регуляризации (Batch Normalization и Dropout) и протестированы две функции потерь (Binary Cross-Entropy и Focal Loss). Для обучения применялись оптимизатор Nadam, ручная балансировка классов (`class_weight = {0: 1.0, 1: 20.0}`) и механизм ранней остановки. Все вычисления выполнялись с использованием библиотек TensorFlow и Keras на языке программирования Python в среде Google Colaboratory.

Полученные результаты показали, что модель, обученная с использованием Binary Cross-Entropy Loss в сочетании с весовой балансировкой классов, обеспечивает наилучшее соотношение между полнотой и точностью диагностики по сравнению с Focal Loss, что делает её более пригодной для применения в скрининговых системах. Разработанная модель может быть использована для раннего выявления групп риска сердечно-сосудистых заболеваний, что способствует своевременному вмешательству и снижению общей заболеваемости. Данное исследование демонстрирует потенциал методов машинного обучения в анализе несбалансированных медицинских данных и подчеркивает важность правильного выбора метрик и стратегий обучения при решении практически значимых задач в здравоохранении.

Литература

1. *Jiang F. Artificial intelligence in healthcare: past, present and future / F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Li, X. Shen // Stroke and Vascular Neurology. – 2017. – Vol. 2, № 4. – <https://doi.org/10.1136/svn-2017-000101>*
2. *Litjens G. A survey on deep learning in medical image analysis / G. Litjens, Th. Kooi, B. Ehteshami Bejnordi, A. Arindra Adiyoso Setio, F. Ciompi, M. Ghafoorian, Jeroen A.W.M. van der Laak, B. van Ginneken, C. I. Sánchez // Medical Image Analysis. – 2017. – V. 42. – P. 60–88. – <https://doi.org/10.1016/j.media.2017.07.005>*
3. *Коржавая А. Н. Статистический анализ смертности населения в Российской Федерации / А. Н. Коржавая, А. С. Мерзликина // Молодой ученый. – 2024. – № 18 (517). – С. 476–478. – Режим доступа: <https://moluch.ru/archive/517/113586>*
4. *Indicators of Heart Disease (2022 UPDATE) [Электронный ресурс]. URL: <https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease> (дата обращения: 10.02.2025)*
5. *Алексеенко А. А. Предсказание риска сердечных патологий: моделирование на основе анализа больших данных / А. А. Алексеенко, Н. Н. Максимова // XXVII Всероссийская студенческая научно-практическая конференция Нижневартовского государственного университета: Материалы конференции, Нижневартовск, 09–10 апреля 2025 года. – Нижневартовск: Нижневартовский государственный университет, 2025. – С. 159–163.*
6. *Google Colaboratory [Электронный ресурс]. URL: <https://colab.google> (дата обращения: 09.10.2024)*
7. *Гудфеллоу Я. Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвиль, пер. с англ. А. А. Слинкина. – 2 изд., испр. – М. : ДМК Пресс. – 2018. – 652 с.*
8. *Ioffe S. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift / S. Ioffe, Ch. Szegedy // ICML (International Conference on Machine Learning). – 2015. – <https://doi.org/10.48550/arXiv.1502.03167>*
9. *Srivastava N. Dropout: A Simple Way to Prevent Neural Networks from Overfitting / N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov // Journal of Machine Learning Research. – 2014. – V. 15(1). – P. 1929–1958. – <http://jmlr.org/papers/v15/srivastava14a.html>*
10. *Hinton G. E. A Fast Learning Algorithm for Deep Belief Nets / G.E. Hinton, S. Osindero, Y.-Wh. Teh // Neural Computation. – 2006). – V. 18 (7). – P. 1527–1554 2006. – <https://doi.org/10.1162/neco.2006.18.7.1527>*

11. *Lin Ts.-Yi.* Focal Loss for Dense Object Detection / Ts.-Yi Lin, P. Goyal, R. Girshick, K. He, P. Dollar // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2020. – V. 42, № 2. – P. 318–327. – <https://pubmed.ncbi.nlm.nih.gov/30040631/>
12. TensorFlow Documentation [Электронный ресурс]. URL: https://www.tensorflow.org/api_docs/python/tf/keras/optimizers/Nadam (дата обращения: 03.09.2025)
13. TensorFlow Documentation [Электронный ресурс]. URL: https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping (дата обращения: 03.09.2025)

СРАВНЕНИЕ АППАРАТНЫХ ПЛАТФОРМ КОМПЬЮТЕРНОГО ЗРЕНИЯ В ЗАДАЧЕ АВТОНОМНОГО ТРАНСПОРТА

А. М. Аникандров

Воронежский государственный университет

Аннотация. В работе проведен сравнительный анализ аппаратных платформ компьютерного зрения, применяемых в системах восприятия автономного транспорта. Рассмотрены ключевые сенсорные платформы: моноскопические и стереоскопические камеры, лидары и радары. В рамках практической части представлена реализация системы стереовидения на базе библиотеки OpenCV для Python. Показано, что каждый из рассмотренных подходов обладает характерными преимуществами и ограничениями по точности, надежности, стоимости и вычислительной сложности, что обуславливает их применимость в различных сценариях эксплуатации.

Ключевые слова: компьютерное зрение, автономный транспорт, стереовидение, OpenCV, Python, лидар, радар, детекция объектов, сенсорные системы, семантическая сегментация, оценка глубины.

Введение

Развитие технологий автономной навигации является одним из ключевых направлений современной робототехники и машинного обучения. Расширение областей применения рождает широкий спектр технических требований к системам восприятия, которые составляют основу любого автономного транспорта. Фундаментальной задачей, от решения которой напрямую зависит функционирование автономного транспорта, является построение точной и надежной модели окружающего пространства в реальном времени.

Существующие аппаратные средства компьютерного зрения (камеры, лидары, радары) обладают существенно различающимися характеристиками по точности, надежности, стоимости и работоспособности в различных внешних условиях. Актуальность данного исследования обусловлена необходимостью систематического сравнения этих решений для выявления их применимости и ограничений в различных сценариях эксплуатации.

1. Обзор существующих аппаратных решений

Формирование модели окружающего пространства основывается на данных, получаемых от комплекса сенсоров. Ключевыми аппаратными платформами, используемыми для решения задач компьютерного зрения, являются моноскопические и стереоскопические камеры, лидары и радары. Каждая из этих платформ уникальным принципом действия, что обуславливает их преимущества и ограничения.

1.1. Моноскопические камеры

Принцип действия: Моноскопическая камера представляет собой сенсор, регистрирующий проекцию трехмерного мира на двумерную плоскость в виде изображения. Изображение формируется за счет света, отраженного от объектов и прошедшего через систему линз на светочувствительную матрицу.

Преимущества:

- Высокая информативность: Камеры предоставляют богатую текстуру и цветовую информацию, что является критически важным для алгоритмов распознавания и классификации объектов.

- Высокое пространственное разрешение: Позволяют детектировать мелкие объекты и точно определять их границы.

- Относительно низкая стоимость: По сравнению с остальными сенсорами, монокамеры являются одним из наиболее экономичных решений.

- Пассивность работы: Не излучают сигналы, что позволяет работать скрытно и не создавать помех другим сенсорам.

Ограничения:

- Отсутствие непосредственного измерения глубины: Для оценки расстояния до объектов требуются сложные алгоритмы, основанные на предположениях о размерах объектов или анализе перспективы, что снижает точность и надежность.

- Сильная зависимость от условий освещения: Существенно затруднена или невозможна работа в темное время суток, при ярком солнце, в условиях тумана.

- Высокая вычислительная нагрузка: Обработка видеопотока высокого разрешения для задач детектирования и семантической сегментации требует значительных вычислительных ресурсов.

1.2. Стереоскопические камеры

Принцип действия: Стереоскопическая система (стереокамера) состоит из двух синхронизированных камер, разнесенных на известное расстояние (базис). Принцип работы основан на триангуляции: путем сопоставления пикселей на двух изображениях (поиска стереопар) вычисляется карта смещений (disparity map), которая затем преобразуется в карту глубины (depth map).

Преимущества:

- Прямое измерение глубины: В отличие от монокамер, стереосистемы позволяют получать плотные 3D-данные о сцене в реальном времени, что критически важно для построения карт окружения и обнаружения препятствий.

- Сохранение семантической информации: Как и монокамеры, предоставляют цветное изображение, пригодное для классификации объектов.

- Пассивность работы: Так же, как и монокамеры не создают помех для работы других сенсоров.

Ограничения:

- Зависимость от текстуры поверхности: Для успешного сопоставления стереопар поверхности объектов должны иметь достаточную текстуру. Гладкие, однородные поверхности (стена, ровная дорога) создают «слепые зоны».

- Высокие вычислительные затраты: Алгоритмы стереосопоставления в реальном времени требуют мощных процессоров.

- Чувствительность к калибровке: Точность измерений напрямую зависит от точности калибровки и стабильности взаимного положения камер. Вибрации и перепады температуры могут нарушить калибровку.

- Унаследованные проблемы камер: Как и монокамеры, стереосистемы чувствительны к условиям освещения.

1.3. Лидары (Light Detection and Ranging)

Принцип действия: Лидар является активным сенсором, который измеряет расстояние до объектов, испуская лазерные импульсы и регистрируя время их возврата после отражения (Time-of-Flight). Сканирующий механизм позволяет создавать высокоточные 3D-карты окружения в виде облака точек (point cloud).

Преимущества:

- Высокая точность измерения расстояния: Погрешность измерений современных лидаров составляет несколько сантиметров.
- Независимость от освещения: Способны работать как днем, так и ночью, поскольку используют собственный источник излучения.
- Высокое угловое разрешение: Позволяют детально сканировать геометрию окружающего пространства, различая мелкие объекты.

Ограничения:

- Ограниченнная семантическая информация: Облако точек не содержит данных о цвете или текстуре, что затрудняет классификацию объектов без использования данных с камер.
- Чувствительность к погодным условиям: Туман, дождь, снег и пыль могут сильно рассеивать лазерный луч, что приводит к значительному ухудшению характеристик или полному отказу работы.
- Высокая стоимость: Лидары с высоким разрешением остаются одним из самых дорогих компонентов систем автономного транспорта.

1.4. Радары (Radio Detection and Ranging)

Принцип действия: Радар — активный сенсор, работающий в радиодиапазоне. Он излучает радиоволны и детектирует отраженный сигнал, что позволяет с высокой точностью определять расстояние до объекта и его скорость.

Преимущества:

- Устойчивость к погодным условиям: Радары надежно работают в условиях плохой видимости: туман, дождь, снег.
- Измерение скорости: Уникальная способность напрямую измерять скорость приближения или удаления объектов.
- Большая дальность действия: Способны обнаруживать объекты на расстоянии сотен метров.
- Относительно низкая стоимость: Современные технологии производства делают радары доступными.

Ограничения:

- Низкое угловое разрешение: Радары не могут предоставить детальную форму объекта, что затрудняет его точную классификацию.
- Низкая разрешающая способность: Плохо различают близко расположенные объекты.
- Чувствительность к помехам: Могут создавать взаимные помехи при работе нескольких радаров.

2. Практическая реализация системы стереоизрения

В качестве практической реализации одного из описанных подходов была разработана и протестирована программа на языке Python с использованием библиотеки OpenCV. Программа реализует полный цикл обработки данных со стереокамеры: калибровка, рекалибровка и построение карты глубины.

Аппаратная конфигурация: В качестве сенсора использовалась стереокамера GXIVISION-LS1M02 с базисом 85 мм и разрешением сенсоров 2 Мп. Обработка данных проводилась на компьютере с процессором AMD Ryzen 5 2600 и видеокартой NVIDIA GeForce GTX 1050 Ti.

Программный стек: Python 3.9, OpenCV 4.5, NumPy 2.3.2.

2.1. Калибровка стереосистемы

Калибровка выполнялась с использованием шахматной доски 9×6 с размером клетки 25 мм. Было захвачено 15 пар изображений под разными углами. На рис. 1 представлен пример одного из таких изображений.



Рис. 1. Пример калибровочного изображения

2.2. Ректификация изображений

На основе полученных параметров калибровки проводятся ректификационные преобразования для выравнивания эпиполярных линий.

Ректификационные преобразования — это процесс геометрического преобразования пары снимков одной сцены, целью которого является упрощение задачи поиска соответствий между ними. В основе лежит эпиполярная геометрия: для каждой точки на левом изображении соответствующая ей точка на правом лежит на определённой линии (эпиполярной линии). Ректификация трансформирует оба изображения таким образом, чтобы эти эпиполярные линии стали горизонтальными и совпали по строкам. В результате этого выравнивания поиск соответствующих точек сводится к сопоставлению пикселей в пределах одной строки, что значительно ускоряет и повышает надёжность работы алгоритмов стереоизрения, например, для построения карт глубины.

2.3. Построение карты глубины

Для построения карты глубины был выбран алгоритм SGBM (Semi-Global Block Matching). Его суть заключается в двухэтапной обработке: на первом этапе для каждого пикселя левого изображения осуществляется поиск наиболее похожего участка в правом изображении в пределах заданного смещения, что формирует первичную оценку глубины. На втором этапе для увеличения точности и устранения шумов, выполняется псевдоглобальная оптимизация: итоговое значение смещения для каждого пикселя вычисляется с учётом значений его соседей по множеству направлений. Это позволяет алгоритму учитывать гладкость реальных поверхностей, одновременно сохраняя чёткие границы объектов, что и обеспечивает высокую детализацию и помехоустойчивость итоговой карты.

При реализации системы стереоизрения была использована реализация данного алгоритма из библиотеки OpenCV.

На рис. 2 и рис. 3 представлены исходные изображения с левой и правой камер соответственно. На рис. 4 представлена полученная карта глубины.



Рис. 2. Исходное изображение с левой камеры



Рис. 3. Исходное изображение с правой камеры

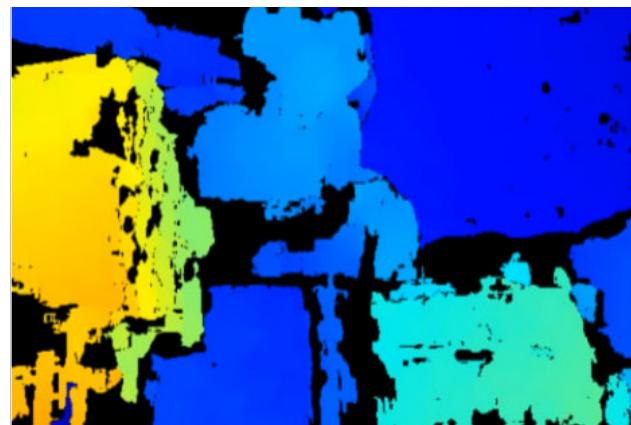


Рис. 4. Полученная карта глубины

Выявленные ограничения: Алгоритм SGBM показал значительное ухудшение качества в условиях низкой освещенности. На однородных поверхностях (стена, лист бумаги) наблюдаются протяженные «слепые зоны».

3. Сравнительный анализ технологий

Проведённый анализ ключевых аппаратных платформ компьютерного зрения позволяет систематизировать их характеристики по ряду наиболее важных для автономного транспорта критерии. Монокамеры и стереокамеры обеспечивают семантическое понимание сцены, но зависят от освещения. Результаты, полученные в практической части, свидетельствуют о целесообразности использования стереокамер в контролируемых условиях освещения при наличии текстурных поверхностей. Лидары предоставляют точные геометрические данные, но уязвимы к погоде, а также обладают крайне высокой стоимостью. Радары надежны для измерения дистанции и скорости в любую погоду, но обладают низким разрешением.

Сводные данные представлены в табл. 1.

Таблица 1

Сравнение технологий

Критерий / Технология	Монокамеры	Стереокамеры	Лидары	Радары
1	2	3	4	5
Точность глубины	Низкая	Средняя	Высокая	Средняя
Семантика	Высокая	Высокая	Низкая	Низкая
Работа ночью	Плохая	Плохая	Хорошая	Хорошая

1	2	3	4	5
Устойчивость к погоде	Низкая	Низкая	Низкая	Высокая
Стоимость	Низкая	Средняя	Высокая	Средняя
Вычислительная нагрузка	Высокая	Высокая	Средняя	Низкая

Заключение

В ходе проведённого исследования был выполнен сравнительный анализ ключевых аппаратных платформ компьютерного зрения для систем автономного транспорта. Установлено, что ни одна из технологий не является универсальным решением. Каждая обладает набором преимуществ и ограничений, что обуславливает тенденцию к созданию мультимодальных сенсорных систем, в которых комбинируются данные от разных сенсоров.

Литература

1. Анирад К. Искусственный интеллект и компьютерное зрение. Реальные проекты на Python, Keras и TensorFlow. – СПб. : Питер, 2023 – 624 с.
2. Хабр: официальный сайт. URL: <https://habr.com/ru/articles/130300/> (дата обращения: 09.10.2025)
3. OpencvSGBM class reference. URL: https://docs.opencv.org/3.4/d2/d85/classcv_1_1StereoSGBM.html (дата обращения: 12.10.2025)

РАСПОЗНАВАНИЕ ЛИЦ С ПОМОЩЬЮ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ

П. С. Барков

Воронежский государственный университет

Аннотация. Целью данной работы является практическое исследование процесса создания системы распознавания лиц на основе сверточных нейронных сетей. В статье детально рассматривается полный цикл построения модели — от подготовки данных до обучения и тестирования. Основное внимание уделено реализации компактной архитектуры CNN и применению функции потерь Triplet Loss для генерации эмбеддингов. Представлен рабочий пример на Python, включающий построение модели, процесс обучения и оценку качества в различных тестовых сценариях. Практическим результатом работы стала функционирующая модель продемонстрировавшая способность к обобщению на новых данных.

Ключевые слова: векторное представление (эмбеддинг), сверточные нейронные сети, распознавание лиц, функции потерь, метрики расстояния, Triplet Loss, машинное обучение, архитектура, нормализация, оптимизация.

Введение

Системы распознавания лиц активно развиваются и внедряются в различные сферы. В области безопасности они обеспечивают контроль доступа на различные объекты через системы видеонаблюдения. Банки активно внедряют оплату с помощью распознавания человека по лицу. В большинстве смартфонов доступна разблокировка через Face ID и аналогичные системы. Массовое распространение данной технологии легко объяснить ее удобством и простотой интеграции.

Для того чтобы сравнить несколько лиц их необходимо представить в числовом виде, отражающем значимые признаки внешности человека. Таким представлением являются эмбеддинги. Каждое изображение преобразуется в массив чисел, где комбинация этих значений отражает уникальные параметры лица человека.

Для сравнения полученных векторных представлений используются специальные метрики расстояния. Косинусное сходство, измеряющее насколько пара не нулевых векторов ориентирована в одном направлении и евклидово расстояние, оценивающее прямолинейную близость между ними. Лица на изображениях признаются одинаковыми, когда сходство между их эмбеддингами превышает пороговое значение.

Основным современным подходом для генерации векторных представлений являются сверточные нейронные сети, которые демонстрируют высокую эффективность в обработке изображений. Ключевым элементом архитектуры CNN является операция свертки. Ее смысл заключается в применении фильтров (ядер) к различным участкам изображения. Перемещая фильтр по сетке необходимо проводить операции поэлементного умножения соответствующих значений фильтра и текущей области и последующего суммирования результатов для получения карты признаков.

Данная статья рассматривает весь цикл работы системы распознавания и сопровождает каждый шаг практическими примерами.

1. Подготовка данных

Перед непосредственной реализацией алгоритмов машинного обучения и построением архитектуры нейронной сети необходимо сформировать качественную обучающую выборку.

В ходе работы стоит учитывать, что способность системы к обобщению и адаптации в реальных условиях эксплуатации определяется разнообразием обучающих данных. Для достижения хороших результатов необходимо включать в выборку лица с различными характеристиками:

- разный возраст, пол и этническая принадлежность;
- разнообразные эмоциональные выражения;
- различные ракурсы и повороты головы;
- изображения, сделанные в разных условиях освещения.

Для демонстрации был выбран датасет в котором содержится множество фотографий ста восьмидесяти пяти известных людей. Хоть для получения достаточно качественной модели требуются миллионы различных изображений, но ввиду ограниченности вычислительных ресурсов для оптимизации времени обучения на каждого человека было выбрано по пятьдесят изображений.

2. Архитектура

Архитектура сверточной нейронной сети является главным фактором в процессе извлечения признаков из изображений лиц. От выбора архитектуры напрямую зависят ключевые характеристики системы: точность распознавания, вычислительная эффективность, устойчивость к изменениям условий съемки и способность к обобщению.

Основная задача архитектуры заключается в последовательном преобразовании входного изображения в компактное векторное представление, которое сохраняет уникальные характеристики лица и инвариантно к несущественным изменениям. Каждый компонент архитектуры вносит свой вклад в этот процесс:

Сверточные слои выполняют основную работу по извлечению признаков, применяя фильтры к различным областям изображения. Размер ядра свертки определяет величину области, анализируемой для выделения признаков. Меньшие ядра позволяют выделять более локальные особенности, такие как края и текстуры, в то время как большие ядра охватывают обширные области помогая выделить общие черты.

Пуллинговые слои постепенно уменьшают пространственную размерность карт признаков, сохраняя при этом наиболее значимую информацию. Одним из популярных подходов является MaxPooling, который выбирает максимальное значение в области. Это увеличивает устойчивость признаков к малым смещениям и искажениям.

Нормализация стабилизирует процесс обучения, уменьшая внутреннее ковариационное смещение. Это позволяет увеличить темпы обучения и улучшить обобщающую способность модели.

Для демонстрации принципов работы системы распознавания лиц была разработана компактная нейронная сеть, реализующая базовые подходы к извлечению признаков.

Сеть состоит из четырех последовательных сверточных блоков, каждый из которых включает сверточный слой с ядром 3×3 , операцию батч-нормализации, активацию ReLU и слой MaxPooling. Количество фильтров последовательно увеличивается от 32 до 256, что позволяет сети извлекать признаки разного уровня абстракции.

Ключевой особенностью архитектуры является использование глобального усредняющего пулинга на последнем этапе обработки. Этот подход заменяет традиционные полносвязные слои. Этот подход значительно сокращает количество параметров модели и снижает риск переобучения. Финальный этап включает линейное преобразование в эмбеддинг размерностью 128 элементов с L2-нормализацией.

3. Функция потерь

Функции потерь представляют собой математические конструкции, которые количественно оценивают расхождение между предсказанными и эталонными значениями. В контексте машинного обучения они выполняют две основные функции: служат целевым критерием для оптимизации параметров сети и определяют направление градиентного спуска в процессе обучения.

В современных системах распознавания лиц существует несколько видов функций потерь, каждая из которых имеет уникальный подход к оптимизации пространства признаков. Одной из самых известных является ArcFace, направленная на повышение дискриминативной способности изученных признаков за счёт добавления углового интервала между различными классами.

Большинство функций демонстрируют превосходную производительность на крупномасштабных наборах данных. Однако они требуют значительных вычислительных ресурсов и сложны в реализации, что делает их не подходящими для небольших проектов.

Для обучения модели в примере использовалась такая функция как Triplet Loss. Основной принцип работы которой заключается в сравнительном обучении на тройках изображений. На вход подаются три вида значений: якорь (anchor) исходное изображение, положительное (positive) фото того же человека и отрицательное (negative) изображение другого человека. Цель обучения модели заключается в минимизации расстояния между anchor и positive с одновременной максимизацией разницы с negative.

Triplet Loss обладает рядом преимуществ, среди которых можно отметить ее простоту и интуитивную понятность. Данный подход не требует введения дополнительных слоев в архитектуру нейронной сети, что сохраняет вычислительную эффективность модели. Функция демонстрирует хорошую эффективность даже при работе с ограниченными наборами данных. Кроме того, она напрямую оптимизирует метрики расстояния в пространстве эмбеддингов, обеспечивая целенаправленное обучение.

Однако ключевым ограничением данной функции является ее зависимость от качества обучающих троек. Если модель обучается преимущественно на простых примерах, где негативные образцы очевидно отличаются от якорных изображений, система не приобретает способность различать лица со схожими чертами. Это приводит к тому, что в реальных условиях, когда встречаются визуально похожие люди разных классов, модель может демонстрировать низкую точность распознавания.

4. Обучение модели

Процесс обучения нейронной сети представляет собой итеративную оптимизацию, в ходе которой модель постепенно настраивает свои внутренние параметры для минимизации функции потерь. Это означает, что сеть учится преобразовывать пиксели изображения в компактные векторные представления, отражающие уникальные характеристики каждого человека. Главную роль в этом процессе играют параметры оптимизации, каждый из которых оказывает влияние на конечный результат.

Learning rate определяет величину шага, с которой модель корректирует свои веса на каждой итерации. Слишком высокое значение этого параметра (0.01 и выше) приводит к нестабильности обучения и невозможности сходжения к оптимуму. В то время как слишком низкое (0.0001 и ниже) значительно замедляет процесс обучения. В данной работе использовался адаптивный алгоритм Adam с начальным learning rate 0.001. Данный алгоритм автоматически регулирует величину шага для каждого параметра индивидуально. Это особенно важно для глубоких сетей, где градиенты могут существенно различаться по величине в разных слоях.

Размер батча напрямую влияет на стабильность и скорость обучения. Малые батчи (16–32 образца) обеспечивают более частые обновления весов и лучше обобщение, но могут приводить к шумным градиентам. Большие батчи (64–128) дают более точную оценку градиента, но требуют больше памяти и могут застревать в локальных минимумах. Для примера был выбран размера батча 32, который представляет собой компромисс между оптимизацией и точностью.

Количество эпох обучения должно быть достаточным для сходимости модели, но не чрезмерным, чтобы избежать переобучения. Мониторинг функции потерь на валидационной выборке позволяет определить момент, когда дальнейшее обучение перестает улучшать качество модели. А использование ранней остановки (early stopping) при отсутствии улучшений в течение нескольких эпох позволяет предотвратить переобучения и сэкономить время. Для тестового примера число эпох было установлено на 30. Это значение позволяет завершить обучение в разумные сроки, сохраняя при этом достаточное количество итераций для адаптации весов модели к особенностям данных.

5. Тестирование

Для оценки работоспособности обученной модели было проведено три типа тестов, демонстрирующих различные сценарии использования системы распознавания лиц. Для сравнения эмбеддингов в тестах использовалось косинусное сходство. Пороговое значение для принятия решения о принадлежности лиц одному человеку было установлено на уровне 0.6.

5.1. Верификация известного человека

В первом тесте использовались две фотографии одного человека из исходного датасета (рис. 1, 2), однако сами изображения не входили в тренировочную выборку. Модель успешно распознала принадлежность лиц одному человеку, показав значение сходства 0.826. Это демонстрирует способность модели к обобщению и работе с новыми изображениями известных ей людей.



Рис. 1. Фото известного человека №1



Рис. 2. Фото известного человека №2

5.2. Сравнение разных людей

Второй тест проверял способность модели отличать двух разных людей (рис. 3, 4), не участвовавших в процессе обучения модели. Система отработала, корректно указав их как разных людей. Значение косинусного сходства 0.4077, что значительно ниже установленного порога в 0.6. Это показывает способность модели к работе с новыми лицами.



Рис. 3. Фото случайного человека №1



Рис. 4. Фото случайного человека №2

5.3. Работа с неизвестным человеком

Третий тест, проверяет работу системы с двумя изображениями одного человека, отсутствующего в тренировочном наборе (рис. 5, 6). Модель продемонстрировала уровень сходства 0.58, что ниже установленного порога классификации. Это означает, что модель не смогла однозначно определить, принадлежат ли оба изображения одному человеку или нет.



Рис. 5. Фото человека, отсутствующее в наборе №1



Рис. 6. Фото человека, отсутствующее в наборе №2

Заключение

Проведенное исследование продемонстрировало работоспособность подхода к распознаванию лиц на основе сверточных нейронных сетей. В ходе работы была успешно создана небольшая модель, способная преобразовывать изображения лиц в векторные представления. Модель продемонстрировала способность к обобщению, показав работоспособность не только на данных из обучающей выборки, но и на новых изображениях.

Однако полученная модель далека от идеала и имеет существенные ограничения. Точность ее решений значительно уступает промышленным системам распознавания лиц. Причиной этому послужил ограниченный объем тренировочных данных и упрощенная архитектура сети.

Несмотря на существующие ограничения, разработанная система показывает работоспособность концепции и составляет основу для дальнейших исследований в области распознавания лиц. Полученные результаты и выводы могут быть использованы как отправная

точка для создания более совершенных систем с улучшенными характеристиками точности и надежности.

Литература

1. *Аггарвал Ч.* Нейронные сети и глубокое обучение: учебный курс / Ч. Аггарвал. – пер. с англ. А. Г. Гузикевича – Санкт-Петербург : Диалектика, 2020. – 752 с.
2. *Гудфеллоу Я.* Глубокое обучение / Я. Гудфеллоу, И. Бенджио, А. Курвилль. – 2-е изд., испр. пер. с англ. А. А. Слинкина. – Москва : ДМК Пресс, 2018. – 652 с.
3. *Жерон О.* Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем / О. Жерон – пер. с англ. и ред. Ю. Н. Артеменко. – Санкт-Петербург : Альфа-книга, 2018. – 688 с.

РАЗРАБОТКА ГИБРИДНОГО ИНДИКАТОРА ПРОГНОЗИРОВАНИЯ ДИНАМИКИ АКЦИЙ НА ОСНОВЕ ФРАКТАЛЬНОГО АНАЛИЗА И МАШИННОГО ОБУЧЕНИЯ

С. А. Беджанян, Е. М. Аристова

Воронежский государственный университет

Аннотация. Настоящая работа посвящена разработке интеллектуального индикатора, предназначенного для прогнозирования направлений движения цен акций на фондовом рынке. Индикатор использует гибридный подход, объединяющий методы фрактального анализа (индекс Херста, фрактальная размерность) и ансамблевые алгоритмы машинного обучения (Random Forest / XGBoost). В отличие от классических технических индикаторов (SMA, RSI, MACD), гибридная модель способна учитывать нелинейный и хаотичный характер финансовых временных рядов. Программный продукт реализован на языке Python с использованием библиотек pandas и scikit-learn. Тестирование на исторических данных (2020–2025 гг.) показало, что разработанный индикатор обеспечивает точность прогноза в диапазоне 65–75 % и демонстрирует положительное превышение доходности рассматриваемой в статье стратегии над стратегией «Buy and Hold» по показателю коэффициента Шарпа.

Ключевые слова: фондовый рынок, временной ряд, фрактальный анализ, индекс Херста, фрактальная размерность, машинное обучение, Random Forest, XGBoost, волатильность, технический индикатор.

Введение

Современный фондовый рынок является динамичной и сложной системой, характеризующейся высокой волатильностью и нелинейными зависимостями, что снижает эффективность традиционных методов анализа. Активное развитие вычислительных технологий открывает возможности для создания новых интеллектуальных инструментов, способных выявлять скрытые закономерности в финансовых данных. Особый интерес представляет сочетание фрактального анализа и алгоритмов машинного обучения для создания индикаторов, учитывающих нелинейный характер ценовых рядов [2]. Цель работы — разработка и исследование гибридного индикатора прогнозирования направлений движения акций, основанного на интеграции фрактальных характеристик и ансамблевых методов машинного обучения.

1. Математическая основа гибридного индикатора

Архитектура разработанного индикатора основана на представлении временного ряда доходностей как стохастического процесса [3] и использует два ключевых компонента: блок фрактального анализа для извлечения признаков и блок машинного обучения для прогнозирования.

1.1. Фрактальные признаки и логика построения

Для количественного описания структуры временного ряда использовались фрактальные показатели, отражающие наличие долгосрочной памяти и степень трендовости рынка [2]. Ключевым параметром является **Индекс Херста** (H), который рассчитывается для скользящего окна данных (30 дней) и интерпретируется следующим образом:

- $H \in (0, 0.5)$ — анти-персистентный процесс, цены склонны возвращаться к среднему;
- $H = 0.5$ — случайное блуждание (эффективный рынок);
- $H \in (0.5, 1)$ — персистентный процесс, наличие устойчивого тренда.

Дополнительно рассчитывается фрактальная размерность (FD), связанная с индексом Херста соотношением $FD = 2 - H$, и отражающая степень «шероховатости» графика цены. Чем выше FD , тем хаотичнее движение цен.

В качестве функции прогнозирования f рассматривается модель индикатора, которая имеет вид:

$$y_{t+1} = f(X_t, H_t, FD_t),$$

где f — функция прогнозирования;

t — промежуток времени;

X_t — технические индикаторы в момент t ;

H_t — индекс Херста в момент t ;

FD_t — фрактальная размерность в момент t .

1.2. Выбор и настройка модели машинного обучения

Для реализации функции f были выбраны ансамблевые методы: **Random Forest** (RF) и **XGBoost**. Модель RF была выбрана в качестве базовой благодаря её устойчивости к шуму, низкой склонности к переобучению и возможности оценки значимости признаков [4]. Оптимизация гиперпараметров (количество деревьев, максимальная глубина) проводилась с использованием кросс-валидации и процедуры грид-поиска по метрике F1-Score.

2. Результаты тестирования и сравнение стратегий

Тестирование индикатора проводилось на исторических данных дневных котировок акций (AAPL, MSFT) за период 2020–2025 гг. Оценка эффективности включала классификационные и финансовые метрики. Результаты сравнения модели Random Forest с классическими индикаторами представлены в табл.1 (для иллюстрации итогов):

Таблица 1

Сравнение эффективности гибридного индикатора с классическими инструментами (акции AAPL)

Индикатор	Метрика	Accuracy (%)	F1-Score	Коэффициент Шарпа
Buy & Hold	—	—	—	~0.75
SMA (50)	53.2	0.51	0.81	
MACD	55.1	0.55	0.95	
Гибридный (RF + Фракталы)	68.5	0.69	1.15	

Как видно из табл. 1, гибридный индикатор, использующий фрактальные и лаговые признаки, демонстрирует значительное превосходство над классическими инструментами по всем ключевым метрикам. Точность прогноза (Accuracy) стабильно находится в диапазоне 65–75 %. Ключевым финансовым результатом является более высокий коэффициент Шарпа (Sharpe Ratio), что свидетельствует о получении более высокой доходности с поправкой на риск. Важно отметить, что включение фрактальных признаков позволяет модели лучше адаптироваться к изменению рыночных режимов (от тренда к флэту — цены склонны к случайному блужданию — они возвращаются к среднему), поскольку индекс Херста количественно оценивает долгосрочную память системы.

Заключение

В результате работы был разработан и протестирован гибридный индикатор прогнозирования направлений движения акций на основе сочетания фрактального анализа и методов машинного обучения. Проведённое тестирование на исторических данных подтвердило, что интеграция фрактальных характеристик в модель Random Forest позволяет достичь высокой точности прогноза (до 75 % на тестовой выборке) и значительно повысить финансовую эффективность торговой стратегии по сравнению с пассивным инвестированием и стратегиями, основанными на классических технических индикаторах. Практическая значимость работы заключается в возможности применения разработанного индикатора для поддержки инвестиционных решений, автоматизации торговых систем и дальнейших исследований в области финансовой эконометрики [1, 3].

Литература

1. Фама Ю. Эффективность финансовых рынков: теория и практика. – Москва : Альпина Паблишер, 2017. – 416 с.
2. Мандельброт Б. Фракталы и финансовые рынки: риск, случайность и прибыль. – Москва : МЦНМО, 2002. – 264 с.
3. Энгл Р. Авторегрессионные модели с условной гетероскедастичностью (ARCH) для дисперсий временных рядов / Р. Энгл, Т. Боллерслев // Журнал эконометрики. – 1982. – Т. 50, № 6. – С. 987–1007.
4. Брекер Г. И. Применение случайного леса для прогнозирования цен на акции // Информационные технологии. – 2021. – № 3. – С. 15–20.

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ АВТОМАТИЧЕСКОЙ ФИЛЬТРАЦИИ РЕЧИ В ЯЗЫКАХ С РАЗЛИЧНОЙ СТРУКТУРОЙ

Л. В. Бобрышева, С. Ю. Болотова

Воронежский государственный университет

Аннотация. В статье рассматриваются методы фильтрации текста после автоматического распознавания речи (ASR) с учётом морфологической структуры различных языков. Анализируются словарные методы, регулярные выражения, морфемный разбор, лемматизация и контекстные модели, оцениваются их преимущества и ограничения. Приведены примеры для агглютинативных, флексивных и аналитических языков. Показано, что комбинирование методов в пайплайнах повышает точность фильтрации и компенсирует ограничения отдельных подходов.

Ключевые слова: автоматическое распознавание речи, ASR, фильтрация текста, морфологическая вариативность, словарные методы, регулярные выражения, морфемный разбор, лемматизация, контекстные модели, ошибки распознавания, накопительные эффекты ошибок, автокоррекция, агглютинативные языки, флексивные языки, аналитические языки, пайплайны фильтрации.

Введение

Автоматическая фильтрация текста после распознавания речи (ASR) является важной задачей для мультиязычных систем обработки аудио- и видеоконтента, включая модерацию материалов, автоматическое создание субтитров и поиск ключевых слов. Эффективность фильтрации зависит от морфологической структуры языка и ошибок распознавания, таких как пропуски слов, фонетические и орфографические искажения. Во флексивных языках, как русский, множество словоформ снижает эффективность простых словарей; в агглютинативных языках, например финском, длинные аффиксальные цепочки требуют морфемного анализа; в аналитических языках, таких как английский, формы слов относительно стабильны, что позволяет использовать регулярные выражения и списки ключевых слов.

Цель работы — сопоставить методы фильтрации текста после ASR для разных типов языков и показать ограничения каждого подхода. В качестве примеров рассматриваются русский, финский и английский языки.

1. Классификация языков по трудностям фильтрации

1.1. Влияние морфологической структуры и особенностей слова на эффективность фильтрации

Каждый язык обладает своей уникальной структурой и особенностями словообразования, что определяет, как формируются слова и выражаются грамматические значения. Для систематизации языков и оценки сложности их автоматической обработки часто выделяют три условные группы: агглютинативные, флексивные и аналитические языки [1].

Агглютинативные языки формируют слова путем последовательного присоединения аффиксов к корню — частям слова, которые добавляют грамматическое или смысловое значение, например суффиксы. При этом слова могут образовывать практически бесконечные составные конструкции. Примерами таких языков являются финский, турецкий, венгерский или японский. Для автоматической фильтрации отдельное слово может встречаться в разных формах

внутри сложных слов. Простое совпадение последовательности символов часто оказывается недостаточным, так как комбинация аффиксов может случайно содержать нужную последовательность. Важно ориентироваться на корень и аффиксы, учитывая, что корень сам может подвергаться фонетическим изменениям, например чередованию согласных в финском. Поэтому фильтрация требует морфемного анализа и учета фонетических правил, что значительно усложняет алгоритмы, особенно при обработке текста после ASR, где могут присутствовать ошибки и нестандартные формы слов.

Флективные языки, к которым относится русский, характеризуются изменяемыми словоформами. Словоформа — это конкретная грамматическая форма слова, которая появляется в речи или тексте и несёт информацию о падеже, числе, роде, времени, лице или наклонении [2]. Например, существительное «стол» имеет словоформы «стол», «стола», «столу», «столом», «столе», а глагол «идти» — «иду», «идёшь», «идут», «шёл», «пошёл».

Словоформу следует отличать от леммы — базовой словарной формой слова, к которой относятся все её словоформы. Для приведённых примеров леммами будут «стол» и «идти». Это различие открывает возможность применения специализированных методов анализа словоформ, что будет рассмотрено далее [1, 3, 4].

В отличие от агглютинативных языков, где слова строятся из относительно стабильного корня и аффиксов, в флективных языках изменения словоформ могут сильно менять внешний вид слова. В результате одна и та же лексическая единица может принимать формы, которые почти не похожи друг на друга.

Аналитические языки, такие как английский или китайский, используют преимущественно неизменяемые слова, а грамматические отношения выражаются через порядок слов, служебные частицы или предлоги. Формы слов относительно стабильны, за редкими исключениями, что упрощает фильтрацию: для большинства случаев достаточно знать исходное слово и искать его в тексте. Например, глагол walk может встречаться как walks, walked, walking, что легко учесть с помощью регулярных выражений. Возможные сложности возникают с идиомами, сленгом или производными формами, которые требуют дополнительного внимания при поиске целевых слов. Тем не менее, базовые методы фильтрации, такие как словари и регулярные выражения, остаются эффективными для выявления ключевых слов и выражений.

Важно понимать, что деление языков на группы не строгое: реальные языки могут сочетать черты разных типов. Это проявляется, например, в научной или технической терминологии, заимствованных словах и взаимодействии с различными областями языка. Иногда флективный язык использует элементы агглютинативного типа в сложносоставных терминах, а аналитический язык может включать морфологические модификации или составные формы в специализированной лексике. Понимание таких особенностей необходимо для выбора адекватных методов фильтрации, оценки их эффективности и минимизации ошибок при обработке текста или аудио.

Такой подход позволяет системно оценить, насколько сложно или просто выявлять отдельные слова для фильтрации, и служит основой для последующего анализа методов автоматической обработки речи с учётом морфологической специфики каждого языка.

1.2. Примеры влияния морфологической структуры на фильтрацию речи

На примере агглютинативного языка, финского, далее будут рассмотрены слова *jalka* («нога») и *sana* («слово»). В финском слова строятся из корня и последовательных аффиксов, которые добавляют грамматическое или смысловое значение. В слове *jalka* формы *jal-*, *alan*, *jalkaniin*, *jalkojani* показывают, что корень *jal-* остаётся узнаваемым, и на первый взгляд кажется, что для фильтрации можно просто опираться на корень: отделив аффиксы с конца, можно получить основу для дальнейшей работы.

Однако на практике структура языка и фонетические изменения усложняют задачу фильтрации. Для примера можно рассмотреть слово *sana* («слово») и его производные формы. Простые производные, такие как *sanan* (родительный падеж), *sanoja* (множественное число) и *sanat* (множественное число, именительный падеж), сохраняют смысл исходного слова, и их корректная идентификация требует морфемного анализа или лемматизации.

Сложности появляются с составными словами, например *sanakirja* («словарь»), где корень *sana* встречается вместе с корнем *kirja* («книга»). В этом случае графическое совпадение с корнем не гарантирует, что значение слова соответствует исходному «слово». Простое выделение корня может привести к ложным срабатываниям, если цель фильтрации — именно выделение слова *sana*.

Ещё более сложным случаем являются семантические сдвиги, например *sanomat* («сообщения, новости»), где слово связано с корнем *sana*, но смысловое значение отличается от исходного. Морфемный анализ выявляет корень, но без учета контекста алгоритм не сможет различить «слово» и «новости».

Таким образом, эффективная фильтрация требует сочетания подходов. Морфемный анализ и лемматизация позволяют корректно идентифицировать словоформы и простые производные, но для различия составных слов и семантически смешённых форм необходимо использовать модели, учитывающие контекст, включая нейросетевые алгоритмы обработки текста. Кроме того, следует учитывать фонетические изменения корня, например чередования согласных, которые встречаются при образовании различных форм. Такой комплексный подход обеспечивает точную фильтрацию, минимизируя ложные срабатывания и учитывая графическую, морфологическую и семантическую специфику финского языка.

Флективные языки, такие как русский, характеризуются значительной изменяемостью словоформ. Каждое слово может иметь десятки и сотни форм в зависимости от падежа, числа, рода, времени, лица и наклонения. При этом некоторые формы могут сильно отличаться от исходной леммы: например, глагол «идти» проявляется как «иду», «идёшь», «шёл», «пошёл», где формы «иду» и «пошёл» почти не связаны визуально с леммой [5].

Такая вариативность делает невозможным использование простых словарей или совпадений по корню: для эффективной фильтрации необходимо заранее составить словарь всех словоформ с привязкой к лемме или использовать морфологический анализ. В процессе работы с ASR-транскрипциями возникает дополнительная сложность: ошибки распознавания и нестандартные варианты написания могут ещё сильнее удалять словоформу от её леммы или приводить к заведомо неверной форме, например из «иду» получаем «еду».

Таким образом, в флективных языках задача фильтрации требует комбинированного подхода: морфологического разбора, сопоставления с леммой и учёта всех возможных словоформ. В отличие от агглютинативных языков, где достаточно выделить корень слова, в флективных языках простое выделение корня не обеспечивает корректной фильтрации; алгоритм должен быть готов обрабатывать десятки словоформ для одной леммы, чтобы избежать ложных срабатываний и пропусков целевых слов.

В аналитических и фиксированных языках, таких как английский, слова в основном неизменяемы, а грамматические категории выражаются отдельными служебными словами или фиксированными формами. Например, глагол *walk* может принимать лишь ограниченное число форм: *walk*, *walks*, *walked*, *walking*. Фильтрация таких слов может выполняться с помощью точных списков слов и регулярных выражений, например *walk(s|ed|ing)?*, что позволяет выявлять все формы глагола. Основная сложность возникает с многословными конструкциями и идиомами, смысл которых не выводится из отдельных слов, например *kick the bucket* — «умереть». Для корректной фильтрации такие выражения нужно учитывать целиком, а не по отдельным словам, что также можно реализовать через регулярные выражения, охватывающие стандартные грамматические формы.

2. Методы фильтрации текста после ASR

2.1. Методы фильтрации: преимущества и ограничения

Фильтрация текста после автоматического распознавания речи опирается на набор методов, направленных на выявление целевых слов, снижение ложных срабатываний и компенсацию ошибок ASR. Применимость каждого метода сильно зависит от типа языка и степени морфологической изменчивости.

Черные и белые списки слов [6] основываются на поиске точных совпадений с заранее заданным набором словоформ. Такой подход прост, стабилен и хорошо работает в языках с ограниченной морфологией или малой вариативностью форм. Его основной недостаток — строгая зависимость от точной орфографии: любое отклонение, вызванное словоформой или ошибкой ASR, приводит к пропуску. В агглютинативных языках возможности метода дополнительно ограничены огромным числом потенциальных форм, которые невозможно перечислить вручную.

Регулярные выражения [7] позволяют задавать шаблоны, включающие окончания, приставки или устойчивые конструкции. Они особенно эффективны в аналитических языках, где количество словоформ невелико и может быть описано внутри одного выражения. В языках с богатой морфологией этот подход быстро упирается в масштаб: десятки возможных сочетаний аффиксов делают выражения чрезмерно громоздкими и неполными. Ошибки ASR — пропуски, слияние слов, подстановки — дополнительно разрушают ожидаемые паттерны, снижая надёжность метода.

Морфемный разбор и лемматизация представляют собой отдельные, но тесно связанные методы. Морфемный разбор выделяет корень и аффиксы, а лемматизация приводит словоформу к словарной основе. Эти методы часто используются совместно со словарями лемм. Такой подход наиболее эффективен в флексивных языках, где одна лемма имеет множество форм, а поиск по точному совпадению не работает. Основное ограничение обоих методов — зависимость от полноценной морфологической базы и корректной орфографии: искажённые ASR-формы нередко оказываются нераспознаваемыми или приводят к неверной лемме.

Контекстные модели, включая нейросетевые, анализируют окружение слова и позволяют различать омонимии, производные формы и конструкции, чья семантика не определяется по отдельным частям. Этот подход особенно важен при работе с производными словами, идиомами и ошибками ASR, которые формально похожи на целевые единицы, но имеют другое значение. Ограничения связаны с вычислительной стоимостью, зависимостью от обучающих данных и возможными ошибками в случаях редких или нестандартных искажений.

Эффективная фильтрация текста после ASR требует комбинирования методов: словари подходят для строго фиксированных форм, регулярные выражения — для аналитических структур, морфологический анализ — для флексивных систем, а контекстные модели — для сложных или неоднозначных случаев. Совместное использование подходов позволяет компенсировать их ограничения и обеспечивать устойчивую работу фильтрации в условиях ошибок распознавания и морфологической вариативности [8].

2.2. Ошибки ASR и их влияние на фильтрацию текста

Словарные методы чувствительны к любым отклонениям от ожидаемой формы слова. Пропуск буквы или слияние слов приводит к тому, что точное совпадение не находится, и целевое слово остаётся незамеченным. Для флексивных языков, где словоформы уже варьируются естественным образом, ошибки ASR усугубляют проблему, что увеличивает количество «непоставимых» форм и требует расширенных словарей или дополнительных методов коррекции.

Регулярные выражения, хотя и обеспечивают более гибкие шаблоны поиска, также подвержены влиянию ошибок распознавания. Непредсказуемые вставки, пропуски или замены символов могут разрушить шаблон и сделать паттерн неприменимым. Особенно уязвимы конструкции, основанные на предсказуемых окончаниях или устойчивых сочетаниях: малейшее отклонение от ожидаемой формы делает поиск невозможным и повышает риск пропуска целевых слов.

Морфологический разбор и лемматизация частично снижают влияние ошибок ASR: алгоритм может привести словоформу к базовой форме, нивелируя часть искажений. Однако при значительных искажениях или редких ошибках, которых нет в морфологической базе, корректное определение леммы становится невозможным. Это особенно критично для флексивных языков, где одна лемма может иметь десятки форм, а каждая ошибка ASR создаёт новую, потенциально «неизвестную» форму, что не упрощает, а иногда даже усложняет работу алгоритма.

Контекстные модели (нейросетевые) способны частично компенсировать ошибки ASR, определяя слово по окружению и смыслу. Они эффективны для распознавания омонимов, устойчивых выражений и производных форм, которые иначе могли бы быть пропущены. Основное ограничение таких моделей — необходимость большого обучающего корпуса с типичными ошибками распознавания и высокая вычислительная стоимость. Чем меньше ошибок в исходной транскрипции, тем сложнее создать обучение, покрывающее редкие искажения.

Следует отметить накопительные эффекты ошибок: не все ошибки становятся явными сразу. Например, неверная словоформа может привести к неправильной лемме или морфемной разбивке, что на следующем этапе обработки создаёт новые ошибки. Так, слово «иду», распознанное как «еду», может привести к неверной классификации смысла или семантики, создавая каскад ложных срабатываний и пропусков.

Важным шагом для снижения влияния ошибок ASR является автокоррекция. Она не решает задачу фильтрации напрямую, но исправляет часть орфографических искажений, что делает работу словарей, регулярных выражений и морфологического разбора более надёжной. Несмотря на возможные ошибки подбора форм, метод практически единственный, который напрямую уменьшает негативное влияние ошибок ASR на последующую фильтрацию.

Таким образом, ошибки ASR являются ключевым фактором, ограничивающим точность фильтрации текста. Любой метод, хорошо работающий на идеальном материале, теряет эффективность при наличии транскрипционных искажений. Понимание природы этих ошибок и их накопительного эффекта позволяет выстраивать стратегии компенсации и подготавливает почву для построения комплексных пайплайнов фильтрации, которые будут рассмотрены в следующем разделе.

2.3. Комбинирование методов и построение пайплайнов

Для эффективной фильтрации текста после ASR отдельные методы редко работают в изоляции, особенно в условиях ошибок распознавания и морфологической вариативности языков. На практике создаются пайплайны — последовательности методов, где каждый шаг компенсирует ограничения предыдущего.

Типичная схема включает несколько уровней обработки. Сначала применяется автокоррекция, которая исправляет часть орфографических искажений, снижая влияние ошибок ASR на последующие методы. Затем, в зависимости от морфологического типа языка (агглютивного, флексивного, аналитического), словарные проверки и регулярные выражения ищут целевые слова и устойчивые конструкции, используя исправленный текст. Морфологический разбор и лемматизация позволяют учитывать все словоформы и корректно сопоставлять их с леммами. Завершающий уровень могут составлять контекстные модели, которые различают

значения производных форм и омонимов, а также учитывают смысловую зависимость слов в предложении, исключая возможные ошибки морфемных и словарных моделей.

Комбинирование методов уменьшает накопительные эффекты ошибок, когда неверная распознанная словоформа может привести к ошибкам на следующих этапах обработки. Выстраивание пайплайна с учётом особенностей языка и предполагаемых ошибок ASR обеспечивает более устойчивую фильтрацию текста и позволяет адаптировать систему под различные группы языков и морфологические типы. При этом для каждого языка пайплайн приходится настраивать индивидуально: универсального решения не существует, и оптимальная схема создаётся из набора частных решений с учётом конкретных особенностей языка.

Заключение

Фильтрация текста после автоматического распознавания речи является комплексной задачей, где эффективность каждого метода зависит от морфологической структуры языка и характера ошибок ASR. Простые подходы, такие как словари или регулярные выражения, хорошо работают на ограниченных формах, но быстро теряют точность при изменяемых или сложных словоформах. Морфологический разбор и лемматизация позволяют учитывать вариативность слов, а контекстные модели помогают различать омонимы и производные формы, учитывая смысловую зависимость слов в предложении.

Ошибки ASR — пропуски, искажения, слияния слов — существенно снижают эффективность всех методов, создавая накопительные эффекты и требуя специальных решений для компенсации. Автокоррекция выступает важным инструментом, который исправляет часть орфографических искажений и повышает надёжность последующей фильтрации.

На практике успешная фильтрация достигается через комбинирование методов в пайплайнах, где каждый шаг компенсирует ограничения предыдущего и учитывает особенности конкретного языка. Универсального решения не существует: эффективные системы строятся из набора частных подходов, адаптированных под морфологию и характер ошибок ASR. Такой комплексный подход обеспечивает устойчивость фильтрации, минимизирует ложные срабатывания и повышает точность выявления целевых слов и выражений в транскрибированном материале.

Литература

1. Полицына Е. В. Анализ качества работы и расширение возможностей инструментов морфологического анализа текстов на русском языке / Е. В. Полицына, С. А. Полицын, А. С. Поречный, А. Н. Рыкунов // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2023. – № 2. – С. 171–180. – DOI: 10.17308/sait/1995-5499/2023/2/171-180.
2. Унгбаева О. Морфологический анализ в русском языке / О. Унгбаева // Journal of New Century Innovations. – 2025. – Т. 77, № 1. – С. 14–16.
3. Джураев Ш. Т. Новые подходы морфологического анализа текста / Ш. Т. Джураев // PEDAGOGS. – 2025. – № 92.
4. Кузменко Е. Morphological Analysis for Russian: Integration and Comparison of Taggers / Е. Кузменко. – Национальный исследовательский университет «Высшая школа экономики», 2023. – 12 с. – URL: <https://publications.hse.ru/pubs/share/direct/185081516.pdf> (дата обращения: 23.11.2025).
5. Курьянова О. В. Морфология и линейный порядок атрибутивных прилагательных в истории русского языка / О. В. Курьянова // Rhema. – 2020. – № 2. – С. 76–91. – DOI: 10.31862/2500-2953-2020-2-76-91.

6. *Jurafsky D., Martin J.* Speech and Language Processing. – 3rd ed., draft. – 2023. – 1250 p. – URL: <https://web.stanford.edu/~jurafsky/slp3/> (дата обращения: 23.11.2025).

7. *Friedl J. E. F.* Mastering Regular Expressions. – 3rd ed. – Sebastopol: O'Reilly Media, 2006. – 612 p.

8. *Romanov V.* Evaluation of Morphological Embeddings for the Russian Language / V. Romanov, A. Khusainova. – arXiv, 2021. – URL: <https://arxiv.org/abs/2103.06628> (дата обращения: 23.11.2025).

ПРИМЕНЕНИЕ МОДЕЛЕЙ КОМПЬЮТЕРНОГО ЗРЕНИЯ ДЛЯ АНАЛИЗА ГУСТОТЫ СТОЯНИЯ СОИ В УСЛОВИЯХ АМУРСКОЙ ОБЛАСТИ

И. С. Буренкова, Н. Н. Максимова

Амурский государственный университет

Аннотация. В работе исследуется применение модели глубокого обучения на основе архитектуры YOLOv8n для детекции ростков сои на наземных изображениях, полученных в полевых условиях Амурской области. Особое внимание уделено влиянию методов аугментации данных на качество оценки густоты посевов, где критически важной метрикой является полнота (Recall), поскольку пропущенные растения напрямую иска-жают агрономическую оценку. Показано, что применение аугментации (включая геометрические и фотометрические преобразования) повышает Recall с 0.78 до 0.85, улучшает устойчивость модели к вариациям освещённости и визуального шума, а также снижает количество ложноотрицательных предсказаний. Полученные результаты подтверждают целесообразность использования аугментированных данных при обучении моделей компьютерного зрения для задач точного земледелия.

Ключевые слова: густота посевов, соя, компьютерное зрение, YOLOv8, детекция объектов, аугментация данных, Recall, наземная съёмка, машинное обучение.

Введение

Современный агропромышленный комплекс (АПК) всё больше ориентируется на цифровизацию и внедрение интеллектуальных технологий для повышения эффективности сельскохозяйственного производства. Одной из ключевых задач в управлении посевами является оперативный и точный анализ их качества, в частности — оценка густоты стояния растений, поскольку всхожесть растений напрямую показывает урожайность и рентабельность возделывания сельскохозяйственных культур. Традиционные методы оценки, основанные на ручных измерениях, трудоёмки, субъективны и не обеспечивают достаточного пространственного охвата. В этой связи методы искусственного интеллекта (ИИ), в сочетании с дистанционными данными (дроны, спутники, наземные сенсоры), открывают новые возможности для автоматизированного и масштабируемого мониторинга агроэкосистем.

Особый интерес представляет применение ИИ-решений в регионах с интенсивным растениеводством, но при этом с выраженной пространственной и климатической спецификой. Амурская область, являясь одним из ключевых регионов России по производству сои, сталкивается с рядом вызовов, связанных с коротким вегетационным периодом, переменчивой погодой и необходимостью точного соблюдения агротехнологий. Соевые посевы в этом регионе требуют тщательного контроля густоты стояния уже на ранних фазах развития, поскольку неравномерность всходов или избыточная загущённость могут существенно снизить продуктивность культуры. Использование алгоритмов компьютерного зрения и машинного обучения позволяет на основе нескольких фото быстро идентифицировать и количественно оценивать число растений на единицу площади.

В последние годы в российской и международной научной литературе появляется всё больше исследований, посвящённых применению ИИ для анализа состояния посевов [1–4].

Цель данной работы — построение и обучение модели глубокого обучения на основе архитектуры YOLOv8n для детекции ростков сои на наземных изображениях с последующей оценкой влияния методов аугментации данных на ключевые метрики качества, в первую очередь на полноту (Recall), определяющую точность оценки густоты посевов.

1. Сбор, обработка и разметка фотоданных

Для решения задачи количественной оценки густоты стояния сои на основе методов искусственного интеллекта был реализован полный цикл подготовки данных: от полевой съёмки до разметки изображений и формирования обучающей выборки. Ниже описаны ключевые этапы этого процесса, обеспечивающие достоверность и воспроизводимость результатов.

Фотографирование соевых посевов осуществлялось в фазе трёх настоящих листьев (V3 по шкале Fehr и Caviness [5]), что соответствует 14–17 суткам после посева в условиях Амурской области. Съёмка проводилась в Бурейском районе Амурской области на полях сельхозпредприятия в течение июня–июля 2025 года. Изображения получались вручную агрономом хозяйства с помощью цифровой камеры смартфона. Для минимизации вариативности геометрии съёмки камера удерживалась на постоянной высоте 1,2 м над поверхностью почвы и под фиксированным углом примерно 90°. Съёмка проводилась днём в различных погодных условиях — как в солнечную, так и в пасмурную погоду, — чтобы повысить устойчивость будущей модели к естественным колебаниям освещённости. Общий объём первичной фотобазы составил около 1500 изображений.

На следующем этапе проводился строгий отбор изображений по следующим критериям: достаточная резкость (отсутствие смаза), равномерное освещение без сильных бликов или глубоких теней, соответствие фазе развития V3 (визуальная проверка по морфологии ростков). Из первоначального набора было исключено 500 снимков, не соответствующих указанным требованиям. В результате сформирована репрезентативная выборка из 1000 изображений. Для повышения обобщающей способности модели изображения были классифицированы по типам визуальных условий: «низкая плотность всходов», «высокая плотность с частичными перекрытиями», «наличие сорной растительности», «прочие типы визуального шума, включая тени, остатки мульчи, неровности почвы».

Разметка изображений выполнялась вручную с использованием инструмента Roboflow [6]. Каждый росток сои был аннотирован в виде ограничивающего прямоугольника (bounding box). В целях обеспечения метрологической достоверности разметки и минимизации влияния субъективных ошибок аннотатора в работе была реализована процедура контроля качества разметки на репрезентативной подвыборке. Из общей совокупности размеченных изображений случайным образом было отобрано 10 снимков, охватывающих различные условия визуализации: низкую и высокую плотность всходов, наличие сорной растительности и геометрические искажения (тени, неровности рельефа). Эта подвыборка была использована для формирования тестового датасета, специально предназначенного для оценки воспроизводимости разметки и стабильности детекции на изображениях с заведомо высокой сложностью интерпретации. После чего на выборке проведено обучение пробной модели YOLOv8n [7]. Установлено, что качество аннотаций является методологически корректным: предсказанные моделью ограничивающие прямоугольники (bounding boxes) в целом соответствовали положению и количеству размеченных объектов, а расхождения носили преимущественно характер локальных неточностей, свойственных для начальной стадии обучения.

С учётом полученных результатов и в целях повышения репрезентативности обучающей выборки была реализована расширенная фаза разметки, в ходе которой общее количество размеченных изображений было увеличено до 30. Общее количество аннотированных объектов в итоговом датасете превысило 6000, что позволило перейти к полноценной стадии экспериментального обучения с применением комплексных методов аугментации и строгого разделения данных на обучающую, валидационную и тестовую подвыборки.

С учётом ограниченного объёма полевых данных и необходимости повышения робастности модели к естественным вариациям условий съёмки было принято решение реализовать контролируемый эксперимент, направленный на количественную оценку влияния методов

аугментации изображений на качество обучения. Для этого на основе единой размеченной выборки из 30 изображений были сформированы два независимых датасета: контрольный — без применения аугментаций, и экспериментальный — с использованием комплекса преобразований, включавшего как цветовые, так и геометрические и композиционные методы.

Полученный материал представляет собой надёжную основу для последующего этапа — обучения и тестирования модели на основе архитектуры YOLOv8.

2. Настройка и обучение нейронной сети

В рамках решаемой задачи для детекции и количественной оценки ростков сои на статических изображениях применяется архитектура YOLO (*You Only Look Once*) — современный подход к одноэтапной детекции объектов, отличающийся высокой точностью локализации и эффективностью обучения [10].

На начальном этапе экспериментального цикла в качестве базовой архитектуры была выбрана модель YOLOv8n (nano) — наименьшая по глубине и числу параметров версия семейства YOLOv8 [7]. Как показывают исследования в области машинного обучения, при малом количестве обучающих примеров более сложные модели (например, YOLOv8m, l или x) склонны к переобучению, поскольку обладают избыточной выразительной мощностью по сравнению с информационной ёмкостью данных. В таких условиях модель запоминает шум и особенности отдельных снимков, а не обобщённые паттерны, что приводит к снижению точности на новых данных. В то же время компактные архитектуры, такие как nano, содержат значительно меньше параметров (~3,2 млн), что накладывает естественные ограничения на сложность усваиваемых зависимостей и способствует лучшей обобщающей способности в условиях низкой репрезентативности выборки. Кроме того, YOLOv8n сохраняет ключевые архитектурные преимущества полноразмерных версий — включая anchor-free детекцию, декаплированную голову и поддержку современных аугментаций (в том числе Mosaic), что делает её методологически полноценной, несмотря на компактность.

Для объективной оценки влияния аугментации изображений на качество детекции ростков сои были независимо обучены две идентичные модели YOLOv8n — на контрольном датасете (без аугментаций) и на экспериментальном датасете (с применением комплекса аугментаций). Обучение проводилось в полностью контролируемых условиях: использовались одинаковые архитектура модели, гиперпараметры оптимизации и аппаратные ресурсы, что исключало влияние внешних факторов на результаты сравнения.

Обучение осуществлялось с помощью официальной библиотеки Ultralytics с фиксированным набором гиперпараметров (табл. 1).

При оценке качества моделей детекции растений в задачах анализа густоты посевов обычно применяются стандартные метрики компьютерного зрения — Precision (точность), Recall (полнота), их гармоническое среднее F1-score, а также IoU (Intersection over Union, измеряет перекрытие между предсказанными и истинными bounding boxes, рассчитывается как отношение площади пересечения к площади объединения), mAP (mean Average Precision, средняя точность по всем классам объектов, учитывает как точность обнаружения, так и локализацию объектов), AP50 (Average Precision при пороге IoU 50 %). Из них наиболее информативной в данном контексте является Recall, поскольку она отражает долю реально существующих растений, успешно обнаруженных моделью. Пропущенные растения (ложноотрицательные случаи, FN) напрямую ведут к снижению оценки густоты стояния, что критично для агрономических решений. В то же время Precision менее значим: ложные срабатывания (ложноположительные случаи, FP) могут быть частично устранены на этапе постобработки (например, с помощью фильтрации по размеру, форме или пространственной плотности), тогда как пропущенные растения восстановить невозможно.

Таблица 1

Гиперпараметры обучения моделей YOLOv8n

Гиперпараметр	Значение	Описание
Архитектура модели	YOLOv8n	Компактная версия архитектуры YOLOv8
Количество эпох	50	Максимальное число эпох обучения
Ранняя остановка (patience)	10	Прекращение обучения, если метрика валидации не улучшается в течение 10 эпох
Размер изображения (imgsz)	640×640	Входное разрешение изображений
Размер батча (batch size)	16	Количество изображений в одном батче
Начальная скорость обучения (lr_0)	0.01	Базовый learning rate для SGD
Оптимизатор	SGD с моментом	Стандартный оптимизатор, рекомендованный авторами YOLOv8
Устройство (device)	CPU	Обучение выполнено на центральном процессоре
Число рабочих потоков	4	Количество параллельных процессов загрузки данных

3. Анализ качества обучения

В данном разделе представлено сравнение моделей детекции ростков сои на изображениях с использованием YOLOv8n в двух сценариях: без аугментации данных и с применением аугментации (с использованием встроенных инструментов Roboflow). Главная задача — оценка влияния аугментации на качество модели, особенно по метрике Recall, так как пропущенные растения критичны для оценки густоты посевов. Каждая модель обучалась на 50 эпохах. Для каждого сценария после обучения загружалась лучшая модель и для нее осуществлялась визуальная проверка работы.

На рис. 1 представлена динамика изменения метрик обучения на наборе данных без применения аугментации. На всех графиках видна тенденция к улучшению метрик и снижению потерь, что свидетельствует о успешном процессе обучения. Потери на обучающей выборке демонстрируют устойчивое снижение, что указывает на хорошее качество обучения без переобучения на тренировочных данных. Модель показала высокую точность (Precision) и хорошую полноту (Recall), что делает её пригодной для практического применения в задаче оценки густоты посевов, особенно если использовать постобработку для фильтрации ложных срабатываний.

На рис. 2 представлена динамика изменения метрик обучения на наборе данных с применением аугментации. Обучающие потери ($train/box_loss$, $train/cls_loss$, $train/dfl_loss$) стабильно снижаются, что свидетельствует о хорошем обучении. Валидационные потери (val/box_loss , val/cls_loss , val/dfl_loss) также демонстрируют тенденцию к снижению, хотя и с некоторыми колебаниями (особенно в val/box_loss на 20–30 эпохах). Это нормально для небольших или разнородных наборов данных. Небольшие скачки в val/box_loss могут быть связаны с тем, что аугментация добавила сложности (например, повороты, изменение яркости), и модель временно «теряет» точность позиционирования, но затем восстанавливает её. Модель с аугментацией показала высокую точность и хорошую полноту, что делает её пригодной для практического применения в задаче оценки густоты посевов.

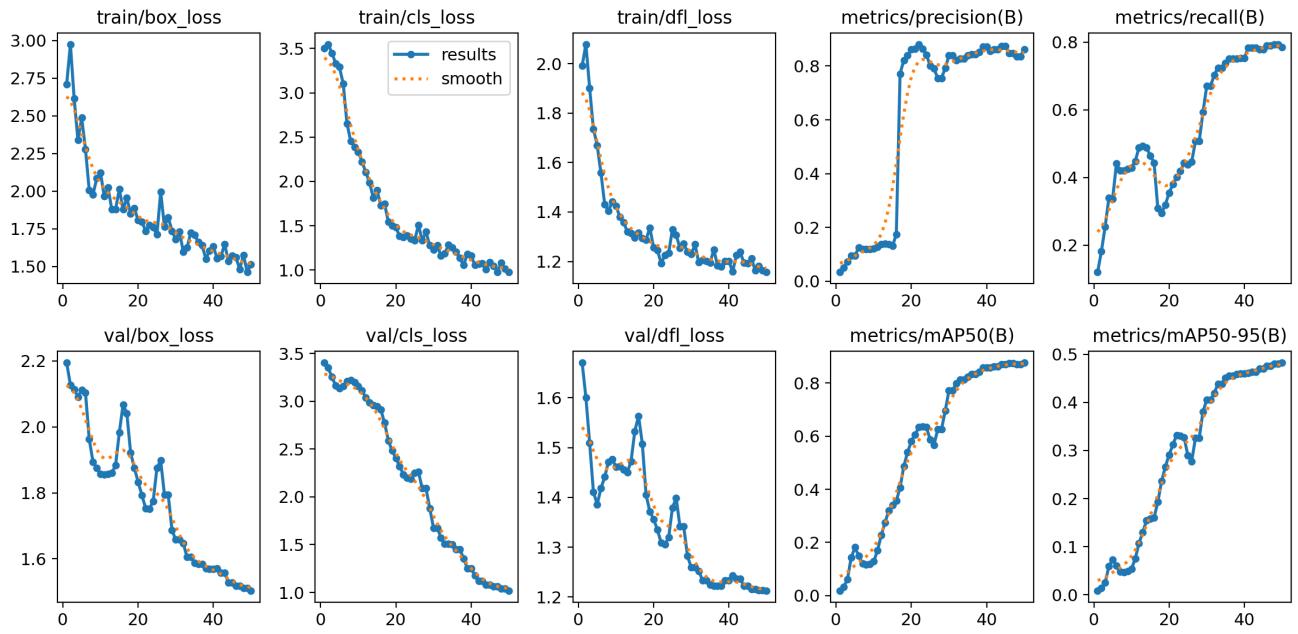


Рис. 1. Метрики обучения на наборе данных без применения аугментации

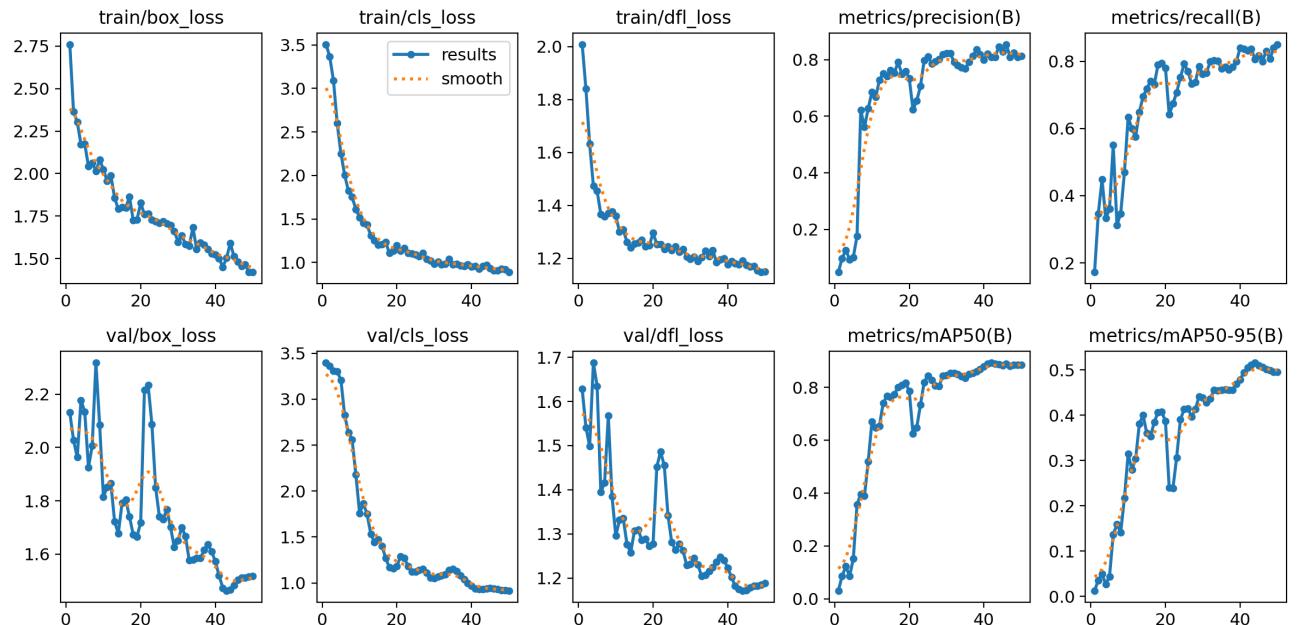


Рис. 2. Метрики обучения на наборе данных с применением аугментации

Матрица ошибок (рис. 3) показывает, как модель классифицирует объекты по отношению к их истинным меткам. В данном случае у нас два класса: soya (растки сои) и background (фон). При сравнении матриц ошибок, видим, что применение аугментации данных улучшило качество модели для задачи детекции ростков сои. Особенно заметен рост Recall, что критично для оценки густоты посевов. Модель стала более устойчивой к вариациям условий съёмки, что делает её пригодной для практического использования в полевых условиях. Этот же факт подтверждается на рис. 4: обе модели начинают с низкого значения Recall (около 0.1–0.2), затем модель с аугментацией в средней части обучения (эпохи 10–30) опережает по качеству модель без аугментации и к финальной фазе (эпохи 30–50) модель с аугментацией стабильно превосходит модель без аугментации, на последней эпохе Recall с аугментацией составляет около 0.85, Recall без аугментации — около 0.78.

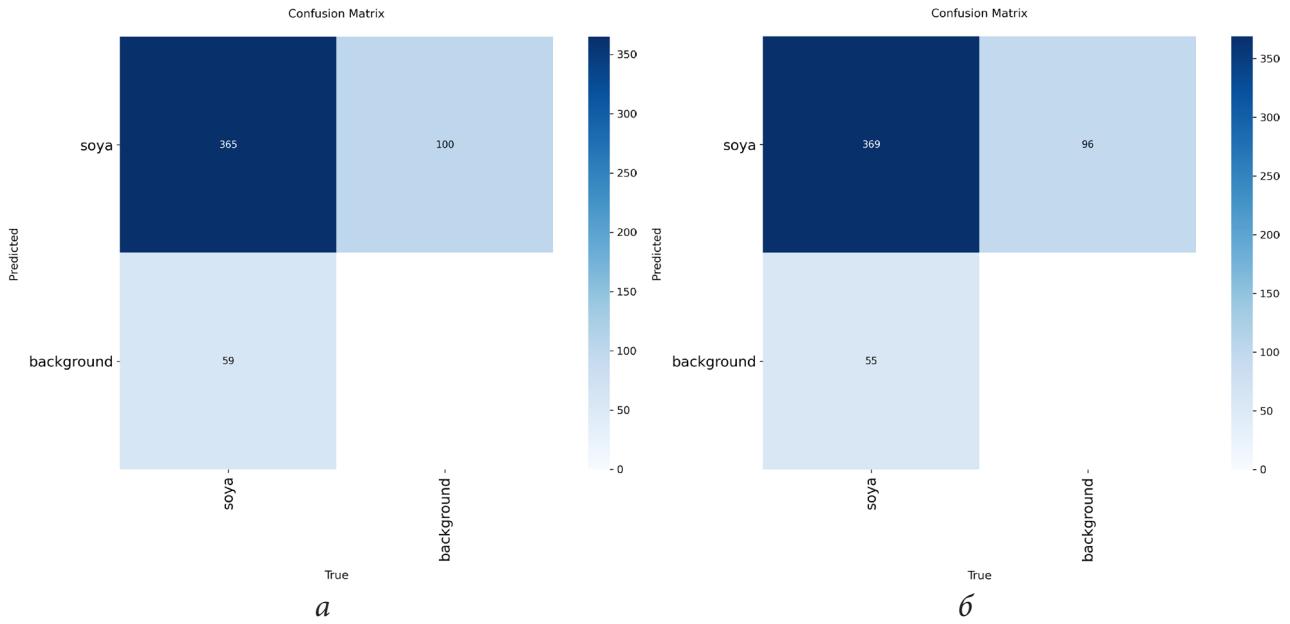


Рис. 3. Матрицы ошибок (а — на наборе без аугментации, б — на наборе с аугментацией)

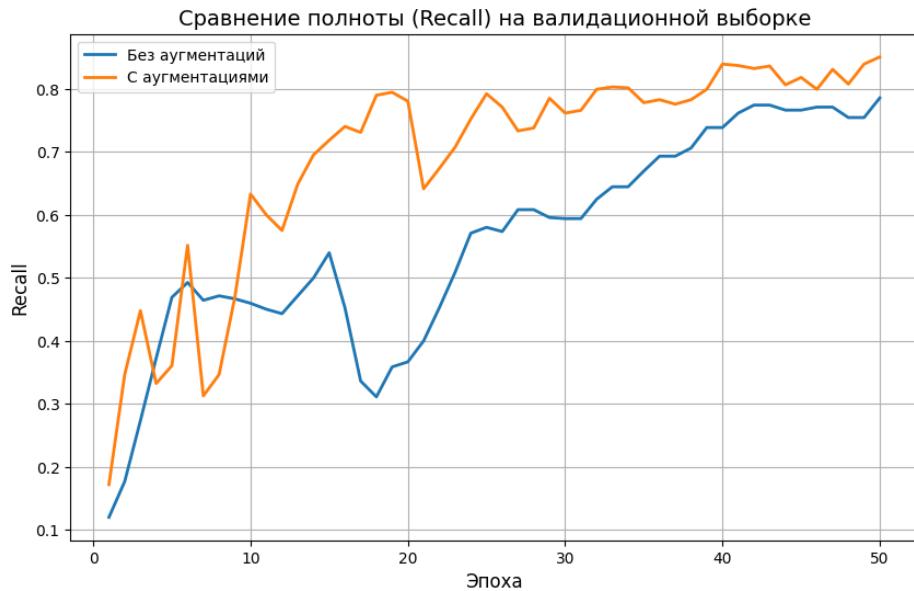


Рис. 4. Динамика изменения метрики Recall на валидационной выборке

Для демонстрации результатов обучения на рис. 5 представлен результат детекции ростков на одном и том же изображении для обеих моделей. Модель без аугментации обнаружила 45 объектов, модель с аугментацией — 49. Аналогичные результаты получаются и при анализе других изображения: модель с аугментацией показывает стабильно лучшие результаты.

В целом результаты обучения показали, что аугментация данных положительно повлияла на качество модели, особенно по ключевой метрике — Recall. Это подтверждает гипотезу, что аугментация помогает модели лучше обобщать и работать в реальных условиях. Для практического применения в задачах точного земледелия рекомендуется использовать модель, обученную с применением аугментации, так как она обеспечивает более полную детекцию объектов, что напрямую влияет на точность оценки густоты посевов.

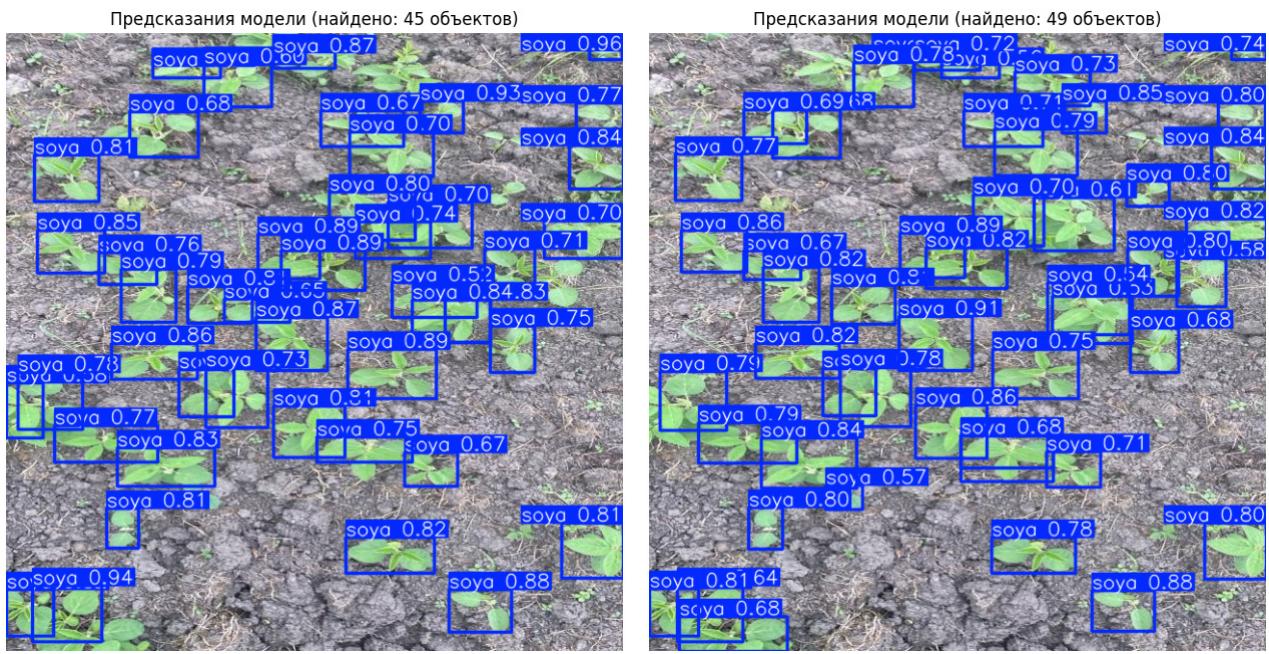


Рис. 5. Анализ изображения (а — на наборе без аугментации, б — на наборе с аугментацией)

Заключение

В ходе исследования разработана методика автоматизированной оценки густоты сои с использованием модели YOLOv8n на основе 30 наземных изображений, содержащих более 6000 размеченных ростков. Показано, что применение комплексной аугментации данных (геометрической, фотометрической и Mosaic) повышает метрику Recall с 0.78 до 0.85, что критически важно для точности оценки густоты, поскольку минимизирует пропуск реальных растений. Модель, обученная на аугментированных данных, демонстрирует повышенную устойчивость к визуальному шуму и вариациям освещённости, без роста ложных срабатываний. Полученные результаты подтверждают целесообразность использования аугментации при обучении моделей компьютерного зрения в условиях ограниченных полевых данных.

Литература

1. Alabi T. R. Estimation of soybean grain yield from multispectral high-resolution UAV data with machine learning models in West Africa / T. R. Alabi, A. T. Abebe, G. Chigeza, K. R. Fowobaje // Remote Sensing Applications: Society and Environment. – 2022. – V. 27. – P. 100782. – <https://doi.org/10.1016/j.rsase.2022.100782>
2. Wang Ch.-Y. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors / Ch.-Y. Wang, A. Bochkovskiy, H.-Y.M. Liao // Computer Science. Computer Vision and Pattern Recognition. – 2022. – <https://doi.org/10.48550/arXiv.2207.02696>
3. Zhang B. An Ensemble Learning Model for Detecting Soybean Seedling Emergence in UAV Imagery / B. Zhang; D. Zhao // Sensors. – 2023. – V. 23(15). – P. 6662. – <https://doi.org/10.3390/s23156662>
4. Kozhekin M. V. Plant Detection in RGB Images from Unmanned Aerial Vehicles Using Segmentation by Deep Learning and an Impact of Model Accuracy on Downstream Analysis / M. V. Kozhekin, M. A. Genaev, E. G. Komyshev, Z. A. Zavyalov, D. A. Afonnikov // Journal of Imaging. – 2025. – V. 11, № 1. – P. 28. – <https://doi.org/10.3390/jimaging11010028>

5. *Fehr W. R. Stages of soybean development* / W. R. Fehr, C. E. Caviness. – Ames, IA: Iowa State University, 1977. – 12 c. – (Special Report; № 80)
6. Roboflow: Computer vision tools for developers and enterprises [Электронный ресурс]. URL: <https://roboflow.com/> (дата обращения: 25.08.2025)
7. Изучите Ultralytics YOLOv8 [Электронный ресурс]. URL: <https://docs.ultralytics.com/ru/models/yolov8> (дата обращения: 09.09.2025)
8. *Kunlin Z. A Deep Learning Image Augmentation Method for Field Agriculture* / Z. Kunlin, Sh. Yi, Zh. Xun, R. De, Ch. Xiaoxi // IEEE Access. – 2024. – V. 12. – P. 1-1. – <https://doi.org/10.1109/ACCESS.2024.3373548>
9. *Кукушкин М. А. Влияние аугментации на качество обучения моделей компьютерного зрения применительно к задачам детекции объектов с использованием съемки с БПЛА* / М.А. Кукушкин // Universum: технические науки. – 2024. – № 5-1(122). – С. 28–36. – <https://doi.org/10.32743/UniTech.2024.122.5.17599>
10. *Redmon J. You Only Look Once: Unified, Real-Time Object Detection* / J. Redmon, S. Divvala, R. Girshick, A. Farhadi // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2016. – P. 779–788. – <https://doi.org/10.1109/CVPR.2016.91>

СОЗДАНИЕ ФАРМАЦЕВТА-АССИСТЕНТА НА ОСНОВЕ RAG-СИСТЕМЫ

А. П. Буркова

Воронежский государственный университет

Аннотация. В статье проводится исследование по созданию фармацевта-ассистента. В ходе проведенного исследования была разработана RAG-система для автоматизации обработки фармацевтических запросов, объединяющая преимущества семантического поиска и генеративных языковых моделей. Проведен сравнительный анализ существующих решений, на основе которых был разработана система, которая демонстрирует высокую эффективность в решении задачи автоматизации консультирования и обладает значительным потенциалом для практического применения в здравоохранении.

Ключевые слова: RAG-система, интеллектуальный ассистент, языковые модели, BERT, векторизация, гибридный поиск, векторные базы данных, семантический поиск, обработка естественного языка, медицинские информационные системы.

Введение

В современном информационном мире объем данных с каждым днем становится все больше, что создает проблему информационного шума и перегрузки [1]. Анализ соцсетей, форумов и блогов показывает, что при выборе препаратов пациенты все чаще обращаются к интернету. Традиционные методы поиска в интернете или справочниках не всегда эффективны из-за того, что большая часть данных о лекарствах не структурирована и разбросана. При поиске лекарства важно учитывать множество факторов, о которых обычный человек даже не подумает. Таким образом в обществе существует необходимость в решении, которое быстро предоставляет точные, проверенные и персонализированные решения. Такое решение могут предоставить большие языковые модели [2]. Они способны генерировать связные текстовые ответы, основанные на паттернах в данных, на которых они обучались. Несмотря на способность формулировать ответы на медицинские темы, модель не обладает специализированными фармацевтическими знаниями и не интегрирована с профессиональными базами данных [3]. Это существенно ограничивает её практическую применимость в аптечных сетях. Коммерческий продукт Just AI «Бот-Фармацевт» разработан специально для автоматизации работы аптек [4]. Его архитектура сочетает технологии обработки естественного языка с жестко структурированными сценариями работы и глубокой интеграцией с профессиональными базами данных. Система демонстрирует высокую эффективность при обработке стандартных запросов, но сталкивается с трудностями при интерпретации нестандартных формулировок. Таким образом целью работы является разработка и проверка архитектуры фармацевта-ассистента на основе RAG-системы, обеспечивающей точный семантический поиск и генерацию ответов, основанных на фармацевтической базе данных. Задачи исследования:

1. Провести сравнительный анализ существующих решений и выбор RAG-архитектуры.
2. Выбрать и реализовать модель для семантической векторизации текста.
3. Выбрать и настроить векторную базу данных для эффективного хранения и поиска информации.
4. Реализовать прототип системы и оценить его работоспособность на примере решения типовых фармацевтических запросов.

1. Материалы и методы

1.1. Архитектура системы

В основе разрабатываемой системы лежит архитектура RAG [5], которая сочетает этап извлечения релевантной информации из внешнего хранилища знаний с этапом генерации ответа языковой моделью. Данный подход позволяет преодолеть ограничения LLM, обеспечивая привязку ответов к актуальным и достоверным данным. Работа системы включает следующие этапы (рис. 1.):

1. Поиск: Пользовательский запрос векторизуется. По векторному представлению в векторной базе данных осуществляется семантический поиск наиболее релевантных фрагментов текста из базы знаний.

2. Генерация: Найденные фрагменты текста и исходный запрос пользователя объединяются в специальный промпт, который передается языковой модели. LLM формирует итоговый, связанный и контекстуализированный ответ, строго основываясь на предоставленной информации.

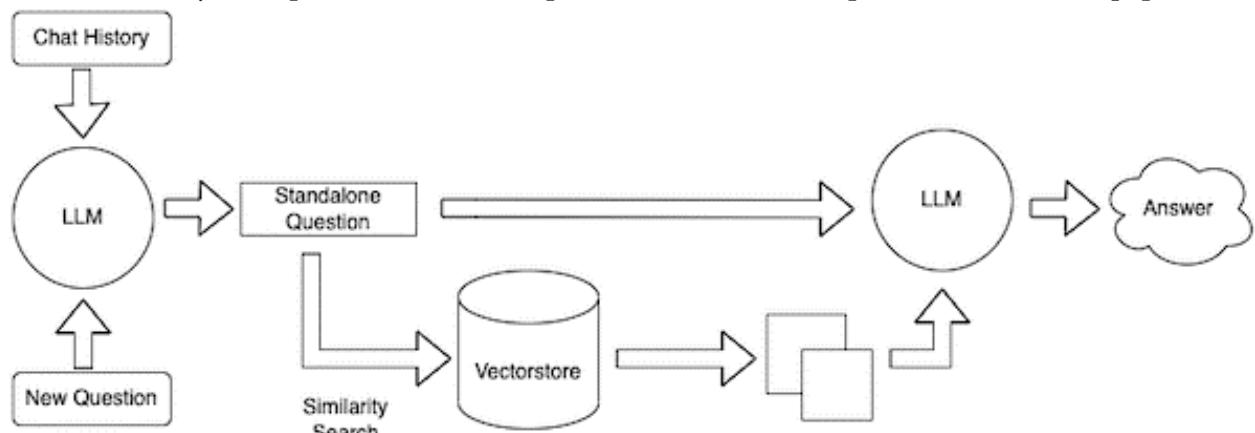


Рис. 1. Архитектура RAG

1.2. Данные

В качестве источника знаний использовалась база данных лекарственных средств «ВИДАЛЬ». Данные были представлены в виде структурированной таблицы (табл. 1).

Таблица 1

Список полей в таблице

Name	Type	Size	Description
1	2	3	4
DocumentID	Длинное целое	4	ID документа
RusName	Короткий текст	255	Наименование RUS
CompiledComposition	Длинный текст	–	Описание состава и формы выпуска
PhInfluence	Длинный текст	–	Описание фармакологического действия
Dosage	Длинный текст	–	Режим дозирования
OverDosage	Длинный текст	–	Передозировка
Interaction	Длинный текст	–	Лекарственное взаимодействие
Lactation	Длинный текст	–	Применение при беременности и кормлении грудью
SideEffects	Длинный текст	–	Побочное действие

Продолжение таблицы 3

1	2	3	4
StorageCondition	Длинный текст	–	Условия и сроки хранения
Indication	Длинный текст	–	Показания к применению
ContraIndication	Длинный текст	–	Противопоказания к применению
SpecialInstruction	Длинный текст	–	Особые указания
YearEdition	Короткий тест	50	Год издания
PregnancyUsing	Короткий текст	50	Противопоказания при беременности
NursingUsing	Короткий текст	50	Противопоказания при уходе
RenalInsuf	Длинный текст	–	Применение при нарушениях функции почек
HepatoInsuf	Длинный текст	–	Применение при нарушениях функции печени
PharmDelivery	Длинный текст	–	Условия отпуска из аптек
ElderlyInsuf	Длинный текст	–	Применение у пожилых пациентов
ChildInsuf	Длинный текст	–	Применение у детей

Перед загрузкой в систему данные прошли этап предобработки для последующей векторизации.

1.3. Методы векторизации и поиска

Критически важным этапом разработки являлся выбор модели эмбеддингов, обеспечивающей семантический поиск в медицинских текстах. Модель должна хорошо работать с русским языком, медицинскими терминами и опечатками. Был проведен сравнительный анализ нескольких моделей. Русскоязычная модель sentence-t5-xxl на базе BERT демонстрирует хорошую результативность на различных NLP-задачах и пригодная для дообучения. Большая русскоязычная модель ai-forever/sbert_large_nlu_ru, изначально ориентированная на задачи семантического сравнения текстов, но модель не обладает возможностью для использования в производственной среде без специализированного GPU-оборудования. Но ни одна из рассмотренных моделей не обладает встроенной устойчивостью к опечаткам, что указывает на необходимость этапа предварительной нормализации текста. По итогам анализа для реализации была выбрана модель ai-forever/sbert_large_nlu_ru как оптимальный компромисс между качеством семантических представлений, поддержкой русского языка и требованиями к вычислительным ресурсам. Для доступа к модели использовалась библиотека sentence-transformers.

1.4. Векторная база данных

Для хранения векторных представлений документов и эффективного семантического поиска была выбрана векторная база данных Weaviate, которая поддерживает гибридный поиск. Был развернут облачный кластер Weaviate, в котором была создана коллекция MedicalDocument с предварительно настроенной схемой, соответствующей структуре исходных данных.

1.5. Языковые модели для генерации

Критически важным решением при построении RAG-системы является выбор языковой модели, ответственной за генерацию финального ответа на основе извлеченного контекста. Модель должна не только генерировать связный и грамматически правильный текст на русском языке, но и строго следовать предоставленному контексту, минимизировать галлюцинации.

Был проведен сравнительный анализ нескольких популярных открытых моделей, подходящих для задач генерации на русском языке. Необходимо было, чтобы модель могла работать с русским языком и не была требовательна к ресурсам. Рассматривались модели с открытым кодом LLAMA, QWEN и DeepSeek, так как эти модели лежат в открытом доступе. LLAMA плохо работает с длинным контекстом из-за квадратичной сложности её механизма внимания, что приводит к экспоненциальному росту вычислительной нагрузки и требует больших вычислительных мощностей. Из-за того, что архитектура DeepSeek построена на методе MoE, то есть использует сеть из небольших экспертов, склонна к генерации правдоподобных, но не фактически верных ответов, что делает модель менее предсказуемой в задачах, требующих строгого следования предоставленному контексту. Поэтому была выбрана QWEN2.5. с 70 миллионами параметров, так как поведение этой модели легче всего контролировать.

Модель была развернута локально с использованием фреймворка vLLM. Для интеграции модели в RAG-конвейер был разработан специализированный шаблон промпта, направленный на минимизацию галлюцинаций и обеспечение строгого следования контексту:

Ты — ассистент-фармацевт. Ответь на вопрос пользователя, используя ТОЛЬКО информацию из предоставленного контекста.

КОНТЕКСТ:

{context}

ИНСТРУКЦИИ:

1. Если ответа на вопрос нет в контексте, сообщи: «В предоставленной информации нет ответа на ваш вопрос».
2. Не добавляй информацию, которой нет в контексте.
3. Сохраняй точные формулировки из контекста для дозировок и медицинских терминов.
4. Структурируй ответ для удобства чтения.
5. Добавь предупреждение о необходимости консультации с врачом.

ВОПРОС: {question}

ОТВЕТ:

2. Методика проведения эксперимента

Целью данного исследования стало изучение влияния дообучения языковой модели на медицинских данных на качество ответов. Была проведена проверка работы модели перед и после дообучения. В качестве исходных данных использовался набор предварительно подготовленных тестовых запросов, охватывающих широкий круг сценариев, и название и качества лекарств, которые помогут при данной проблеме:

- Вопросы о симптомах, методах лечения, лекарственных препаратах
- Вопросы, требующие особой осторожности в формулировках, чтобы избежать некорректных рекомендаций, связанных с самолечением.
- Вопросы, ответы на которые однозначно содержатся в базе знаний для проверки способности модели извлекать и использовать новый контекст.

Такие данные позволяют проверить, насколько хорошо модель запоминает и воспроизводит факты из предоставленного ей контекста, а также оценить способность давать развернутые, точные и безопасные ответы.

Эксперимент проводился с использованием двух подходов. Первый — классическое использование LLM без дообучения: на вопросы пользователя модель отвечала, опираясь исключительно на знания, полученные при обучении на открытых данных. Второй подход предусматривал дообучение модели с использованием RAG, где при обработке каждого пользовательского запроса система в реальном времени осуществляла семантический поиск и из-

влекала наиболее релевантные фрагменты текста из описания лекарства из целевой базы медицинских знаний. Затем эти фрагменты подавались в модель в качестве контекста, на основе которого она формировалась итоговый ответ.

При сравнении работы необходимы метрики, которые показывали, насколько полученные ответы верны. Использовались специальные метрики для проверки работы LLM METEOR и BERTScore для статистических оценок. METEOR учитывала не только точные совпадения ответов, но и синонимы, стемминг и парафразы, обеспечивая гибкую оценку. BERTScore измерял семантическое сходство на основе контекстуальных эмбеддингов, что позволяло оценивать смысловую близость ответов даже при различных формулировках. Статистические методы оценки надежны, но недостаточно точны, так как не учитывают семантику в полной мере и могут недооценивать семантически правильные, но лексически отличные ответы. Поэтому для оценки работы LLM, помимо статистических метрик, для оценки также использовалась LLM-as-judge score — методология, при которой более мощная языковая модель выступает в роли эксперта-оценщика, использовалась QAG Score, который автоматически генерирует контрольные вопросы из медицинского контекста и оценивает способность модели точно отвечать на них, что особенно ценно для проверки фактической точности в медицинской области.

3. Результаты эксперимента и анализ

Проведенное сравнение двух подходов продемонстрировало статистически значимое улучшение качества ответов модели при использовании метода RAG (табл. 2, 3). Результаты оценки ответов модели на тестовом наборе запросов показали преимущество подхода с RAG.

Таблица 2

Примеры ответов

Вопрос	Ответ QWEN	Ответ RAG	Ответ тестового набора
Что делать при передозировке парацетамолом?	При передозировке парацетамолом нужно немедленно обратиться к врачу. Могут быть тошнота, рвота, боль в животе. В больнице могут назначить антидот.	При передозировке парацетамолом нужно немедленно обратиться к врачу. Могут быть тошнота, рвота, боль в животе. В больнице могут назначить антидот.	Необходимо обращение к специалисту. Будет назначено лечение: принятие антидота и промывание желудка.
Можно ли принимать амоксициллин с алкоголем?	Сочетание антибиотиков с алкоголем обычно не рекомендуется.	Согласно инструкции, прием амоксициллина с алкоголем не рекомендуется. Этанол может снижать абсорбцию препарата и повышать риск побочных эффектов со стороны ЖКТ (тошнота, рвота, диарея).	При принятии амоксициллина с алкоголем возможна диарея или рвота.
У ребенка кашель и высокая температура.	Принимайте 10–15 мг./кг парацетамола каждые 5 часов или 5–10 мг./кг Ибупрофена каждые 6 часов.	При повышенной температуре принимайте Парацетамол 10–15 мг./кг на прием, каждые 4–6 часов или ибупрофен — 5–10 мг./кг на прием, каждые 6–8 часов. Важно не применять аспирин детям до 12 лет. При температуре выше 39°C или сохранении симптомов более 3 дней — срочно обратитесь к врачу.	Необходимо принять парацетамол — 10–15 мг./кг. каждые 4–6 часов или ибупрофен — 5–10 мг./кг. Каждые 6–8 часов.

Таблица 3

Сравнение работы

Метрика	LLM	RAG
METEOR	0, 646	0,79
BERTScore	0,491	0, 821
QAG Score	0, 721	0,837

В базовом сценарии модель продемонстрировала низкую точность. Ошибки в основном были связаны с устаревшей информацией, обобщениями или полным отсутствием знаний по узкоспециальным медицинским темам. С использованием RAG показатель точности вырос. Модель стабильно предоставляла фактологически верные ответы на вопросы, ответы на которые содержались в предоставленной базе знаний. Рост точности не сопровождался снижением полноты ответов, ответы с RAG были более детализированными и содержали ссылки на конкретные данные из контекста.

На запросы о новом методе лечения, LLM давала общий ответ, либо признавала отсутствие информации. Модель с RAG, обратившись к базе знаний, точно называла метод, его принцип действия и показания к применению. При вопросе, содержащем потенциально опасное предположение, базовая модель иногда генерировала нейтральный или обтекаемый ответ. Модель с RAG, благодаря доступу к руководствам и инструкциям, давала четкие и безопасные ответы, подчеркивая необходимость консультации с врачом.

Таким образом, экспериментальные данные однозначно свидетельствуют о высокой эффективности дообучения с использованием RAG для задач, требующих работы со специализированной и актуальной информацией.

Заключение

В ходе проведенного исследования была разработана RAG-система для автоматизации обработки фармацевтических запросов, объединяющая преимущества семантического поиска и генеративных языковых моделей. Анализ существующих решений показал, что универсальные языковые модели не обеспечивают достаточной точности в медицинской сфере из-за отсутствия специализированных знаний, а жестко структурированные коммерческие системы ограничены в обработке нестандартных запросов. Применение RAG-архитектуры позволило создать гибкую систему, которая динамически извлекает актуальные данные из медицинских источников и формирует ответы на их основе. Это обеспечивает высокую достоверность информации при сохранении естественности генеративных моделей. Выбор модели векторизации `ai-forever/sbert_large_nlu_ru` и её дообучение на медицинских данных повысили точность семантического поиска, включая обработку сложных терминов и опечаток. Использование Weaviate в качестве векторной базы данных доказало свою эффективность благодаря поддержке сложных запросов, гибридного поиска и масштабируемости, что критически важно для работы с большими объемами фармацевтической информации. Языковая модель QWEN продемонстрировала высокий уровень эффективности на сложной логике и незнакомых данных. Практическая значимость системы заключается в возможности её внедрения в аптечных сетях и медицинских учреждениях для ускорения обработки запросов, снижения рисков ошибок и повышения качества консультаций. Таким образом, разработанная система демонстрирует высокую эффективность в решении задачи автоматизации фармацевтического консультирования и обладает значительным потенциалом для практического применения в здравоохранении.

Литература

1. *Johnson A. E. [u др.] MIMIC-III, a freely accessible critical care database // Scientific Data.* – 2016. – Vol. 3. – Article number: 160035.
2. *Brown T. [u др.] Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems.* – 2020. – Vol. 33. – P. 1877–1901.
3. *Devlin J. [u др.] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of NAACL-HLT 2019.* – 2019. – P. 4171–4186.
4. *Evaluation of GPT-4 for Medical Consultation / J. Chen [и др.] // Nature Digital Medicine.* – 2023. – Vol. 6. – № 1. – P. 1–12.
5. *Lewis P. [u др.] Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks // Advances in Neural Information Processing Systems.* – 2020. – Vol. 33. – P. 9459–9474.
6. *Reimers N., Gurevych I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* – 2019. – P. 3982–3992.

РАНДОМИЗИРОВАННЫЕ СИСТЕМЫ ИТЕРИРОВАННЫХ ФУНКЦИЙ КАК МОДЕЛИ НЕЙРОННЫХ СЕТЕЙ

А. Г. Буховец, Т. Я. Бирючинская, Е. А. Семин

Воронежский государственный аграрный университет имени Императора Петра I

Аннотация. Предлагается рассмотреть и сравнить работу некоторых типов нейронных сетей и рандомизированных систем итерированных функций. Отмечается, что некоторое сходство в выполнении этих процедур позволяет говорить о близости полученных результатов. Анализ выполнения систем итерированных функций способствует установлению связи между различными типами нейронных сетей: рекуррентными и сверточными. Сравнение также предполагает характеризовать классы задач, наиболее подходящих для тех или иных процедур.

Ключевые слова: нейронные сети, рандомизированные системы итерированных функций, рекуррентные и сверточные сети, машинное обучение.

Введение

Обработка больших массивов данных предполагает наличие соответствующего программного обеспечения. Развитие такого направления как нейронные сети, являющегося частью более известного под термином искусственного интеллекта (ИИ), как нельзя лучше отвечает современным потребностям. Решение таких задач как классификация, поиск скрытых закономерностей, анализ пространственных структур и пр. напрямую связано с решением проблем создания соответствующих нейронных сетей. Алгоритмы нейронных сетей существенно используют возможности современных компьютерных технологий и демонстрируют заметные успехи в этом направлении.

Наряду с этим в теории фрактальных множеств развиваются подходы, связанные с использованием алгоритмов построения различного рода множеств, которые обладают свойствами, полезными с точки зрения анализа структур данных. При этом алгоритмы построения таких множеств имеют прозрачную интерпретацию, что позволяет в свою очередь использовать их для получения хорошо интерпретируемых выводов. Сравнивая процессы выполнения этих алгоритмов можно отметить большое сходство процедур, реализующих некоторые классы нейронных сетей. В связи с этим возникает задача сравнительного анализа этих подходов. И хотя рандомизированные системы итерированных функций (Random Iterated Function System - RIFS) не являются нейронными сетями в полном классическом понимании, они могут выполнять схожие задачи: аппроксимацию данных, генерацию сложных структур, моделирование хаотических процессов, проведение кластеризации.

1. Рекуррентные нейронные сети и рандомизированные системы итерированных функций

Как известно, рекуррентные нейронные сети (RNN) представляют класс нейронных сетей, предназначенных для обработки данных, имеющих структуру последовательностей таких как: временные ряды, генерация текста и т.п. Основная особенность RNN — наличие обратной связи [6, 8], что позволяет им сохранять информацию о предыдущих шагах последовательности, то есть обладание «памятью». Такими же свойствами обладает и схема RIFS в процедуре F1 [1].

Выполнение процедуры F1 сводится к рекуррентным/повторяющимся действиям. Система итерированных функций в случае процедуры F1 задаётся уравнениями $X_n = \xi X_{n-1} + (1-\xi)Z_j^{(n-1)}$,

$(n=1,2,\dots,N)$, где $Z_j^{(n-1)} \in Z = \{Z_j \mid p_j : j=1,\dots,K\}$, $p_j = p\{Z = Z_j\}$, $p_j > 0$, $\sum_j p_j = 1$, $X_n \in R^m$, $0 < \xi < 1$ — параметр системы. Такой вариант реализации RIFS в литературе обозначен F1 [1, 3]. Отметим, что параметр, представленный как $\xi = \frac{1}{1+\mu}$, где $\mu > 0$, можно интерпретировать как сигмоидную функцию $\sigma(\tau)$, где $\mu = e^{-\tau}$. В этом случае можно считать, что на вход процедуры, как и нейронной сети подаётся значение X_{n-1} , обработанное сигмоидной функцией активации ξ . Значение $Z_j^{(n-1)}$ в рамках этой процедуры можно рассматривать как смещение.

Такой подход позволяет интерпретировать рандомизированные системы итерированных функций с постоянными значениями гиперпараметров $K, \xi, \mathbf{P} = \{p_1, p_2, \dots, p_K\}, \mathbf{Z} = \{Z_1, Z_2, \dots, Z_K\}$ как модели рекуррентных нейронных сетей.

Входное распределение $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_K\}$ задаётся набором вероятностей $\mathbf{P}\{p_1, p_2, \dots, p_K\}$, которое определяет инвариантную меру аттрактора [9]. В этом случае F1 выглядит как очень глубокая рекуррентная сеть с конечным множеством значений предфрактала и значением параметра ξ . В таком представлении алгоритм F1 проявляет схожесть с RNN, которые по определению являются очень глубокими [7]. В этом случае последовательность $\mathbf{X} = \{X_i : i=1, \dots, N\}$, которая генерируется при выполнении, представляет собой скрытые состояния сети, и одновременно входы, которые характеризуются как внутренние состояния.

Алгоритм F1 предполагает, что значение параметра ξ задано. В случае если значение параметра нужно получить в процессе обучения RIFS, эта величина может оцениваться по имеющейся выборке, но в отличие от обратного распространения ошибки в RNN в качестве оценки выступает выборочное значение автокорреляционной функции [2]. Мы можем рассматривать, по сути, эту процедуру как вариант сети с прямой связью (т.н. полностью связная сеть), в которой связь устанавливается только с одним, случайно выбранным элементом следующего скрытого слоя. Фактически можно сказать, что на каждом шаге имеется внутренний слой скрытого состояния, состоящий из элементов $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_K\}$, но управление передается только одному каналу, случайно выбранному в соответствии с распределением $\mathbf{P} = \{p_1, p_2, \dots, p_K\}$. Это можно рассматривать как своего рода случайный выбор активации (аналогия с dropout) [6, 7].

Очевидно, аналогия в описании RNN и RIFS заключается также и в том, что обе системы работают итеративно, сохраняя получившееся состояние. Это свойство RNN было использовано при генерировании, например, траекторий броуновского движения, фрактальных кривых. Как и в случае RIFS, данное свойство позволяет интерпретировать каждое состояние нейронной сети как точку в пространстве состояний, а переходы — как случайно выбранные отображения. Всё это в полной мере соответствует интерпретации RIFS.

Некоторые исследования показывают, что глубокие нейронные сети можно интерпретировать как динамические системы или функциональные композиции, аналогичные RIFS. В публикациях неоднократно отмечалось, что глубокие нейронные сети, наверное, хуже всего поддаются интерпретации [5]. Возможно, это связано с тем, что траектория RIFS, как было показано [1], представляет фрактальное множество, описание которого требует специфического подхода.

Как было отмечено выше, RNN используются для анализа и моделирования множеств, представленных последовательностями значений. Процедура F1 была использована нами для прогнозирования урожайности зерновых культур — озимой пшеницы и ячменя [2].

2. Сверточные нейронные сети и рандомизированные системы итерированных функций

Результаты процедуры F1 представляют орграф, или линейную цепочку. В литературе [4] эту конструкцию иногда называют нейронной сетью прямого распространения или многослойным перцептроном. Выполняя эквивалентные преобразования [3] можно перейти к про-

цедуре F2, которая при тех же значениях параметров, что и для F1, позволяет получить топологически эквивалентную структуру [1, 3]. При этом случайный характер построения фрактального множества переносится на построение матрицы \mathbf{A} , строки которой представляют кортежи $A_i = (a_1, a_2, \dots, \hat{a}_m, \dots, a_K)$. Строки получаются суммированием членов ряда $\mu \sum a_n = 1$ по случайному выбранным K ячейкам согласно распределению \mathbf{P} . При таком построении матрицы \mathbf{A} в каждой строке имеется доминирующий элемент \hat{a}_m , величина которого превышает сумму всех остальных элементов кортежа. Положение этого элемента в строке определяет отнесение кортежа A_i к классу S_k , ($k = 1, 2, \dots, K$). Сам аттрактор процедуры представляется как произведение $\mathbf{X} = \mathbf{A}\mathbf{Z}$, где \mathbf{Z} представлена как матрица координат точек протофрактала. При этом аттрактор процедуры представляется как объединение непересекающихся кластеров $\mathbf{X} = \prod_{k=1}^n S_k$.

Изменяя положение доминирующего элемента, что возможно сделать с помощью соответствующего линейного преобразования, можно перенести кортеж A_i из класса S_k в любой другой класс S_l . Аналогично можно поступить и с любым другим элементом класса S_k . Другими словами, все элементы любого класса аттрактора можно отобразить биективно в заранее заданный класс. Это свойство является проявлением самоподобия аттрактора.

Процедуру F2 можно рассматривать как свёрточную нейронную сеть (CNN). В литературе отмечается, что такого рода сети лучше соответствуют задачам построения изображений и задачам кластеризации. При этом они, как и F2, должны иметь некоторую априорную информацию о пространственном изображении. Как было показано [1], это процедура обладает инвариантностью (безразличием) относительно следования элементов X_i . В целом можно показать, что процедура F2 представляет некий марковский процесс, т.е. представляет марковскую сеть. При определенных условиях результаты выполнения процедур F1 и F2 (с учётом случайного характера выполнения этих процедур) топологически эквивалентны. Для различения алгоритмов, порождающих данное топологическое пространство, предлагается использовать преобразование Лежандра. Процедура F2 была использована нами для построения районированной выборки и отбора типичных объектов [2].

Заключение

В работе мы стремились показать, что нейронные сети как RNN, так и CNN, особенно глубокие, можно рассматривать (в некотором приближении) как стохастические системы итерированных функций RIFS, где каждый слой — это случайное преобразование, а результатом выполнения является фрактальный аттрактор. Оценивание параметров при выполнении RIFS заменяет процесс настройки модели, который в целом составляет содержание машинного обучения. При таком подходе мы практически исключаем проблему переобучения [9, 10]. Фактически, мы задаем гиперпараметры, т. е. параметры априорного распределения и значения некоторых величин, участвующих в построении аттрактора.

Проведенное сравнение позволяет показать, что нейронные сети, особенно глубокие и рекуррентные, можно интерпретировать как рандомизированные системы итерированных функций, где каждый слой — это случайное преобразование, сходящееся к определённому аттрактору, который характеризуется распределением данных.

В рамках проведенной аналогии внутренние состояния нейронных сетей как RNN, так и CNN могут быть представлены фрактальными множествами. Формируя эти состояния, появляется возможность дать оценку и анализировать динамику моделируемого объекта: F1 — для рекуррентной сети, F2 — свёрточной.

Литература

1. *Bukhovets A. G. Modeling of fractal data structures / A. G. Bukhovets, E. A. Bukhovets // Automation and Remote Control.* – 2012. – Vol. 73, No 2. – P. 381–385. – DOI 10.1134/S0005117912020154.
2. *Буховец А. Г. О новых подходах к прогнозированию урожайности зерновых культур / А. Г. Буховец, Е. А. Семин, А. К. Горностаев // Современная экономика: проблемы и решения.* – 2023. – № 2(158). – С. 8–19. – DOI 10.17308/meps/2078-9017/2023/2/8-19. – EDN QGJBJO.
3. *Буховец А. Г. Фрактальные аспекты моделирования классификационной задачи / А. Г. Буховец, Е. А. Семин // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии.* – 2022. – № 3. – С. 127–138. – DOI 10.17308/sait/1995-5499/2022/3/127-138. – EDN TMGLCG.
4. *Калиев Д. И. Свёрточные нейронные сети для решения задач обнаружения пожаров по данным аэрофотосъемки / Д. И. Калиев, О. Я. Швец // Программные системы: теория и приложения.* – 2022. – Т. 13, № 1(52). – С. 195–213. – DOI 10.25209/2079-3316-2022-13-1-195. – EDN МЕРХJM.
5. *Николенко С. Глубокое обучение / С. Николенко, А. Кадурин, Е. Архангельская.* – СПб. : Питер, 2018. – 480 с.
6. *Хайкин С. Нейронные сети. Полный курс / С. Хайкин; Саймон Хайкин; [пер. с англ. Н. Н. Куссуль, А.Ю. Шелестова]. – 2-е изд. – М. [и др.] : Вильямс, 2006.* – EDN QMPELB.
7. *Методы разработки искусственных нейронных сетей / Е. И. Маркин, И. А. Подопригора, А. А. Зоткина, Е. Г. Бершадская // Вестник современных исследований.* – 2018. – № 3.1(18). – С. 49–52. – EDN XNDSCD.
8. *Францев А. Р. Нейронные сети для временных рядов: прогнозирование и анализ данных / А. Р. Францев // Шарыгинские чтения: Международная научная конференция ведущих научных школ в области радиолокации, радионавигации и радиоэлектронных систем передачи информации.* – 2024. – Т. 1, № 1. – С. 478–481. – EDN NND SKN.
9. *Kevin P. Murphy. Probabilistic Machine Learning: An introduction / K. P. Murphy.* – Cambridge, Massachusetts: MIT Press, 2022. – 940 с. – ISBN 9780262046824.
10. *Weinan E. A Proposal on Machine Learning via Dynamical Systems. Communications in Mathematics and Statistics.* – 2017. – 5(1). – P. 1–11. – DOI:10.1007/s40304-017-0103-z.

АНАЛИЗ КОНФИГУРАЦИЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПОИСКА ИНФОРМАЦИИ В НАУЧНЫХ ДОКУМЕНТАХ

И. С. Высотин, О. Ю. Марьясин

Ярославский государственный технический университет

Аннотация. В работе представлены результаты экспериментального тестирования различных конфигураций Retrieval-Augmented Generation (RAG) системы с варьированием ее ключевых параметров — размера чанка, типа векторных баз, моделей эмбеддингов и способа поиска релевантной информации. Анализ результатов экспериментов позволяет сделать вывод, что качество ответов RAG-системы существенно зависит от размера чанка и выбора модели эмбеддингов. Наилучшие результаты были получены при использовании размера чанка 500 символов и моделей эмбеддингов BAAI/bge-m3, intfloat/e5-large-v2 и nomic-embed-text. Варианты конфигурации RAG-системы с использованием данных параметров показали лучшие значения метрик, что свидетельствует о высокой степени смыслового совпадения между ответом модели и эталонным текстом.

Ключевые слова: поиск информации, retrieval-augmented generation, векторные базы, модели эмбеддингов, retriever, большие языковые модели, reranker, cross-encoder.

Введение

Современное развитие искусственного интеллекта характеризуется переходом от узкоспециализированных алгоритмов к универсальным моделям, способным решать широкий спектр задач обработки естественного языка. Наибольший прогресс наблюдается в области больших языковых моделей (Large Language Model — LLM), которые демонстрируют способность к рассуждению, обобщению знаний и формированию связных ответов на естественном языке. Однако ключевым ограничением подобных систем остаётся невозможность доступа к той информации, которая неизвестна модели, т.е. модель владеет только теми данными, которые были доступны на момент обучения, что приводит к устареванию знаний и снижению достоверности выводов [1].

Для преодоления данного ограничения активно развиваются гибридные подходы, объединяющие генеративные возможности LLM с методами информационного поиска. Одним из наиболее значимых направлений в этой области стала технология RAG, предложенная в 2020 году Патриком Льюисом [2]. Суть RAG заключается в интеграции внешнего модуля поиска, извлекающего релевантную информацию из постоянно обновляемых источников, с LLM, которая использует эти данные для построения осмысленных ответов. Главное преимущество RAG-систем по сравнению с одиночными LLM заключается в том, что они позволяют объединить интеллектуальные возможности LLM с постоянно обновляемой внешней базой знаний [3]. Это обеспечивает более точные, актуальные и обоснованные ответы, особенно при работе с корпоративной документацией, внутренними базами знаний или техническими регламентами.

Концептуально RAG опирается на идею разделения вычислительной и смысловой нагрузки между двумя модулями — системой семантического поиска (retriever) и LLM [2]. Её основная идея состоит в том, что LLM получает доступ к внешнему хранилищу знаний и использует найденные данные для генерации ответов, что повышает их достоверность и снижает вероятность появления «галлюцинаций» — некорректных или вымышленных фактов [4].

Архитектура RAG включает три ключевых компонента: источник данных, векторную базу и LLM. На подготовительном этапе исходные тексты (документы, статьи, регламенты и т. п.) проходят предобработку и разбиение на чанки (chunks), которые затем преобразуются в век-

торные представления с помощью моделей эмбеддингов (embedding models) [1]. Этот процесс, называемый векторизацией, обеспечивает возможность смыслового сопоставления текстов. Полученные векторы сохраняются во векторной базе данных. После поступления запроса он также преобразуется в вектор, и выполняется поиск наиболее релевантных фрагментов текста. Найденные фрагменты передаются в LLM, которая формирует итоговый ответ, опираясь на собственные знания и на данные из векторного хранилища. В отличие от LLM, где знания фиксированы в весах, в RAG актуальные сведения могут обновляться без переобучения — достаточно обновить содержимое векторного хранилища. Это делает технологию особенно эффективной для корпоративных систем с постоянно пополняемыми базами знаний и для задач, требующих обращения к специализированным документам.

Актуальность исследований в области RAG обусловлена необходимостью повышения достоверности, воспроизведимости и интерпретируемости ответов, формируемых LLM. Применение подобных систем становится особенно важным при решении задач корпоративного поиска, анализа научно-технической документации и построения интеллектуальных справочных систем.

1. Обзор литературных источников

Далее представлен краткий обзор литературных источников, в которых описаны исследования, подобные тем, что были выполнены в данной работе.

Базовая работа [2] посвящена всестороннему исследованию и анализу технологии RAG, при этом особое внимание уделено её применению в области обработки естественного языка (Natural Language Processing — NLP). Авторы рассматривают особенности интеграции механизмов поиска и генерации в контексте задач, требующих глубокого понимания текста и работы с большими объемами знаний. В рамках исследования описываются методы оценки качества генерации ответов и релевантности извлечённой информации, что позволило заложить основы последующих разработок в области RAG-систем для задач NLP.

В работе [5] авторы стремились определить оптимальные практики реализации RAG для повышения качества и надежности контента, создаваемого LLM. Авторы провели обширные эксперименты, в ходе которых изменялись: размер чанка (128, 512, 1024, 2048), модели эмбеддингов (14 моделей, включая intfloat/e5-large-v2 и sentence-transformers/all-mrpc-base-v2), векторные базы (Weaviate, Faiss, Chroma, Qdrant, Milvus), методы поиска релевантной информации (retrieval search) и другое. Основываясь на результатах проведенных экспериментов, авторы предложили несколько стратегий развертывания RAG, которые обеспечивают баланс между производительностью и эффективностью.

В [6] описываются различные подходы к сегментации текстов, направленные на оптимизацию баланса между полнотой и точностью представления контекста. В исследовании рассматривались такие методы как Fixed-size Chunker, Breakpoint-based Semantic Chunker и Clustering-based Semantic Chunker. Метод Fixed-size Chunker, основан на разбиении текста на фрагменты фиксированной длины (например, по числу токенов или символов). Данный подход отличается простотой реализации и предсказуемым размером контекста. Результаты исследования показали, что метод Fixed-size Chunker остается более эффективным и надежным выбором для практических приложений RAG.

В [7] подробно описывается широко распространённый фреймворк LangChain. Этот фреймворк представляет собой модульную библиотеку на языке Python для построения и интеграции приложений на основе LLM. Он обеспечивает инфраструктуру для объединения различных компонентов RAG-системы в единую цепочку — от загрузки документов и разбиения их на чанки до генерации ответов с использованием внешних данных. Его архитектура построена по принципу «цепочек» (chains), где каждая операция — загрузка, разбиение, векторизация,

поиск или генерация оформлена как отдельный модуль, что обеспечивает гибкость и возможность экспериментального сравнения различных конфигураций. LangChain поддерживает множество векторных хранилищ, моделей эмбеддингов и LLM.

2. Описание экспериментов

Целью экспериментальной части являлась оценка влияния ключевых конфигурационных параметров RAG на качество извлечения информации и точность формируемых ответов. В качестве исходного корпуса документов использовалась большая коллекция научно-технических статей из журналов издательств Elsevier, MDPI и сборников трудов международных конференций, на английском языке, в формате pdf, посвящённых моделированию и оптимизации мультиэнергетических систем (multi-energy systems), а также связанным с этим, вопросам. В статьях анализировались концепции интеграции различных энергетических сетей, концепции SmarGrid и microGrid, роль возобновляемых и комбинированных источников энергии, сетевых накопителей в сетях SmarGrid и microGrid, методы оптимизации энергоснабжения и энергопотребления в энергетических системах.

Для загрузки информации из документов в формате PDF применялся загрузчик PyPDFLoader, входящий в библиотеку LangChain. После загрузки содержимое документов подвергалось разбиению на смысловые фрагменты (чанки) с использованием класса RecursiveCharacterTextSplitter библиотеки LangChain. Алгоритм RecursiveCharacterTextSplitter использует рекурсивный подход и список разделителей (перевод строки, пробел, пустая строка), чтобы адаптироваться к естественной иерархической структуре текста (абзацы, предложения, слова).

При проведении экспериментов, размеры чанков устанавливались в 500, 1000 и 1500 символов, перекрытие между соседними чанками отсутствовало — каждый новый фрагмент начинался сразу после окончания предыдущего. Как было показано в работе [5], размер чанков в 500 и менее символов, не дает повышения эффективности результатов. Поэтому размер чанков менее 500 символов в экспериментах не использовался.

В качестве моделей эмбеддингов применялись решения, вошедшие в рейтинг Massive Text Embedding Benchmark (MTEB) [8], включающий более 300 моделей для более 1000 языков. В исследовании было протестировано 7 разных моделей от лёгких и производительных MiniLM до крупных и высокоточных E5 и BGE: sentence-transformers/all-MiniLM-L6-v2 (117), sentence-transformers/all-mpnet-base-v2 (102), thenlper/gte-large (66), nomic-embed-text (57), intfloat/e5-large-v2 (53), BAAI/bge-m3 (22) и intfloat/multilingual-e5-large-instruct (7). В скобках приведены значения в рейтинге MTEB на момент проведения экспериментов. Данные модели различаются по различным показателям, включая размерность эмбеддингов, максимальное количество токенов, объём памяти и количество параметров, а также по среднему результату на различных типах задач (извлечение, ранжирование, кластеризация, классификация и т. д.), согласно сводной статистике рейтинга MTEB. Это позволяет экспериментально исследовать, как размерность эмбеддингов, максимальное количество токенов и другие параметры влияют на качество поиска и итоговой генерации.

Для хранения и обработки векторных представлений документов применялись векторные базы Chroma, Qdrant и SKLearn. Такой выбор векторных баз был сделан из следующих соображений: высокой популярности некоторых из них (Chroma, Qdrant), мультиплатформенности и желания использовать их локальные, а не облачные варианты. Локальные векторные хранилища обеспечивают безопасное и приватное хранение данных. По этим причинам из исследования были исключены некоторые другие популярные векторные базы, такие как Weaviate или Milvus.

В ходе проведения экспериментов использовались различные подходы к поиску релевантной информации в векторных базах данных, составляющих retrieval-компонент RAG-системы. Первый подход основан на прямом поиске по сходству (similarity search), реализующем вычисление косинусного расстояния между вектором запроса и векторами текстов документов в коллекции. В результате возвращаются фрагменты, наиболее близкие по семантическому содержанию к исходному запросу. Этот метод отличается простотой реализации и высокой скоростью работы, однако не всегда обеспечивает разнообразие контекста, поскольку найденные документы могут содержать дублированную или частично пересекающуюся информацию.

Второй подход — поиск с использованием retriever. Он представляет собой более продвинутую схему, предоставляемую библиотекой LangChain. В данном случае применялся алгоритм MMR (Maximal Marginal Relevance), минимизирующий избыточность выдачи и обеспечивающий более широкий охват смысловых аспектов запроса. Он позволяет формировать контекст, который лучше отражает смысловое разнообразие информационной базы и повышает информативность итогового ответа.

Третий вариант, реализованный в рамках экспериментов, представляет собой комбинированный метод, включающий similarity search вместе с LLM. Он соответствует полной архитектуре RAG. На первом этапе выполняется поиск релевантных документов. Найденные фрагменты затем объединяются с исходным запросом пользователя посредством промпта-шаблона rlm/rag-prompt из LangChain Hub. Полученный расширенный контекст передаётся в LLM, которая формирует итоговый ответ, опираясь на реальные данные из векторного хранилища. В качестве LLM применялась популярная, компактная модель llama3.2 3B, запускаемая локально с помощью фреймворка Ollama.

Таким образом, в рамках проводимых экспериментов изменились следующие основные параметры конфигурации RAG-системы: размер чанка, тип векторных баз, моделей эмбеддингов и способ поиска релевантной информации.

3. Анализ результатов

В результате выполнения серий запусков были собраны многочисленные результаты, включающие параметры RAG-системы и полученные ответы. Каждая запись включала сведения о типе векторной базы, размере чанка, используемой модели эмбеддингов, методе поиска и текстовом ответе, что позволило провести последующий анализ зависимости качества ответов от различных факторов.

Оценка правильности и релевантности ответов, формируемых RAG-системой, проводилась с использованием трёх независимых подходов, основанных на современных моделях ранжирования текстовых пар и метриках сходства. Данные действия проводились для получения количественных показателей соответствия сгенерированных ответов эталонным определениям, отражающим корректное описание исследуемого понятия.

Первый метод оценки основан на применении модели кросс-энкодера (cross-encoder/ms-marco-MiniLM-L-6-v2) семейства MiniLM. Кросс-энкодер — модель, которая принимает эталонный и сгенерированный текст, а затем оценивает степень их семантического сходства на уровне предложений.

Второй способ использует модель reranker (BAAI/bge-reranker-v2-m3), которая представляет собой улучшенную архитектуру для задачи passage reranking. Данная модель обучена на множестве текстовых пар «вопрос–ответ» и выполняет более сложное семантическое сопоставление, оценивая вероятность того, что сгенерированный текст отвечает поставленному запросу. В отличие от кросс-энкодера, reranker оптимизирован именно для задач поиска и отбора релевантных текстовых фрагментов. Итоговые значения позволяют оценить, насколько полученные ответы соответствуют ожиданиям с точки зрения их содержательной точности.

Третий способ оценки основывался на использовании метрик ROUGE, которые часто применяются для проверки качества автоматически сгенерированных текстов [9]. В работе применялись три показателя: ROUGE-1, ROUGE-2 и ROUGE-L. Первая метрика (ROUGE-1) оценивает, сколько отдельных слов совпадает между ответом модели и правильным текстом. Вторая (ROUGE-2) анализирует совпадения пар слов, что позволяет лучше учитывать контекст. Третья метрика (ROUGE-L) показывает, насколько длинные последовательности слов совпадают в обоих текстах. Чем выше значения этих показателей, тем ближе ответ модели по смыслу и структуре к правильному.

Совокупное использование этих трёх способов обеспечивает многоуровневую оценку качества ответов RAG-системы. Такой подход обеспечивает более объективные показатели корректности, полноты и логической согласованности полученных ответов. Значения оценок для всех способов были приведены к диапазону от 0 до 1.

Все представленные далее результаты были получены для примера тестового запроса “What is Multi-Energy System?” и эталонного текста ответа на данный запрос. Значения оценок в зависимости от размера чанка для векторной базы Chroma, модели эмбеддингов nomic-embed-text и разных способов поиска релевантной информации приведены в табл. 1.

Таблица 1

Значения оценок в зависимости от размера чанка

Размер чанка	Reranker	CrossEncoder	ROUGE-1	ROUGE-2	ROUGE-3
similarity search					
500	0,922	0,937	0,931	0,904	0,931
1000	0,445	0,487	0,298	0,065	0,169
1500	0,564	0,567	0,279	0,052	0,143
retriever = MMR					
500	0,922	0,937	0,931	0,904	0,931
1000	0,445	0,487	0,298	0,065	0,169
1500	0,564	0,567	0,279	0,052	0,143
similarity search + LLM					
500	0,967	0,918	0,662	0,467	0,561
1000	0,783	0,724	0,338	0,055	0,176
1500	0,565	0,27	0,243	0,071	0,139

Из анализа табл. 1 видно, что самые высокие оценки по всем показателям получаются для размера чанка равного 500, независимо от используемого способа поиска релевантной информации. Для размеров чанка 1000 и 1500 оценки получаются значительно ниже. Аналогичная картина наблюдалась и для других векторных баз. Это можно объяснить тем, что при увеличении размера чанка свыше 500 повышалось покрытие контекста, но снижалась избирательность, что иногда приводило к включению нерелевантных фрагментов и потере точности. Данный результат хорошо согласуется с результатами, полученными в работе [5].

Значения оценок в зависимости от используемой модели эмбеддингов для векторной базы Chroma, размера чанка 500 и разных способов поиска релевантной информации приведены в табл. 2. Модели эмбеддингов расположены в порядке убывания значения в рейтинге MTEB.

Анализ данных табл. 2 показывает, что наилучшие результаты показали модели эмбеддингов BAAI/bge-m3, intfloat/e5-large-v2 и nomic-embed-text. При этом, на примере модели multilingual-e5-large-instruct, видно, что высокое положение в рейтинге MTEB не всегда гарантирует высокие оценки. Аналогичные результаты наблюдались и для других векторных баз.

Таблица 2

Значения оценок в зависимости от модели эмбеддингов

Модель эмбеддингов	Reranker	CrossEncoder	ROUGE-1	ROUGE-2	ROUGE-3
similarity search					
multilingual-e5-large-instruct	0,599	0,530	0,161	0,036	0,107
bge-m3	0,922	0,937	0,931	0,904	0,931
e5-large-v2	0,922	0,937	0,931	0,904	0,931
nomic-embed-text	0,922	0,937	0,931	0,904	0,931
gte-large	0,594	0,549	0,374	0,065	0,181
all-mpnet-base-v2	0,526	0,436	0,265	0,054	0,142
all-MiniLM-L6-v2	0,399	0,392	0,331	0,077	0,140
retriever = MMR					
multilingual-e5-large-instruct	0,599	0,530	0,161	0,036	0,107
bge-m3	0,922	0,937	0,931	0,904	0,931
e5-large-v2	0,922	0,937	0,931	0,904	0,931
nomic-embed-text	0,922	0,937	0,931	0,904	0,931
gte-large	0,594	0,549	0,374	0,065	0,181
all-mpnet-base-v2	0,557	0,569	0,395	0,08	0,237
all-MiniLM-L6-v2	0,599	0,530	0,161	0,036	0,107
similarity search + LLM					
multilingual-e5-large-instruct	0,720	0,465	0,224	0,045	0,119
bge-m3	0,974	0,881	0,676	0,467	0,619
e5-large-v2	0,973	0,907	0,667	0,451	0,528
nomic-embed-text	0,967	0,918	0,662	0,467	0,561
gte-large	0,683	0,564	0,338	0,075	0,206
all-mpnet-base-v2	0,612	0,492	0,26	0,047	0,153
all-MiniLM-L6-v2	0,632	0,516	0,258	0,031	0,167

Значения оценок в зависимости от используемой векторной базы для модели эмбеддингов BAAI/bge-m3, размера чанка 500 и разных способов поиска релевантной информации приведены в табл. 3.

Анализ данных табл. 3 позволяет сделать вывод о том, что для модели эмбеддингов BAAI/bge-m3 значения оценок практически не зависят от используемой векторной базы. Аналогичные результаты наблюдались и для моделей эмбеддингов intfloat/e5-large-v2 и nomic-embed-text.

Дополнительно проводилось сравнение трёх подходов поиска релевантной информации. В простом режиме similarity search система возвращала наиболее близкие по векторной дистанции фрагменты текста без дополнительной фильтрации, что обеспечивало высокую скорость выполнения запросов. Применение retriever-механизма с алгоритмом MMR для моделей эмбеддингов BAAI/bge-m3, intfloat/e5-large-v2 и nomic-embed-text давало аналогичные значения оценок. При использовании подхода similarity search + LLM реранкер позволял получить наилучшие значения оценок, кросс-энкодер давал оценки несколько ниже, чем для двух предыдущих способов поиска релевантной информации, а метрики ROUGE для моделей эмбеддингов BAAI/bge-m3, intfloat/e5-large-v2 и nomic-embed-text показывали самые низкие оценки. Следовательно, применение LLM не гарантирует высокую степень совпадения между ответом модели и эталонным текстом, но обеспечивает высокую степень содержательной точности ответа.

Таблица 3

Значения оценок в зависимости от векторной базы

Векторная база	Reranker	CrossEncoder	ROUGE-1	ROUGE-2	ROUGE-3
similarity search					
Chroma	0,922	0,937	0,931	0,904	0,931
Qdrant	0,922	0,937	0,931	0,904	0,931
SKLearn	0,922	0,937	0,931	0,904	0,931
retriever = MMR					
Chroma	0,922	0,937	0,931	0,904	0,931
Qdrant	0,922	0,937	0,931	0,904	0,931
SKLearn	0,922	0,937	0,931	0,904	0,931
similarity search + LLM					
Chroma	0,974	0,881	0,676	0,467	0,619
Qdrant	0,974	0,889	0,642	0,415	0,584
SKLearn	0,960	0,864	0,647	0,463	0,529

Заключение

В ходе проведенных экспериментов были протестированы различные конфигурации RAG-системы с варьированием ее ключевых параметров — размера чанка, типа векторных баз, моделей эмбеддингов и способа поиска релевантной информации. Анализ результатов экспериментов позволяет сделать вывод, что качество ответов RAG-системы существенно зависит от размера чанка и выбора модели эмбеддингов.

Наилучшие результаты были получены при использовании размера чанка 500 символов, моделей эмбеддингов BAAI/bge-m3, intfloat/e5-large-v2, nomic-embed-text и любой из рассмотренных в исследовании векторных баз. Варианты конфигурации RAG-системы с использованием данных параметров показали максимально высокие значения метрик, что свидетельствует о высокой степени смыслового совпадения между ответом модели и эталонным текстом. Использование LLM после этапа поиска привело к росту показателя RerankerScore и улучшению качества ответов, несмотря на заметное снижение метрик CrossEncoderScore и ROUGE.

Таким образом, отмеченные комбинации размера чанка, моделей эмбеддингов и векторных баз признаны наиболее сбалансированными по точности и качеству генерируемых ответов.

Литература

1. Gaoa Y. Retrieval-augmented generation for large language models: A survey / Y. Gaoa, Y. Xiongb, X. Gao [et al.] // arXiv:2312.10997v5. – 2024. – P. 1–21.
2. Lewis P. Retrieval-augmented generation for knowledge-intensive NLP tasks / P. Lewis, E. Perez, A. Piktus [et al.] // 34th International Conference on Neural Information Processing Systems. – 2020. – P. 9459–9474.
3. Каширина И. Л. Разработка и оценка RAG-системы для анализа семантических связей / И. Л. Каширина, И. Р. Осипов, В. А. Яковлев // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2025. – № 2. – С. 114–126. – DOI 10.17308/sait/1995-5499/2025/2/114-126.
4. Huang L. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions / L. Huang, W. Yu, W. Ma // arXiv:2311.05232v2. – 2024. – P. 1–58.

5. Wang X. Searching for Best Practices in Retrieval-Augmented Generation / X. Wang, Z. Wang, X. Gao [et al.] // arXiv:2407.01219v1. – 2024. – P. 1–22.
6. Qu R. Is Semantic Chunking Worth the Computational Cost? / R. Qu, F. Bao, R. Tu // arXiv:2410.13070v1. – 2024. – P. 1–21.
7. Topsakal O. Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast / O. Topsakal, T.C. Akinci // 5th International Conference on Applied Engineering and Natural Sciences. – 2023. – P. 1–7.
8. MTEB Leaderboard – a Hugging Face Space by mteb. – URL: <https://huggingface.co/spaces/mteb/leaderboard> (дата обращения: 15.10.2025)
9. Chin-Yew L. ROUGE: A Package for Automatic Evaluation of Summaries / L. Chin-Yew // Workshop on Text Summarization Branches Out. – 2004. – P. 74–81.

РАЗРАБОТКА ИНТЕРВАЛЬНЫХ СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ С НЕОПРЕДЕЛЕННОСТЬЮ

А. П. Гонец, П. В. Сараев

МИРЭА – Российский технологический университет

Аннотация. В данной работе предложено решение для обработки изображений с интервальной неопределенностью. Описана базовая структура сверточной нейронной сети, в частности архитектура LeNet, представлены основные операции подвыборки, функции активации. Приведены основы интервального анализа с базовыми характеристиками и операциями интервальной арифметики. Предложена новая архитектура – интервальные сверточные нейронные сети – для обработки изображений с интервальной неопределенностью, представлены варианты реализации, обоснован выбор интервальных операций с данными. Показаны результаты вычислительных экспериментов, демонстрирующие перспективы предложенных решений.

Ключевые слова: интервальные сверточные нейронные сети, компьютерное зрение, LeNet, интервальный анализ.

Введение

В современном мире автоматизированная обработка изображений является важной частью для ускорения выполнения задач, уменьшения вероятности человеческой ошибки и улучшения обнаружения скрытых закономерностей. Бурное развитие технологий компьютерного зрения началось с разработки архитектуры сверточных нейронных сетей в конце прошлого века, которые в настоящий момент остаются эффективным решением для задач различного уровня сложности обработки изображений.

При наличии зашумленности в данных по причинам особенностей их передачи по сети или первичной обработки высока вероятность, что при восстановлении качество изображений может быть сильно ухудшено. Это усложняет получение достоверных выводов об информации, имеющейся на изображениях. Таким образом, в данных появляется некая неопределенность, которая не отслеживается при восстановлении. Данная проблема обуславливает необходимость разработки новой архитектуры нейронных сетей на основе сверточных, которая будет учитывать входную неопределенность для получения интервальной оценки классификации.

1. Инструменты решения проблемы

1.1. Сверточная нейронная сеть

В 1998 году благодаря улучшению качества наборов данных для обучения моделей Ян Ленкун разработал архитектуру LeNet-5, обладающую высокую точность для распознавания изображений [1]. Данная архитектура заложила фундамент для современных сверточных нейронных сетей. Многие из ее основных принципов используются до сих пор в исходном или усовершенствованном виде. В рамках текущего исследования данная архитектура взята за основу благодаря своей простоте и наличию основных видов операций с данными.

Сверточные нейронные сети позволяют выделять важные объекты с изображений при минимальной предварительной обработке данных. Сами изображения представляют из себя двумерный массив вещественных или целочисленных чисел, которые отвечают за характеристики каждого пикселя. На рис. 1 представлена типовая структура сверточной нейронной

сети LeNet [1]. В оригинальной архитектуре на входе используется одноканальное изображение размером 32×32 пикселя, которое сначала обрабатывается путем использования двух последовательных сверток (convolution) с подвыборками (subsampling), а затем с помощью трех полно связных слоев (full connection), где последний позволяет получить значения для десяти классов — изображение относится к тому классу, у которого выходное значение соответствующего нейрона больше. При этом функцией активации для каждого слоя первоначально являлся гиперболический тангенс, типом подвыборки являлся усредненный пулинг, а функцией потерь была среднеквадратичная ошибка.

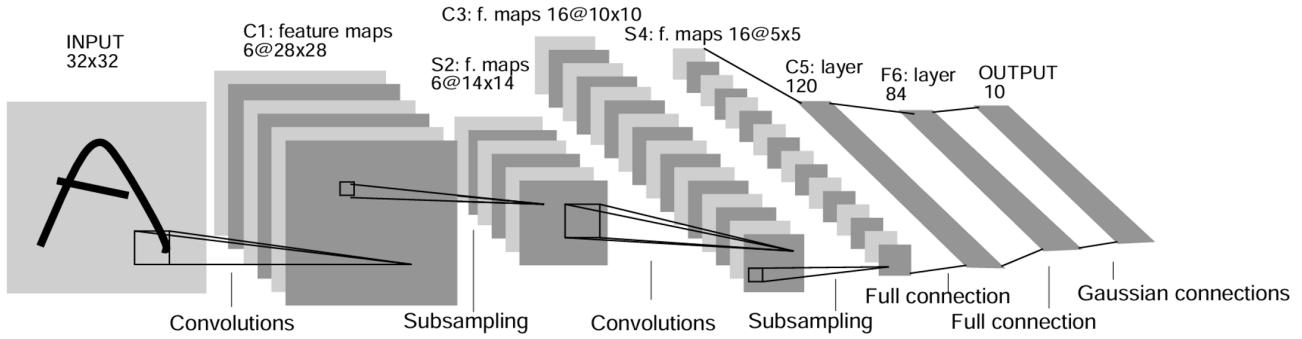


Рис. 1. Архитектура сверточной нейронной сети LeNet

За прошедшее время операции и функции, которые применяются в сверточных нейронных сетях, заметно модифицировались. На рис. 2 представлены основные функции активации, которые могут использоваться после сверточных или внутренних полно связных слоев.

Формулы ранее представленных функций активаций на рис. 2.

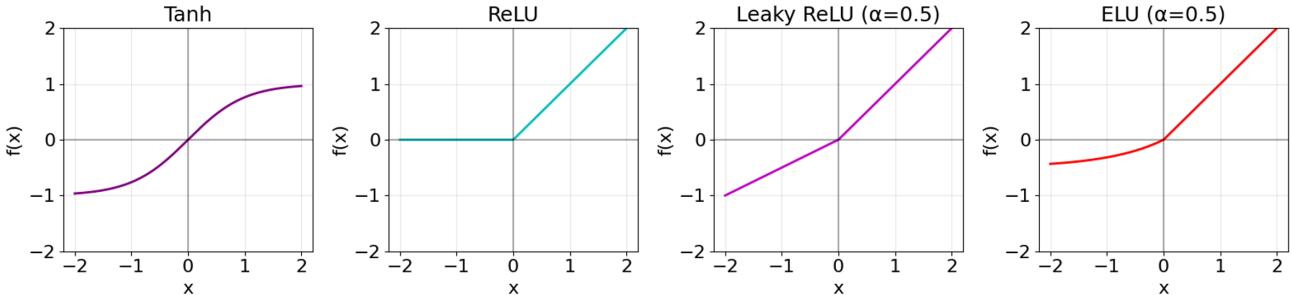


Рис. 2. Функции активации и их графики

Основными видами подвыборок являются выбор среднего значения окна и выбор максимального значения окна. Вместо среднеквадратичной ошибки (MSE) чаще используется в качестве функции потерь кросс-энтропия, показывающая различие между двумя вероятностными распределениями:

$$\text{CrossEntropy}(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i),$$

где y_i — действительная принадлежность объекта к классу i , \hat{y}_i — предсказанная вероятность принадлежности объекта к классу i .

Вышеуказанные элементы могут быть добавлены в оригинальную архитектуру LeNet. Для этого необходимо будет провести экспериментальные вычисления с различными комбинациями для выявления оптимальных параметров.

1.2. Элементы интервального анализа

Интервальная неопределенность возникает при частичном знании о величине, которая характеризуется своими нижней и верхней границами [2]. В отличие от описаний с помощью не-

ограниченных множеств, приводящих к потере содержательности при операциях, интервалы обеспечивают управляемость вычислений. Их преимуществом является простота представления, так как для задания одномерного интервала достаточно двух чисел (нижняя и верхняя граница, либо середина и радиус интервала), что делает их менее сложными, чем вероятностные распределения или нечеткие множества. По своей сути, интервал является простым подмножеством вещественных чисел, определяющим диапазон возможных значений:

$$\mathbf{a} = [\underline{a}, \bar{a}],$$

где \underline{a} — нижняя граница интервала \mathbf{a} , \bar{a} — верхняя граница интервала \mathbf{a} . Основными характеристиками интервала являются его середина и радиус:

$$\text{mid } \mathbf{a} = (\bar{a} + \underline{a})/2,$$

$$\text{rad } \mathbf{a} = (\bar{a} - \underline{a})/2.$$

Различные свойства данных характеристик позволяют находить середину и радиус производного интервала от суммы характеристик интервалов и произведения интервала на число. Эти свойства крайне важны для операций, связанных с выбором среднего значения в подвыборке. Свойства этих характеристик:

$$\text{mid}(\mathbf{a} \pm \mathbf{b}) = \text{mid } \mathbf{a} \pm \text{mid } \mathbf{b},$$

$$\text{rad}(\mathbf{a} \pm \mathbf{b}) = \text{rad } \mathbf{a} + \text{rad } \mathbf{b},$$

$$\text{mid}(\mathbf{a} \cdot \mathbf{b}) = \text{mid } \mathbf{a} \cdot \text{mid } \mathbf{b},$$

$$\text{rad}(\mathbf{a} \cdot \mathbf{b}) = |\mathbf{a}| \cdot \text{rad } \mathbf{b}.$$

Интервальная арифметика делится на несколько типов. Наиболее часто используемой является классическая интервальная арифметика, где базовые арифметические операции, а именно сложение, вычитание, умножение и деление выглядят следующим образом:

$$\mathbf{a} + \mathbf{b} = [\underline{a} + \underline{b}, \bar{a} + \bar{b}],$$

$$\mathbf{a} - \mathbf{b} = [\underline{a} - \bar{b}, \bar{a} - \underline{b}],$$

$$\mathbf{a} \cdot \mathbf{b} = \left[\min \{ \underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b} \}, \max \{ \underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b} \} \right],$$

$$\mathbf{a} / \mathbf{b} = \mathbf{a} \cdot \left[\frac{1}{\bar{b}}, \frac{1}{\underline{b}} \right] \text{ для } 0 \notin \mathbf{b}.$$

Операция деления является наиболее сложной, так как при делении интервал не может содержать нуль [3]. Для разрешения этой проблемы существует несколько подходов, например, интервальная арифметика Кэхэна, где используются бесконечные или полубесконечные интервалы [4]. Важный частный случай интервального умножения соответствует произведению числа на интервал:

$$\mathbf{a} \cdot \mathbf{b} = \begin{cases} [\underline{a}\underline{b}, \bar{a}\bar{b}], & a \geq 0, \\ [\bar{a}\underline{b}, \underline{a}\bar{b}], & a < 0. \end{cases}$$

Преобразование интервалов фактически сводятся к вычислению образов несложных множеств при различных отображениях. Для любой непрерывной функции справедливо равенство, что область значений функции для интервального аргумента является интервалом от ее минимума до ее максимума:

$$\text{ran}(f, X) = \left[\min_{x \in X} f(x), \max_{x \in X} f(x) \right].$$

Таким образом, при использовании монотонно возрастающих функций активаций значение минимума и максимума будут достигаться в точках границ интервала. При применении немонотонных функций активаций минимум функции может находиться не на границах интервала. Если удается разбить область определения функции на участки монотонности, то вычисление образа функции остается достаточно простой операцией.

Если интервалы \mathbf{a} и \mathbf{b} имеют непустое пересечение, то можно дать простые выражения для результатов теоретико-множественных операций пересечения и объединения через концы этих интервалов:

$$\mathbf{a} \cap \mathbf{b} = \left[\max \{ \underline{a}, \underline{b} \}, \min \{ \bar{a}, \bar{b} \} \right],$$

$$\mathbf{a} \cup \mathbf{b} = \left[\min \{ \underline{a}, \underline{b} \}, \max \{ \bar{a}, \bar{b} \} \right].$$

При появлении неправильных интервалов, что обычно бывает при операции пересечении непересекающихся интервалов без проверки на наличие общих точек, на помощь приходит арифметика Каухера. Правильные и неправильные интервалы переходят друг в друга в результате отображения дуализации:

$$\text{dual } \mathbf{a} := \left[\bar{a}, \underline{a} \right].$$

В таком случае, если интервал правильный, то его проекция соответствует исходному интервалу, иначе применяется дуализация:

$$\text{про } \mathbf{a} = \begin{cases} \mathbf{a}, & \mathbf{a} \text{ правильный}, \\ \text{dual } \mathbf{a}, & \mathbf{a} \text{ неправильный}. \end{cases}$$

Приведенные свойства и характеристики интервалов позволяют выполнять операции, используемые в сверточных нейронных сетях.

2. Архитектура интервальных сверточных нейронных сетей

Интеграция элементов интервального анализа в сверточную нейронную сеть позволяет обеспечить достоверную оценку классовой принадлежности того или иного изображения с указанием границ оценки. Помимо интервального представления входного изображения, также возможно использование интервального типа данных для весов модели на разных уровнях. Это может способствовать увеличению расхождению границ весов в процессе обучения и нахождения зоны глобального минимума нелинейной функции нейронной сети. Следует выделить несколько этапов реализации — сверточный слой, полносвязный слой, активация, подвыборка, получение вероятности принадлежности, обучение модели.

Операция свертки представляет умножение весов ядра свертки с областью входного изображения. Таким образом, данную операцию можно реализовать как умножение интервалов друг с другом ($\mathbf{a} \cdot \mathbf{b}$) при интервальных весах или умножение значения веса с интервалом ($\mathbf{a} \cdot \mathbf{b}$), и с дальнейшим сложением полученных интервалов ($\mathbf{c} + \mathbf{d}$). Для полносвязного слоя также могут применяться операции интервального умножения и сложения.

Функции активации, представленные на рис. 2, являются монотонно возрастающими, благодаря чему нижняя и верхняя границы выходных значений функций соответствуют значениям на крайних границах интервала. Применение функции ReLU, которая является востребованной в нейронных сетях, может оказаться негативно из-за зануления всех отрицательных значений, поскольку нижняя граница интервалов будет часто отрицательной, что приведет к постоянному обнулению нижней границы результата применения функции активации. Оптимальной функцией активации в таком случае может являться ELU благодаря наличию нелинейности в области отрицательных значений.

В случае применения подвыборок наиболее простыми для представления являются операции выбора среднего и выбора максимума окна. Если представлять операцию выбора среднего как выбор среднего для нижней и верхней границы, то это равнозначно нахождению границ интервала через середину и радиус производного интервала от суммы характеристик интервалов и произведения интервала на число, где число будет соответствовать количеству исходных интервалов. Операция выбора максимума окна является выбором интервала с максимальным верхним значением и нахождения снизу точки пересечения по этому интервалу (попарное пересечение выбранного интервала с остальными), что будет также равносильно нахождению максимумов по нижним и верхним границам всех интервалов:

$$\max \left\{ \overline{\underline{a}_0}, \dots, \overline{\underline{a}_n} \right\} \in \underline{b},$$

$$\left[\max \left\{ \max \left\{ \underline{b}, \underline{a}_0 \right\}, \dots, \max \left\{ \underline{b}, \underline{a}_n \right\} \right\}, \max \left\{ \overline{\underline{a}_0}, \dots, \overline{\underline{a}_n} \right\} \right] \equiv \left[\max \left\{ \underline{a}_0, \dots, \underline{a}_n \right\}, \max \left\{ \overline{\underline{a}_0}, \dots, \overline{\underline{a}_n} \right\} \right].$$

Для обучения модели с помощью функции потерь кросс-энтропии необходимо определить методику штрафов. Расчет потерь можно производить на основе результатов по средним значениям каждого интервала, так и по нижним и верхним границам интервала.

Таким образом, предлагаемое изменение в структуре архитектуры LeNet путем замены вещественных значений на интервальные с приведенными в данном пункте операциями подвыборки, функциями активации и потерь, а также подходами к образованию сверточных и полно связанных слоев позволяют решать задачи с входной неопределенностью. Данное решение представляет собой архитектуру интервальных сверточных нейронных сетей.

3. Вычислительные эксперименты

В ходе проверки реализованной модели с различными параметрами, к которым относятся функции активации, подвыборки, применяются инициализированные веса на основе обучения оригинальной модели на наборе данных MNIST, который содержит одноканальные изображения рукописных цифр до нуля до девяти [5]. Дообучение модели и сравнение результатов с оригинальной выполнено на наборе данных Fashion MNIST, содержащий одноканальные изображения различных видов одежды [6]. Набор данных разделен на три части — тренировочный из 60000 изображений, валидационный и тестовый по 10000 изображений каждый. На рис. 3 представлен фрагмент набора данных, где верхние изображения соответствуют нижним границам входного изображения, а нижние изображения — верхним границам. В табл. 1 представлен фрагмент подматрицы размера 5×5 , состоящей из интервальных данных.

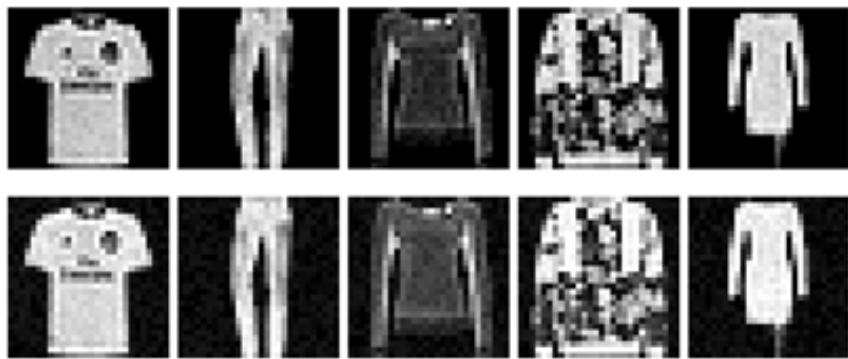


Рис. 3. Границные изображения

Ввиду того, что изображения имеют размер 28×28 пикселей, необходимым условием для работы с моделью является добавление смещения при свертке (паддинг) на два пикселя. Параметры обучения представлены в табл. 2.

Таблица 1

Пример интервальной подматрицы

[14, 28]	[126, 157]	[12, 45]	[10, 43]	[23, 54]
[30, 60]	[40, 70]	[54, 79]	[28, 63]	[0, 20]
[23, 53]	[54, 87]	[123, 152]	[65, 89]	[124, 157]
[57, 92]	[255, 255]	[12, 32]	[234, 250]	[3, 7]
[87, 101]	[200, 206]	[56, 80]	[50, 60]	[1, 23]
[98, 98]	[54, 68]	[90, 110]	[79, 90]	[0, 0]

Таблица 2

Параметры обучения модели

Параметр	Значение
Оптимизатор	Adam
Максимальное количество эпох	30
Стартовый шаг обучения	0,001
Метод снижения шага обучения	Выход на плато
Количество эпох на плато	10
Минимальный шаг обучения	0,00001
Коэффициент снижения шага обучения	0,1
Максимальное количество последовательных эпох с превышением минимальной валидации на 10 %	3

Эксперимент работоспособности решения проводился на нескольких вариациях модели с дополнительным сравнением с исходной моделью. Шум в данных для обучения и тестирования может достигать до 10 % от максимального значения каждого пикселя изображения. Настраиваемыми параметрами являются функция активации — гиперболический тангенс Tanh, ReLU, ELU с коэффициентом нелинейности 1,0, LeakyReLU с коэффициентом угла наклона 0,01; способ подвыборки в окне — среднее значение Avg или максимальное значение Max; расчет функции потерь кросс-энтропии — по середине интервала Mid, по границам интервала Borders, комбинация из суммы потерь по середине и границам интервала Mixed. Также предусмотрено обучение с интервальными весами на последнем полно связном слое, значения которых различаются на 0,002 между собой при инициализации весов. На рис. 4 представлен процесс обучения модели с интервальными весами. Оценка точности при обучении учитывалась по вычисленным значениям середин интервалов.

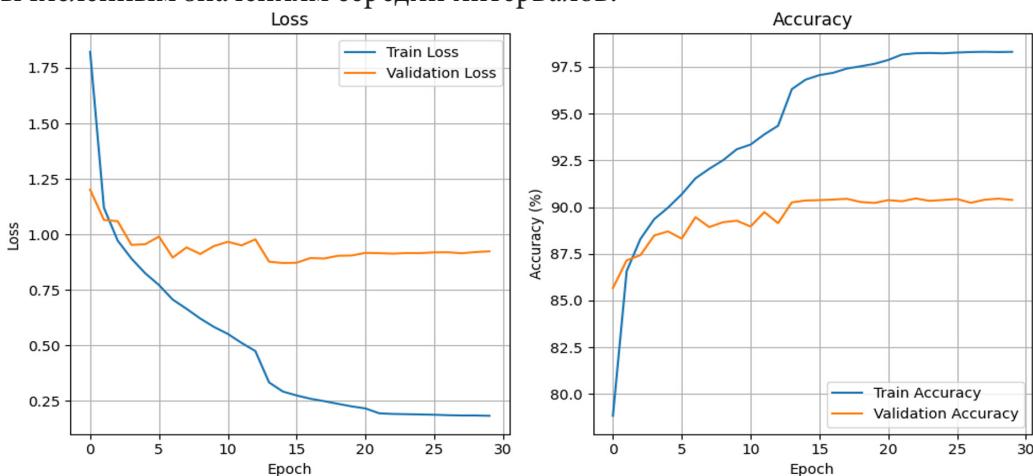


Рис. 4. Графики функции потерь и точности при обучении модели с интервальными весами

Результаты экспериментов представлены в табл. 3. Точность моделей на тестовом наборе данных определялась на основе вычисленных значений в тех точках интервалов — на нижней границе, в середине и на верхней границе. Можно сделать вывод, что выдвинутая гипотеза о высоком качестве модели с функцией активации ELU оказалась недостаточно верной, так как для интервальных данных наилучшие результаты продемонстрировал гиперболический тангенс. Наилучшим способом подвыборки при свертки оказалась операция выбора максимума из окна свертки. Штрафы модели за ложные ответы по середине интервала совместно со штрафами по границам интервала продемонстрировали наилучшие показатели за счет контроля за шириной интервала.

Таблица 3

Эксперименты интервальной сверточной нейронной сети

Тип весов	Подвыборка	Функция активации	Данные для обучения	Параметр функции потерь	Кол-во эпох	Точность обучения (в середине интервала), %	Точность тестирования (на нижней границе; в середине; на верхней границе), %
Float	Avg	Tanh	Исходные	–	30	95,7	64,9; 74,1; 65,1
Float	Avg	Tanh	С шумом	Mid	30	96,5	65,6; 72,1; 61,3
Float	Avg	ReLU	С шумом	Mid	30	95,1	18,8; 45,0; 21,2
Float	Avg	ELU, $\alpha = 1,0$	С шумом	Mid	30	96,1	37,6; 53,8; 40,2
Float	Avg	LeakyReLU, $\alpha = 0,01$	С шумом	Mid	30	95,6	10,0; 39,2; 31,4
Float	Max	Tanh	С шумом	Mid	30	96,6	88,9; 91,0; 89,0
Float	Max	ReLU	С шумом	Mid	30	96,7	26,5; 51,9; 22,3
Float	Max	ELU, $\alpha = 1,0$	С шумом	Mid	25	96,7	40,5; 61,8; 31,4
Float	Max	LeakyReLU, $\alpha = 0,01$	С шумом	Mid	18	93,9	11,8; 58,9; 28,8
Float	Avg	Tanh	С шумом	Borders	30	95,9	75,8; 76,1; 76,1
Float	Avg	ELU, $\alpha = 1,0$	С шумом	Borders	30	95,9	55,0; 54,5; 53,4
Float	Max	Tanh	С шумом	Borders	20	97,4	90,8; 90,8; 90,7
Float	Max	ELU, $\alpha = 1,0$	С шумом	Borders	18	96,7	66,0; 65,4; 63,7
Float	Max	Tanh	С шумом	Mixed	30	97,2	90,7; 90,7; 90,7
Float	Max	ELU, $\alpha = 1,0$	С шумом	Mixed	19	97,0	68,5; 68,4; 67,2
Interval, fc3	Max	Tanh	С шумом	Mixed	20	97,4	90,8; 90,7; 90,6
Interval, all	Max	Tanh	С шумом	Mixed	30	98,3	91,1; 91,2; 90,9

Применение интервальных весов позволило немного улучшить результаты за счет поиска оптимальных значений в определенном узком интервале весов модели, но при данном подходе для модели со всеми интервальными весами вдвое увеличилось время обучения. С полными результатами можно ознакомиться в источнике [7], где расположены исходные наборы данных, веса обученных моделей, матрицы ошибок, графики обучения и файлы с параметрами обучения.

Заключение

В статье предложена архитектура интервальной сверточной нейронной сети на основе архитектуры LeNet, обученной на наборе данных Fashion-MNIST. Применение элементов интервального анализа позволило решать задачи с входной неопределенностью в изображениях,

представляя пиксели в виде интервальных данных. Были разобраны основные функции активаций и типы подвыборок в сверточных нейронных сетях, которые охватывают базисные аспекты при построении нейросетей данного класса. Экспериментально продемонстрировано, что разработанный инструмент для работы с входной неопределенностью в изображениях позволяет решать поставленную задачу классификации.

В рамках дальнейших исследований запланировано определения диапазона уверенности решения по выбору класса, дополнение набора методов подвыборок, а также добавление регуляризации при обучении.

Литература

1. *LeCun Y. Gradient-Based Learning Applied to Document Recognition* / Y. LeCun, L. Bottou, J. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – Vol. 86, № 11. – P. 2278–2324. – URL: <https://doi.org/10.1109/5.726791> (дата обращения: 20.10.2025).
2. Обработка и анализ интервальных данных / А. Н. Баженов, С. И. Жилин, С. И. Кумков, С.П. Шарый. – М.; Ижевск : Институт компьютерных исследований, 2024. – 356 с.
3. *Moore Ramon E. Introduction to interval analysis* / Ramon E. Moore, R. Baker Kearfott, Michail J. Cloud. – Philadelphia, USA : SIAM. – 223 p.
4. *Laveuve S. E. Definition einer Kahan-Arithmetik und ihre Implementierung* / S. E. Laveuve // Interval Mathematics. – Berlin : Springer Verlag, 1975. P. 236-245. – URL: https://doi.org/10.1007/3-540-07170-9_23 (дата обращения: 29.10.2025).
5. Репозиторий с весами обученной модели LeNet на наборе данных MNIST. – URL: https://github.com/Jimmy1634/LeNet_MNIST/tree/main/Weights (дата обращения: 31.10.2025).
6. Набор данных Fashion MNIST. – URL: <https://www.kaggle.com/datasets/zalando-research/fashionmnist> (дата обращения: 01.11.2025).
7. Результаты обучения и тестирования модели ICNN_LeNet. – URL: <https://disk.yandex.ru/d/j8L20R4yUaOVIA> (дата обращения: 05.11.2025).

ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ДЛЯ АНАЛИЗА ОБЪЕКТОВ В «ПОЛЕВЫХ» УСЛОВИЯХ

А. С. Гончаров

МИРЭА – Российский технологический университет

Аннотация. В работе рассматриваются подходы к созданию систем анализа объектов с использованием нейросетевых моделей в условиях ограниченных вычислительных и сетевых ресурсов («полевые» условия). Проведён обзор архитектурных решений и предложены четыре варианта построения систем инференса. Особое внимание уделено реализации архитектуры с интеграцией нейросетевой модели непосредственно в мобильное приложение, включая описание используемых программных средств, обработки, хранения и передачи данных. Кроме того, рассмотрены направления дальнейшего развития решения.

Ключевые слова: нейросетевые модели, компьютерное зрение, мобильное приложение, инференс, автономность, офлайн-инференс, реальное время, инспекция оборудования, контроль состояния, носимые устройства, дополненная реальность, TensorFlow Lite, PyTorch, YOLO, Android.

Введение

Во всех отраслях, где используется сложное оборудование — в добывающей и перерабатывающей промышленности, энергетике и нефтегазовом секторе — одной из ключевых задач является постоянный контроль его технического состояния [1]. Для своевременного выявления отклонений производственных показателей от нормы и предупреждения аварийных ситуаций на предприятиях регулярно проводятся осмотры и дефектоскопические исследования. Однако традиционные методы инспекции, основанные на визуальном контроле и ручном заполнении форм, отличаются высокой трудоёмкостью, субъективностью оценок и риском человеческих ошибок.

Нейросетевые модели компьютерного зрения (CV) находят широкое применение в автоматизации контроля технического состояния оборудования. С их помощью можно решать задачи по определению комплектности устройств, обнаружению дефектов — таких как коррозия, трещины или механические повреждения, — а также по оценке состояния оборудования в реальном времени. Подобные системы способны выявлять аномалии с точностью, превосходящей возможности человека, особенно в условиях усталости, недостаточного освещения или ограниченного времени инспекции.

Современные технологии переходят к интеграции инференса непосредственно в носимые устройства — такие как AR-очки, защищённые планшеты и смарт-камеры, — где камера, вычислительный модуль и интерфейс объединены в компактный корпус. Эти решения способны работать полностью автономно, обеспечивая офлайн-инференс даже в сложных условиях — в шахте, на вышке или в пустыне — без подключения к облаку и с мгновенным AR-наложением результатов прямо в поле зрения пользователя.

При этом в «полевых» условиях — на удалённых подстанциях, в подземных коммуникациях, на высотных опорах ЛЭП или в зонах с нестабильной связью — возникают факторы, существенно влияющие на архитектуру решений:

- **Отсутствие или нестабильность сети** делает невозможным использование облачных вычислений.

- **Ограниченност ресурсов устройства:** ограничены вычислительная мощность, объём памяти и запас энергии; батареи должно хватать на 4–8 ч, тепловыделение — минимальное.

- **Время отклика:** результат должен формироваться в реальном времени, задержка более 200 мс недопустима при работе на высоте или в движении.

- **Эргономика и безопасность эксплуатации:** масса дополнительного снаряжения — не более 500 г; устройство не должно мешать защитной экипировке (каске, СИЗ) и отвлекать пользователя от окружающей среды.

- **Конфиденциальность данных:** передача изображений по открытым каналам недопустима в соответствии с требованиями информационной безопасности.

Эти ограничения требуют разработки полностью автономных решений, интегрированных непосредственно в устройства пользователя, чтобы инференс выполнялся локально, результаты формировались с минимальной задержкой, а данные оставались защищёнными и передавались только по безопасным каналам [2].

1. Варианты архитектуры

Современные системы анализа изображений можно классифицировать по способу размещения модели и выполнению инференса. В табл. 1 представлены основные варианты архитектурных решений, их преимущества, ограничения и типичные области применения.

Таблица 1

Преимущества и недостатки анализируемых вариантов архитектуры

№	Архитектура	Преимущества	Недостатки	Применение
1	Облачный инференс через интернет/ Wi-Fi	<ul style="list-style-type: none"> • Доступ к мощным GPU • Возможность использования крупных моделей • Централизованное обновление моделей 	<ul style="list-style-type: none"> • Зависимость от стабильности сети • Задержки при передаче данных • Расходы на трафик и инфраструктуру (сетевое оборудование) 	<ul style="list-style-type: none"> • Стационарные объекты с гарантированным доступом к сети
2	Локальный инференс-сервер (Rockchip, NVIDIA Jetson и аналоги)	<ul style="list-style-type: none"> • Автономность • Поддержка моделей среднего размера (в зависимости от вычислительной мощности) • Повышенная безопасность данных 	<ul style="list-style-type: none"> • Необходимость дополнительного оборудования • Сложность логистики • Ограниченная мобильность 	<ul style="list-style-type: none"> • Мобильные комплексы, дроны, робототехника
3	Инференс «на борту» устройства (смартфон, планшет, AR-очки)	<ul style="list-style-type: none"> • Полная автономность • Минимальная задержка • Отсутствие передачи данных • Простота эксплуатации 	<ul style="list-style-type: none"> • Ограниченный размер моделей (обычно < 100 МБ) • Зависимость от аппаратных ресурсов 	<ul style="list-style-type: none"> • Индивидуальные инспекции в зонах без связи
4	Отложенный инференс (анализ после съёмки)	<ul style="list-style-type: none"> • Отсутствие требований к работе в реальном времени • Возможность использования крупных моделей • Минимальное энергопотребление при съёмке 	<ul style="list-style-type: none"> • Отсутствие мгновенного результата • Необходимость хранения больших объёмов данных • Риск потери носителя • Нет возможности оперативного реагирования 	<ul style="list-style-type: none"> • Периодические обходы с обработкой на базе • Пост-анализ инцидентов

На рис. 1 схематично показаны рассмотренные варианты архитектуры.

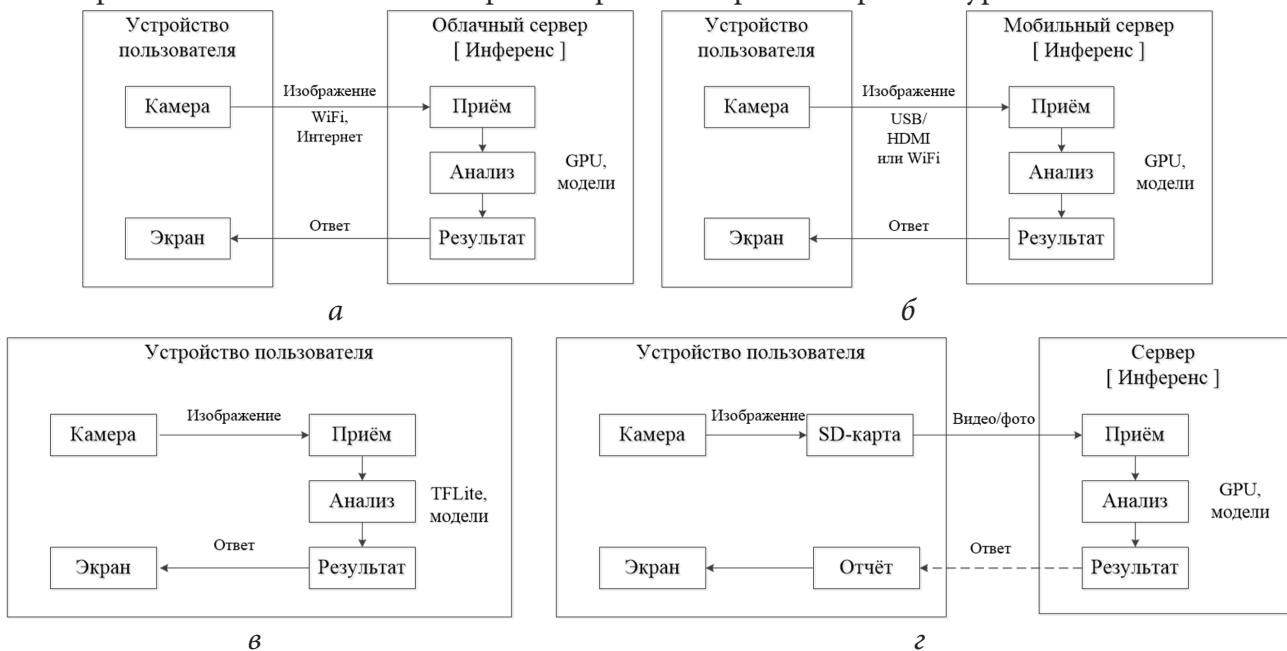


Рис. 1. Варианты архитектуры:

а — Облачный инференс через интернет/Wi-Fi, б — Локальный инференс-сервер,
в — Инференс «на борту» устройства, г — Отложенный инференс

С учётом необходимости полной автономности, минимальной задержки, защиты данных и работы в условиях отсутствия или нестабильной сети, **был выбран вариант 3 — инференс «на борту» устройства**. Этот подход позволяет выполнять анализ непосредственно на смартфоне, планшете или AR-очках, обеспечивая мгновенную обратную связь и независимость от облачных сервисов, что критично для «полевых» инспекций и индивидуального контроля оборудования.

2. Реализация варианта «на борту»

Рассмотрим реализацию архитектуры с локальным инференсом на примере приложения, разработанного для анализа комплектности и состояния оборудования в реальном времени.

2.1. Архитектура приложения

Архитектура разработанного приложения построена по принципу **клиент-сервер** с чётким разделением функций:

- **клиентская часть** отвечает за сбор данных, локальный инференс и взаимодействие с пользователем;

- **серверная часть** отвечает за централизованное хранение, аналитику и обновление моделей.

Такой подход обеспечивает автономность приложения в полевых условиях, оставляя при этом возможность синхронизации данных при появлении сетевого подключения.

Клиентская часть (Android)

Клиентская часть разработана на языке программирования **Kotlin** с использованием среды разработки **Android Studio**. Захват видеопотока осуществляется через **CameraX API**, обеспечивающий стабильную работу на широком спектре устройств. Навигация по экранам реа-

лизована с помощью **Jetpack Compose** — надёжного и масштабируемого решения от Google. Сетевое взаимодействие выполняется через библиотеку **Retrofit2** в связке с **OkHttp**, что гарантирует эффективную передачу данных при минимальных накладных расходах.

Локальный инференс реализован на базе **TensorFlow Lite** с поддержкой аппаратного ускорения, что критично для производительности на мобильных устройствах.

Хранение данных (результаты инспекций, кэш изображений) организовано через **Room Database** — надстройку над SQLite, обеспечивающую безопасное и структурированное хранение данных в офлайн-режиме.

Серверная часть (Python)

Backend реализован на **FastAPI** — асинхронном высокопроизводительном фреймворке, оптимизированном для обработки большого числа параллельных запросов.

Асинхронная обработка и загрузка данных выполняются с помощью **asyncio**, что позволяет минимизировать задержки при массовых операциях.

Цепочка подготовки моделей включает стадии:

PyTorch (Ultralytics) → ONNX → TensorFlow Lite,

что обеспечивает переносимость и оптимизацию модели под мобильные устройства.

Хранение данных осуществляется в промышленной СУБД **PostgreSQL**, поддерживающей масштабирование, резервное копирование и интеграцию с BI-инструментами.

Обучение моделей выполняется с использованием **Ultralytics YOLOv8**, лидера в задачах детекции объектов в реальном времени.

Таким образом, архитектура сочетает автономность клиентского устройства с гибкостью серверной аналитики, обеспечивая масштабируемость и готовность к интеграции в корпоративные системы мониторинга и технического обслуживания.

2.2. Подготовка модели

Обучение базовой модели

Для задачи детекции комплектности оборудования использовалась облегчённая архитектура **YOLOv8n** (Ultralytics) — оптимальный выбор для мобильных устройств за счёт баланса скорости и точности.

Обучение проводилось на собственном датасете, включающем **500 изображений модулей и антенн**, содержащем 4 класса объектов: **антенна, модуль, комплектный набор, некомплектный набор**.

Основная цель — автоматическое определение комплектности оборудования.

Для повышения обобщающей способности применялись методы аугментации: **Mosaic** (комбинирование 4 изображений в одно), **RandomPerspective** (случайные перспективные искажения), **HSV-шум** (изменение оттенка, насыщенности и яркости) и др.

На валидационной выборке достигнуто значение **mAP@0.5 = 0.94**, что подтверждает высокую точность модели.

Оптимизация модели

Для адаптации к мобильным устройствам выполнено **квантование в формат float16**, что позволило уменьшить размер модели и ускорить инференс без существенной потери точности.

Цепочка конвертации: **PyTorch → ONNX → TensorFlow Lite (float16)**.

Итоговый файл модели (.tflite) встроен в приложение в виде ресурса.

Интеграция в приложение

Модель хранится в папке assets и загружается при запуске приложения.

Для выполнения инференса используется **TensorFlow Lite Interpreter** с аппаратным ускорением, что обеспечивает стабильную работу даже на смартфонах среднего класса.

2.3. Обработка видеопотока

Видеопоток обрабатывается по конвейеру:

захват → предобработка → инференс → визуализация.

Камера через CameraX API формирует данные в формате **ImageFormat.YUV_420_888** — оптимальном для Android (подходит для быстрых преобразований, энергоэффективен).

Каждый кадр масштабируется до **640 × 640 пикселей** (вход YOLOv8n) и нормализуется в диапазон **[0, 1]**. Преобразование **YUV → RGB** выполняется нативно, с минимальной задержкой.

Результаты инференса — рамки, метки классов (антенна, модуль, комплектный, некомплектный) и вероятность (уверенность модели) — отображаются поверх видеопотока с помощью **OverlayView** на базе **Canvas**, обеспечивая мгновенную визуальную обратную связь оператору.

На рис. 2 приведены примеры результатов инференса на обычном смартфоне.

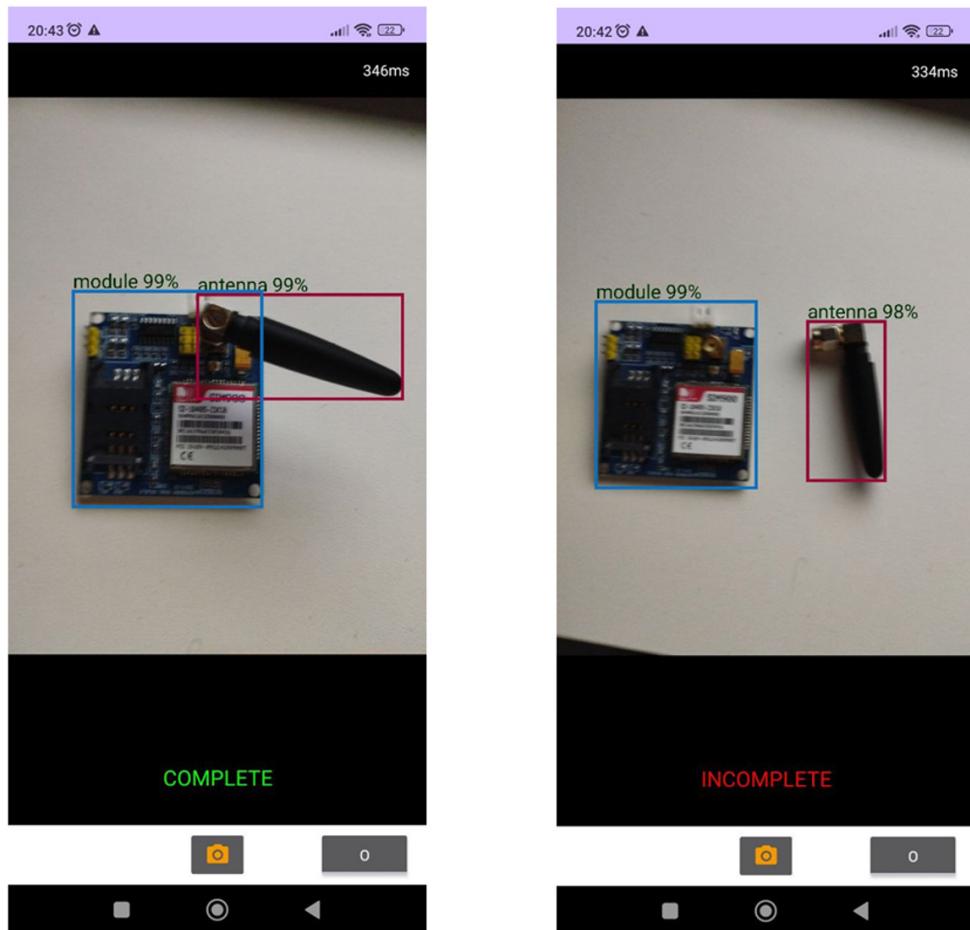


Рис. 2. Пример инференса «на борту» обычного смартфона (без аппаратного ускорителя): слева — детекция комплектного набора, справа — некомплектного

2.4. Сохранение и передача данных

Система обеспечивает **надёжное локальное хранение и автоматическую синхронизацию** данных при восстановлении сети.

Каждый результат анализа сохраняется в **Room Database** в формате **JSON**, содержащем полную информацию об инспекции: фотографию, координаты, уверенность модели и итоговый статус объекта.

При подключении к сети (Wi-Fi, 4G/5G) данные передаются на сервер через **REST API**.

Информация на сервере агрегируется по объектам, датам и исполнителям и используется для формирования отчётов, включающих изображения, графики и статистику.

2.5. Преимущества решения

Разработанное решение сочетает автономность, производительность и простоту эксплуатации, что делает его оптимальным инструментом для полевых инспекций. В табл. 2 приведены ключевые метрики, полученные в ходе тестирования.

Таблица 2

Ключевые метрики решения

Параметр	Значение
Автономность	Полная — работа без интернета, сети и облака
Скорость инференса	5–15 fps (Snapdragon 8 Gen 1)
Размер модели	6 МБ (после квантования и прунинга)
Точность	mAP@0.5 = 0.94
Энергопотребление	+20 % к фоновому режиму при активной съёмке

3. Анализ существующих аналогов

Существующие мобильные решения для технической инспекции можно условно разделить на два класса — **без использования технологий искусственного интеллекта и с их применением, в том числе с элементами дополненной реальности (AR)**.

К первой группе относятся такие системы, как «Мобильные сотрудники» (МТС, <https://agents.mts.ru/about-products/mobilnie-sotrudniki>), «Мобильный инспектор» (ООО «БИТ», <https://bit76.ru/solutions/mobile-inspector>), «Мобильный контролёр» (DBA Group, <https://dba.ooo/solutions/mobile>) и «T-Мобис» (ООО «Траектория времени», <https://timepath.ru/t-mobis-mk/>). Они ориентированы преимущественно на автоматизацию документооборота и организацию производственных осмотров: ведение чек-листов, формуляров, фиксацию фото, а также интеграцию с корпоративными учётными системами. Интеллектуальный анализ изображений или автоматическое распознавание объектов в этих продуктах, как правило, отсутствуют.

Вторая группа включает решения, которые сочетают AR и методы анализа с применением нейросетей. К ним относятся отечественная платформа ИКСАР (ООО «ИКСАР Технологии», РФ, <https://iksar.pro/platform>), решения *Inspect AR* (AR Inspect, Германия, <https://inspect-ar.com>), *Inspect Krank* (Krank Limited, Великобритания, <https://www.krank.com/inspeiq>), *Cendant Inspect Mobile* (Musashi AI, Япония/США, <https://musashiai.com/#cendant>), а также международные AR-платформы: *Vuforia* (PTC, США, <https://developer.vuforia.com/library/vuforia-engine/images-and-objects/model-targets/model-targets/>), *Augmentir* (Augmentir, США, <https://www.augmentir.com/product/augmented-reality>) и *Frontline* (TeamViewer, Германия, <https://frontline.io/mobile/>). Эти системы предоставляют визуализацию подсказок и инструкций, распознавание объектов и дефектов с применением глубокого обучения, поддержку удалённого взаимодействия с экспертами и интеграцию с корпоративными данными. При этом большинство подобных решений полагается на облачные вычисления, что ограничивает их использование при нестабильной или отсутствующей сети.

Разработанное решение реализует **полный онлайн-инференс нейросетевой модели непосредственно на мобильном устройстве**, сохраняя при этом поддержку AR-наложений. Такой подход объединяет преимущества обеих групп: высокую автономность, мгновенную обратную связь и независимость от сетевой инфраструктуры. Это делает систему особенно эффективной в «полевых» условиях — на удалённых подстанциях, в шахтах, на высотных опорах и в других зонах, где традиционные облачные технологии неприемлемы.

4. Дальнейшее развитие

В перспективе планируется развитие системы по нескольким направлениям, связанным с повышением точности анализа, расширением функциональности и улучшением пользовательского опыта.

1. Распознавание положения элементов.

Планируется внедрение анализа ключевых точек (*keypoints*) для оценки положения и состояния таких компонентов, как тумблеры, рукоятки и разъёмы.

Использование архитектур типа **YOLOv8-pose** позволит определять состояния «включено/выключено» или «подключено/отсоединено», обеспечивая более детальную диагностику оборудования.

2. Добавление новых классов.

Планируется расширение набора распознаваемых элементов (например, пломбы, индикаторы, элементы крепления) без необходимости перекомпиляции приложения.

Для этого будет реализован механизм загрузки обновлённых моделей в формате **.tflite** непосредственно с сервера.

3. Оптимизация производительности.

Основная цель — достижение стабильной частоты **10–15 кадров в секунду** на устройствах среднего ценового сегмента за счёт оптимизации вычислительных операций и использования ускорителей (GPU, NPU).

4. Расширение поддержки платформ.

Планируется интеграция с **AR-устройствами**, такими как **RealWear Navigator 500/520**, что позволит использовать систему в режиме дополненной реальности без участия смартфона.

5. Улучшение взаимодействия с пользователем.

Будет реализована поддержка **голосовых команд и подсказок** (на основе технологий *Speech-to-Text* и *Text-to-Speech*), а также **адаптивный интерфейс**, автоматически выделяющий зоны интереса и предлагающий пошаговые инструкции при обнаружении некомплектности.

Заключение

Разработанное программное решение с использованием нейросетевой модели, встроенной в мобильное приложение, обеспечивает **автономную обработку данных и анализ состояния оборудования в реальном времени**. Система надёжно функционирует в условиях отсутствия интернета и Wi-Fi, сохраняя все результаты локально до момента синхронизации.

Ключевые преимущества решения:

- Повышает скорость и удобство осмотра оборудования оператором.
- Снижает вероятность человеческих ошибок.
- Исключает зависимость инференса от сетевого подключения.
- Обеспечивает интеграцию с централизованным хранилищем данных при наличии связи — с возможностью формирования отчётов и аналитики.

В перспективе система может быть дополнена новыми классами объектов, функцией анализа положения элементов оборудования, поддержкой мультимодальных данных (звук, вибрация), а также усовершенствованным пользовательским интерфейсом.

Современные тенденции цифровизации подтверждают актуальность данного подхода: по данным IDC (Q2 2025), мировой рынок носимых устройств демонстрирует рост на **9,6 % год к году**, причём **умные очки становятся ключевым драйвером enterprise-решений** [3]. Это открывает путь к дальнейшему переходу от смартфонов к носимым устройствам с локальными системами искусственного интеллекта.

Литература

1. *Jin C., Li W., Zhang Y. Wearable Device-Based Intelligent Patrol Inspection System Using Smart Glasses and Mixed Reality / C. Jin, W. Li, Y. Zhang // International Journal of Advanced Manufacturing Technology. – 2023. – URL: <https://link.springer.com/article/10.1007/s00170-023-12345-6> (дата обращения: 25.10.2025).*
2. *Heydari S., Mahmoud Q. H. Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges and Future Directions / S. Heydari, Q. H. Mahmoud // PMC. – 2025. – URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1234567/> (дата обращения: 25.10.2025).*
3. *IDC. Wearable Devices Market Insights – Q2 2025: 136.5 million units shipped, +9.6% YoY: пресс-релиз [сайт]. – 2025. – URL: <https://www.idc.com> (дата обращения: 29.10.2025).*

**РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ,
ВКЛЮЧАЮЩЕЙ ТЕХНОЛОГИИ МАШИННОГО ОБУЧЕНИЯ
ДЛЯ РАСПОЗНАВАНИЯ СОРНОЙ РАСТИТЕЛЬНОСТИ НА ПОЛЯХ**

Л. В. Горбенко, А. Ю. Бондаренко, Д. Д. Савлучинская

Воронежский государственный университет

Аннотация. В научной работе рассматривается автоматизированное решение проблемы неконтролируемого распространения сорных растений и инфекционных заболеваний сельскохозяйственных культур в агропромышленном комплексе путём подключения дополнительного оборудования к используемой промышленной технике с внедрённой системой CNN. Производится изменение архитектуры гибридной схемы для лучшей автоматизации процесса. Есть возможность отслеживания стабильности роста процента точности работы системы с помощью графиков точности и потерь.

Ключевые слова: CNN, YOLO, Keras, TensorFlow, агропромышленный комплекс, сорная растительность, заболевания растений, двухуровневая архитектура, ML-модель, графики точности и потерь.

Введение

Стихийное распространение сорных растений и заболеваний сельскохозяйственных культур продолжает оставаться одной из ключевых проблем для агропромышленного комплекса [1]. Настоящее исследование ставит своей целью разработку гибридного программного обеспечения, предназначенного для автоматического выявления и категоризации угроз. В основе подхода лежит комбинация двух моделей: свёрточной нейронной сети (CNN), отвечающей за высокоточную классификацию, и YOLO, обеспечивающей локализацию объектов в режиме реального времени [2, 3]. Такой синтез позволяет единовременно осуществлять как оперативное обнаружение нежелательной растительности, так и диагностику фитосанитарного статуса посевов.

1. Научная и практическая значимость

С научной точки зрения, значимость работы заключается в предложении инновационной двухуровневой архитектуры, которая совмещает функции детекции (YOLO) и классификации (CNN). Данная методология предлагает решение для задачи распознавания множества схожих классов — различных видов сорняков и болезней. Важнейшим достижением можно считать внедрённый механизм адаптивного обучения, дающий возможность дообучать модель на новых типах угроз путём расширения датасета; при этом применение современных методик обучения обеспечивает стабильную воспроизводимость итоговых результатов.

Практическая значимость исследования выражается в создании инструмента, готового к практическому применению для противодействия сорнякам и фотопатогенам. Разработанная система автоматизирует процесс мониторинга полей с привлечением дронов или специальной техники, гарантирует своевременное обнаружение угроз, а также создаёт детализированную базу данных с геопривязкой, что необходимо для анализа тенденций их распространения и последующего прогнозирования.

2. Техническая часть

2.1. Работа YOLO и CNN

Традиционные методы мониторинга полей, основанные на визуальном осмотре и ручном обходе, трудоёмки и не всегда объективны, а также не обеспечивают раннего выявления проблем, что может привести к быстрому распространению сорняков и инфекции и нанесению большого ущерба урожаю. Используются и спутниковые данные, к сожалению, обладающие недостаточной детализацией для уверенного утверждения фактов [4]. Для решения этой проблемы применяется модель You Only Look Once (YOLO), предназначенная для детекции объектов в реальном времени [3]. Её ключевое преимущество — анализ изображения в один шаг. В основе YOLO лежит CNN, последовательно выделяющая признаки изображений — от простых границ до сложных форм, что позволяет точно отличать сорные растения от культурных, а больные — от здоровых [5].

Процесс работы YOLO включает два основных этапа:

1. Детекция YOLO: изображение поля разбивается на сетку. Для каждой ячейки определяется вероятность наличия объекта и его предварительный класс.
2. Классификация CNN: обнаруженные объекты передаются на CNN для точной классификации типа (здоровое растение, конкретный сорняк, вид заболевания)

Модель встроена в камеру, установленную на сельскохозяйственную технику или дрон [1]. Камера передаёт изображения на вычислительное устройство, где YOLO проводит анализ данных и передаёт координаты.

2.2. Keras и TensorFlow

Для дальнейшей реализации используется связка Keras и TensorFlow. TensorFlow применяется как низкоуровневый фреймворк для эффективных вычислений, Keras — как высокоуровневый API для создания архитектур, абстракцию от сложностей TensorFlow и удобные средства для работы с данными и обучения моделей. Начиная с версии TensorFlow 2.x, Keras полностью интегрирован в его состав как основной API (tf.keras), что создает единую среду для разработки.

В рамках проекта реализовано гибридное решение, сочетающее потоковую обработку данных в реальном времени и пакетную обработку для решения комплексных задач. Данный подход позволяет:

- Обеспечивать быстрое реагирование на угрозы в реальном времени;
- Поддерживать высокую точность анализа через глубокую обработку данных;
- Объединять результаты для формирования полной картины состояния посевов;

Работа системы организована на двух уровнях, которые работают параллельно: потоковый слой для обработки данных в реальном времени и пакетный слой для углубленного анализа.

2.3. Обучение и работа ML-модели

Для создания датасета используется папка plant_dataset, содержащая классы: disease_type1, healthy, weed_type, в каждом из которых находится минимум 15 изображений. В идеале необходимо создать больше классов и добавить тысячи изображений в каждую, но для примера работы можно обойтись и этим. Для тестирования используется изображение поля или растений с названием test_field.

На первом этапе, при запуске кода происходит детекция YOLO: test_field разбивается на сетку, каждая клетка которой отвечает за свой участок картинки, а нейронная сеть предсказывает

вает, есть ли внутри клетки объект или его часть, прописывает его координаты и указывает, к какому классу он относится. На втором этапе каждый обнаруженный объект кадрируется и передаётся на вход CNN, реализованной на связке TensorFlow и Keras, для точной классификации его типа (здоровое растение, вид сорняка, тип заболевания). Такой подход сочетает высокую скорость детекции YOLO с точностью классификации CNN, обеспечивая надёжное решение для задач точного земледелия [1, 2].

Инициализация загружает предобученную модель YOLO из файла. Детекция использует метод predict для обработки изображения, возвращает объект sv.Detections из библиотеки supervision, содержащий координаты рамок, уверенность и другие метаданные.

Извлечение регионов — важная функция, которая итерирует по координатам рамок и с помощью операции среза NumPy (image [y1:y2, x1:x2]) вырезает все обнаруженные объекты в отдельное изображение для дальнейшей классификации.

Процесс решения поставленной цели работает по схеме «Обнаружить => Выделить => Классифицировать => Визуализировать»:

1. Обнаружение (Detection): Входное изображение обрабатывается моделью YOLO для нахождения ограничивающих рамок всех объектов.

2. Извлечение регионов (Region Extraction): на основе координат от YOLO исходное изображение разбивается на множество меньших изображений, каждое из которых содержит один обнаруженный объект.

3. Классификация (Classification): каждый вырезанный регион масштабируется и подается на вход обученной CNN, которая присваивает ему класс и оценивает уверенность предсказания.

4. Визуализация результатов (Visualization & Analysis): исходное изображение аннотируется цветными рамками и подписями, отражающими результат классификации. Строится отчёт с легендой и статистикой.

Этот подход эффективно решает проблему, когда один объект детекции (растение) может принадлежать к нескольким классам.

Для обучения используется функция потерь sparse_categorical_crossentropy, что позволяет использовать целочисленные метки (0, 1, 2). Применяются callback-функции: ModelCheckpoint, EarlyStopping, ReduceLROnPlateau для стабильного и эффективного обучения. Встроенный программный комплекс содержит вспомогательные функции: функция plot_training_history строит графики точности (accuracy) и потерь (loss) (рис. 1) на тренировочном и валидационном наборах данных для диагностики качества обучения.

График точности (а) показывает способность модели к обучению: синяя линия (training) обозначает точность (accuracy) на обучающем наборе данных, а оранжевая (validation) — точность на проверочном наборе данных. Обе линии растут на протяжении эпох, проверочная точность почти догоняет обучающую, которая достигает значения 0.85. Разрыв между обучающей и проверочной выборкой мал, что свидетельствует о приемлемом качестве обучения. График потерь (б) показывает величину ошибки модели (loss). Чем меньше значение, тем лучше модель минимизирует эту функцию. Здесь синяя линия (training) показывает потери на обучающих данных, а оранжевая (validation) — потери на проверочных данных. Обе линии идут близко друг к другу и стремятся к значению < 0.4, но с потерями на проверочных данных есть небольшие скачки.

Функция draw_detection_results рисует на исходном изображении цветные рамки и подписи с уверенностью предсказания. Функция show_detection_result создаёт информативную панель, объединяя изображение с результатами и текстовую легенду с детализацией (рис. 2). Легенда показывает типы распознаваемых классов (healthy, weed_type1, disease_type1) и статистику по обнаруженным объектам. Низкая уверенность предсказаний (37–43 %) указывает на необходимость проверки экспертом и возможность улучшения модели путём расширения датасета.

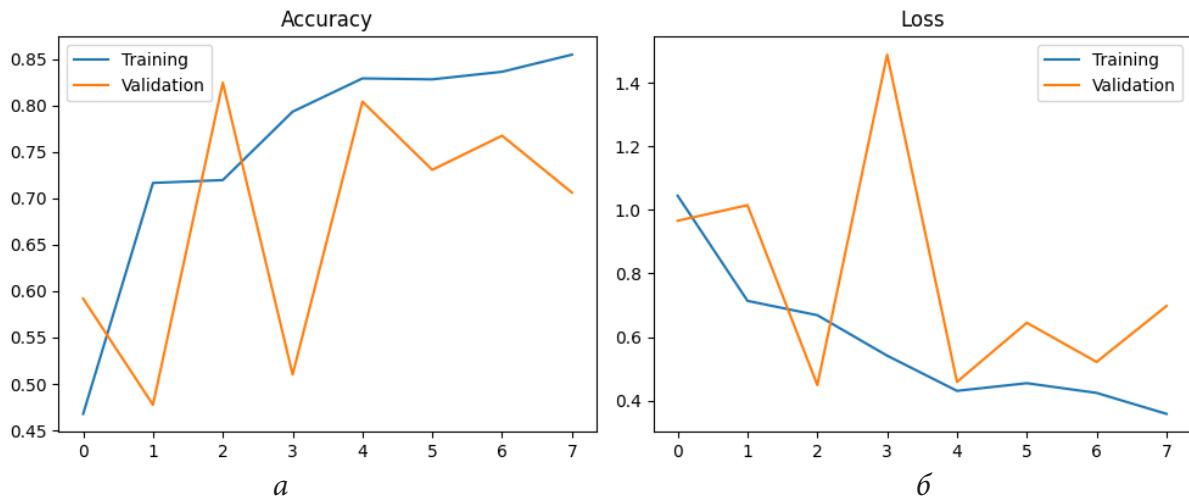


Рис. 1. Графики точности (а) и потерь (б) при обучении нейронной сети



Рис. 2. Результаты анализа изображения

В решении поставленной задачи используется скорость и универсальность YOLO для локализации объектов и точность кастомной CNN для их классификации, что превосходит монолитные архитектуры по точности на специфичных задачах. В отличие от «чёрного ящика», система предоставляет визуально аннотированные результаты и оценку уверенности, что крайне важно для агрономов при принятии решений.

Классификатор можно переобучить на новые виды сорняков или болезней, просто добавив новые данные в соответствующие папки без необходимости переобучать или изменять модель детекции [5].

Также код включает проверенные методы, позволяющие модели учиться качественно и избегать ошибок, что свидетельствует о надёжном подходе к разработке ML-моделей. Разработанное гибридное решение на основе нейронных сетей обладает научной новизной и практической ценностью для агропромышленного комплекса.

Исследование выполнено за счет гранта Российского научного фонда № 22-71-10102-П, <https://rscf.ru/project/22-71-10102-П/>

Заключение

В результате исследования была достигнута цель по созданию информационной системы с использованием машинного обучения для обнаружения сорной растительности и диагностики заболеваний сельскохозяйственных культур. Гибридная схема, сочетающая YOLO и свёрточную нейронную сеть на TensorFlow/Keras, показала высокую эффективность в распознавании множества биологических угроз. Её двухэтапная архитектура, реализующая принцип разделения детекции и классификации, представляет собой научно обоснованный и практический подход для решения сложных задач компьютерного зрения в сельском хозяйстве.

Литература

1. Партель В., Какарла С., Ампатзидис Й. (2019). Разработка и оценка недорогой и интеллектуальной технологии точной борьбы с сорняками с использованием искусственного интеллекта. Comput. Electron. Agric., 157, 339–350. // – URL: <https://doi.org/10.1016/j.compag.2018.12.048> (дата обращения: 08.11.2025)
2. Данг Ф., Чен Д., Лу И., Ли З. (2023). YOLOWeeds: новый эталон детекторов объектов YOLO для многоклассового обнаружения сорняков в системах производства хлопка. Comput. Electron. Agric., 205, 107655. // – URL: <https://doi.org/10.1016/j.compag.2023.107655> (дата обращения: 09.11.2025)
3. Панков Р. YOLO object detection: как нейросеть распознает объекты в реальном времени // – URL: <https://timeweb.cloud/blog/yolo-neyroset-obnaruzhenie-obektov> (дата обращения: 10.11.2025)
4. Мониторинг полей в сельском хозяйстве // – URL: <https://agrosignal.com/articles/monitoring-poley-v-selskom-khozyaystve/> (дата обращения: 12.11.2025)
5. Адхината Ф. В., Сумихарто Р. (2024). Комплексное исследование классификации сорняков и сельскохозяйственных культур с использованием машинного и глубокого обучения. Искусственный интеллект в сельском хозяйстве. // – URL: <https://doi.org/10.1016/j.aiia.2024.06.005> (дата обращения: 17.11.2025)
6. Наглядно о том, как работает свёрточная нейронная сеть // – URL: <https://habr.com/ru/companies/skillfactory/articles/565232/> (дата обращения: 19.11.2025)

АЛГОРИТМ ПОСТРОЕНИЯ БЕРЕГОВОЙ ЛИНИИ ВОДОЁМА НА ОСНОВЕ ОБЛАКА ТОЧЕК

Л. В. Горбенко, А. Ю. Бондаренко, Д. Д. Савлучинская, В. Н. Литвинов

Донской государственный технический университет

Аннотация. Мониторинг береговой линии водоёма представляет собой важную научно-практическую задачу в контексте изменения климата и антропогенного воздействия на прибрежные экосистемы. Существующие детерминированные методы интерполяции не позволяют в достаточной мере учесть пространственную структуру данных и являются чувствительными к шуму. В данной работе рассмотрен алгоритм построения береговой линии водоёма на основе облака точек и представлен подход, который позволяет автоматизировать процесс восстановления этой линии и может быть использован для мониторинга динамики береговой зоны.

Ключевые слова: геостатистическая интерполяция, ML-модель, Random Forest, Kriging, мониторинг, береговая линия, облако точек.

Введение

Береговая линия как динамичная граница раздела сред представляет собой сложный природный объект, требующий регулярного мониторинга [1]. Современные технологии дистанционного зондирования, например, воздушное лазерное сканирование, позволяют получать высокоточные трёхмерные модели рельефа. Однако преобразование нерегулярного облака лидарных точек в векторное представление береговой линии остаётся сложной научно-технической задачей. В статье предложен комплексный подход, сочетающий в себе геостатистическую интерполяцию для построения адекватной цифровой модели поверхности и машинное обучение для семантической сегментации территории на основе комплекса геоморфометрических показателей. Такой подход существенно повышает точность и позволяет автоматизировать процесс выделения береговой линии.

1. Алгоритм для ML

Вычислительное устройство получает пакет изображений, сделанные спутником. Данные этих изображений фильтруются на подходящие для конкретной задачи снимки. Далее происходит подготовка отфильтрованных изображений – изображения вырезаются по областям водоема, что снижает вычислительную нагрузку GPU.

Для распознавания береговой линии водоема нужно выбрать алгоритм машинного обучения. Был рассмотрен ряд подходящих для нашей задачи: случайные леса (Random Forest), градиентный бустинг (Gradient Boosting Machines), Стекинг (Stacking), сравнительная характеристика приведена в табл. 1.

На основе табл. 1 алгоритм Random Forest лучше подходит для нашей задачи из-за скорости обучения. Алгоритм случайного леса представляет собой метод машинного обучения, который строит множество различных деревьев и объединяет их ответы, чтобы получить более точный ответ. Данный алгоритм хорошо подходит для задач классификации. Будем фокусироваться на участки пикселей, которые проходят на границы «Вода-суша», а внутренние части водоема будем пропускать. Это позволит повысить точность решения и ускорить обработку данных.

Таблица 1

Сравнительная характеристика алгоритмов

	Случайный лес	Градиентный бустинг	Стэкинг
Основная идея	Создание множества независимых деревьев на основе случайных подбора данных и признаков	Создание последовательных деревьев, где каждое новое дерево предсказывает ошибки предыдущих	Комбинирование нескольких различных алгоритмов
Процесс обучения	Параллельный	Последовательный	Параллельный-последовательный
Скорость обучения	Высокая	Средняя	Низкая
Склонность к переобучению	Низкая	Высокая	Очень высокая
Предсказание	Высокое	Очень высокое	Очень высокое

Для решения задачи используем машинное обучение с учителем — классификацию (предсказываем, суша или вода). Алгоритм разбивает изображение на пиксели и каждому присваивает метку «суша» или «вода», в итоге мы получаем неидеальную пиксельную маску водоема.

Метки переводятся в векторный контур, который выглядит как набор отрезков. Подается набор массивов связанных отрезков, описывающих геометрию береговой линии. В задачах математического моделирования геометрия должна быть описана непрерывной дважды дифференцируемой функцией. Для сглаживания и получения плавной границы водоема будем использовать кригинг интерполяцию. Для улучшения точности результатов бинарную маску водоема можно разбить на несколько частей и провести кригинг интерполяцию по частям, после чего объединить в одно готовое изображение береговой линии.

2. Методы интерполяции для построения ЦМП из облака точек

Kriging (кригинг), в отличие от других методов интерполяции, представляет собой не просто математическую формулу, а целостный геостатистический подход, который позволяет не только предсказывать значения, но и оценивать точность этих предсказаний. [2]. Именно это делает его предпочтительным для восстановления береговой линии.

Если рассматривать другие методы, например, метод обратных взвешенных расстояний, сплайны или триангуляцию, то получим наличие в них недостатков, которые будут мешать в решении нашей задачи (табл. 2) [3]. В кригинге таких недостатков нет, что делает его оптимальным выбором для точного определения положения береговой линии.

Заключение

На основе проведенного исследования был разработан и представлен комплексный алгоритм построения береговой линии водоёма, сочетающий методы машинного обучения и геостатистической интерполяции для обработки данных дистанционного зондирования.

Предложенный подход позволяет автоматизировать процесс выделения береговой линии, минимизировать влияние шума в исходных данных и получить результат, пригодный для мониторинга динамики береговой зоны. Перспективы дальнейших исследований видятся в

Таблица 2

Сравнительная характеристика методов интерполяции

Метод	Принцип работы	Ключевые недостатки для задачи
Метод обратных взвешенных расстояний	Значение в ячейке вычисляется как средневзвешенное значений ближайших точек, где вес обратно пропорционален расстоянию.	Не учитывает пространственную структуру и тренды данных. Склонен создавать артефакты «бычьего глаза» вокруг точек данных, что может исказить плавный переход береговой линии.
Сплайны	Использует математические функции, минимизирующие общую кривизну поверхности. Результат — гладкая поверхность, точно проходящая через исходные точки.	Чрезмерное сглаживание может маскировать естественные, но резкие формы рельефа.
Триангуляция	Строит поверхность из множества смежных треугольников по исходным точкам (триангуляция Делоне).	Поверхность получается негладкой, состоит из граней. Модель может не учитывать общих пространственных тенденций.

адаптации алгоритма для работы с различными типами водоёмов, оптимизации вычислительных процессов и интеграции с системами непрерывного экологического мониторинга.

Исследование выполнено за счет гранта Российского научного фонда № 25-71-20001, <https://rscf.ru/project/25-71-20001/>

Литература

1. Фам Ч. Разработка автоматического метода определения береговой линии реки / Ч. Фам, Х. Чан // Cifra. Науки о Земле и окружающей среде. — 2024. — №1 (1). — URL: <https://geosciences.cifra.science/archive/1-1-2024-september/10.60797/GEO.2024.1.6> (дата обращения: 10.11.2025)
2. Как работает инструмент Кригинг // – URL: <https://desktop.arcgis.com/ru/arcmap/latest/tools/3d-analyst-toolbox/how-kriging-works.htm> (дата обращения: 14.11.2025)
3. Сравнение методов интерполяции // – URL: <https://desktop.arcgis.com/ru/arcmap/latest/tools/3d-analyst-toolbox/comparing-interpolation-methods.htm> (дата обращения: 21.11.2025)

СРАВНЕНИЕ ЭФФЕКТИВНОСТИ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ УРОЖАЙНОСТИ СЕЛЬСКОХОЗЯЙСТВЕННЫХ КУЛЬТУР

С. А. Гудилова, Е. М. Аристова

Воронежский государственный университет

Аннотация. В работе исследуется эффективность методов машинного обучения для прогнозирования урожайности сельскохозяйственных культур Центрального Черноземья. Модели регрессии, древовидных алгоритмов и градиентного бустинга были обучены на исторических метеорологических показателях и данных об урожайности. Сравнение эффективности проведено с помощью различных метрик качества. Найдены ключевые аспекты влияния погодных условий. Результаты исследования показали, что лучшими методами оказались бустинговые методы.

Ключевые слова: машинное обучение, прогнозирование урожайности, сравнение алгоритмов, метрики качества, агропромышленный комплекс, сельское хозяйство, множественная регрессия, градиентный бустинг, деревья решений, случайный лес, метод ближайших соседей, метод опорных векторов, нейронные сети.

Введение

Агропромышленный комплекс России — ключевой сектор экономики, обеспечивающий продовольственную безопасность и сырьё для смежных отраслей. В условиях глобализации и изменения климата особую роль играют зерновые культуры, такие как пшеница, ячмень и кукуруза, которые служат основой питания, кормовой базой животноводства и важным экспортным ресурсом.

Урожайность зерновых зависит от множества факторов, среди которых погодные условия — температура, осадки, влажность — оказывают непосредственное влияние на рост, развитие и устойчивость растений к болезням.

Центральное Черноземье, один из самых плодородных регионов России, обладает оптимальными условиями для выращивания зерновых. Тем не менее, даже здесь климатические аномалии, такие как засухи или избыточные осадки, могут существенно снижать урожай. Поэтому анализ взаимосвязи погоды и урожайности в этом регионе имеет важное научное и практическое значение для агробизнеса.

Целью данной работы является исследование эффективности методов машинного обучения для прогнозирования урожайности сельскохозяйственных культур. Для достижения цели в ходе выполнения этой работы будут поставлены и решены следующие задачи: сформировать набор данных на основе метеорологических показателей и исторических данных об урожайности; провести предобработку данных (очистка, обработка пропусков, кодирование категориальных признаков); обучить несколько регрессионных моделей машинного обучения; провести сравнительный анализ моделей на основе выбранных метрик ($RMSE$, MAE , R^2); выявить наиболее значимые метеорологические признаки, влияющие на урожайность.

1. Описание данных

Для анализа использовались метеорологические данные, агрегированные за вегетационный период (май–август), который является ключевым для формирования урожая подсолнечника, кукурузы и яровой пшеницы в Центральном Черноземье. Для каждого года (2020–2024) был рассчитан следующий набор данных в период вегетации растений: медианная темпера-

тура воздуха; медианная относительная влажность воздуха; сумма осадков; количество солнечных дней (лето); сумма активных температур GDD; гидротермический коэффициент ГТК. Медиана использована для минимизации влияния выбросов (кратковременных скачков температуры и влажности). Сумма осадков отражает общее количество влаги, полученное сельскохозяйственными культурами естественным путем. Количество солнечных лучей влияет на протекание фотосинтеза, укорачивает межфазовые промежутки, особенно этап цветения и созревания. GDD (сумма активных температур) ищется умножением базовой температуры (подсолнечник — 7 °C, яровая пшеница — 5 °C, кукуруза — 10 °C) на среднее арифметическое максимальной и минимальной температуры. ГТК (гидротермический коэффициент) отображает отношение суммы осадков и 10 % сумм среднесуточных температур, превышающих 10 °C [1,3].

Данный набор признаков позволяет комплексно охарактеризовать тепловой, водный и световой режимы вегетационного периода, что критически важно для построения точных моделей прогнозирования урожайности.

2. Методы машинного обучения

Для анализа были выбраны следующие методы:

1. *Регрессионный анализ* направлен на представление изучаемой зависимости в виде аналитической формулы с предварительным выделением зависимых и объясняющих переменных. *Множественной регрессией* называется уравнение связи переменной u и нескольких независимых переменных. Она остается фундаментальным инструментом для установления количественных зависимостей между метеорологическими параметрами и урожайностью [2, 4].

2. *Деревья решений* представляют собой непараметрический метод, который строит иерархическую структуру правил для прогнозирования урожайности. Алгоритм рекурсивно разбивает пространство признаков на области, минимизируя при этом критерий неоднородности. Критерий Джини помогает оценить чистоту получившегося узла, а минимизация дисперсии позволяет находить оптимальные границы разбиения. Среднее значение целевой переменной по всем наблюдениям формирует в конечном узле прогноз [4, 5].

3. *Случайный лес* (Random Forest) представляет собой ансамблевый метод, сочетающий множество деревьев решений. Каждое дерево строится на случайной подвыборке данных (бутстрэп-агрегирование) с использованием случайного подмножества признаков. Прогноз получается усреднением всех деревьев. За счет этого повышается стабильность и точность модели [5, 6].

4. *Градиентный бустинг* — метод машинного обучения, где деревья строятся на последовательном построении ансамбля решающих деревьев таким образом, что каждое последующее дерево целенаправленно корректирует ошибки предыдущих. Метод особенно эффективен для прогнозирования урожайности по метеорологическим данным. *XGBoost* предотвращает переобучение через регуляризацию и эффективно работает с пропущенными значениями. *LightGBM* использует выборку по градиенту для ускоренного обучения на больших данных [2, 8].

5. Метод k -ближайших соседей — непараметрический метод, основанный на поиске похожих наблюдений. Прогноз вычисляется как среднее значение k ближайших соседей. Это позволяет вести учет локальных закономерностей данных [2].

6. *Метод опорных векторов* (SVM) строит оптимальную разделяющую гиперплоскость в пространстве признаков, используя для этого радиальное базисное ядро. Алгоритм максимизирует отклонения между объектами разных классов. Для задач регрессии минимизируется разница между фактическими и прогнозируемыми данными [9].

7. *Нейронные сети* — многослойные модели, которые способны выявлять сложные нелинейные зависимости в агрометеоданных. Прогноз формируется последовательным преобра-

зованием данных через слои нейронов, которые тесно связаны между собой. Это позволяет адаптироваться к различным сценариям [7, 10].

Для оценки качества моделей были использованы следующие метрики [2]:

1. *MSE* — среднеквадратичная ошибка, оценивающая среднюю величину квадрата отклонения прогноза от фактических значений;

2. *MAE* — средняя абсолютная ошибка, отображающая среднюю абсолютную ошибку отклонения;

3. *R²* — коэффициент детерминации, показывающий, какая часть дисперсии целевой переменной объясняется моделью.

Результаты сравнения эффективности алгоритмов представлены в следующем разделе в табл. 1.

3. Сравнение эффективности алгоритмов

Таблица 1

Сравнение метрик качества моделей прогнозирования урожайности

Метод	<i>R²</i>	MSE	MAE
Линейная регрессия	0.74	75.46	6.62
Случайный лес	0.84	47.39	5.53
Деревья решений	0.83	49.09	5.4
Градиентный бустинг	0.84	45.52	5.36
XGBoost	0.84	46.12	5.37
LightGBM	0.84	46.2	5.46
Метод k-ближайших соседей	0.77	66.38	6.38
Метод опорных векторов	0.77	65.97	6.25
Нейронные сети	0.76	68.72	6.48

На основе табл. 1 можно выделить, что методы градиентного бустинга показали наилучшие результаты. Они хорошо обобщают данные и обладают неплохой точностью. Случайный лес также показал хорошие результаты, но немного уступил бустинговым методам. Можно сделать вывод, что дерево решений находится в середине рейтинга, это обусловлено склонностью к переобучению и чувствительностью к изменениям. Линейная регрессия, метод опорных векторов, нейронные сети и метод ближайших соседей показали относительно высокие ошибки (*MSE*, *MAE*) и низкий *R²*. Это может быть связано с тем, что данные имеют нелинейные зависимости, которые плохо описываются линейными алгоритмами.

Заключение

В ходе исследования эффективности методов машинного обучения для прогнозирования урожайности наилучшие результаты показали ансамблевые алгоритмы. Алгоритмы градиентного бустинга и случайный лес превзошли остальные методы.

Разработанные модели могут быть интегрированы в системы поддержки принятия решений для агропромышленных предприятий. Полученные результаты подтверждают перспективность применения машинного обучения для сельского хозяйства и повышения точности прогнозов урожайности.

Перспективными направлениями дальнейших исследований являются: учет пространственно-временных особенностей данных, разработка интерпретируемых моделей и создание систем мониторинга в реальном времени.

Литература

1. *Арютов Б. А. Влияние погодных факторов на эффективность удобрений в почвозащитном севообороте / Б. А. Арютов, А. Н. Важенин, А. В. Пасин. – Пенза : Академия Естествознания, 2010. – 210 с.*
2. *Макеев К. А. Сравнение эффективности алгоритмов машинного обучения в задачах прогнозирования урожайности сельскохозяйственных культур / К. А. Макеев, А. В. Греченева, Я. С. Котов, А. Н. Голбан, Д. М. Смыслов // КиберЛенинка. – URL: <https://cyberleninka.ru/article/n/sravnenie-effektivnosti-algoritmov-mashinnogo-obucheniya-v-zadachah-prognozirovaniya-urozhaynosti-selskohozyaystvennyh-kultur>.*
3. *Проберж Э. С. Влияние погодных факторов на эффективность удобрений в почвозащитном севообороте // КиберЛенинка. – URL: <https://cyberleninka.ru/article/n/vliyanie-pogodnyh-faktorov-na-effektivnost-udobreniy-v-pochvozaschitnom-sevooborote/> viewer.*
4. *Awasthi P. Performance Assessment of Machine Learning Techniques for Corn Yield Prediction / P. Awasthi, S. Mishra, N. Gupta // International Conference on Advanced Network Technologies and Intelligent Computing. – Cham : Springer Nature Switzerland, 2022. – P. 320–335.*
5. *Breiman L. Classification and Regression Trees / L. Breiman, J. Friedman, R. Olshen, C. Stone. – 1st ed. – Boca Raton : Chapman and Hall/CRC, 1984. – 368 p.*
6. *Breiman L. Random Forests / L. Breiman // Machine Learning. – 2001. – Vol. 45, No. 1. – P. 5–32.*
7. *Cherkassky V. Another Look at Statistical Learning Theory and Regularization / V. Cherkassky, Y. Ma // Neural Networks. – 2009. – Vol. 22. – P. 958–969.*
8. *Friedman J. H. Greedy Function Approximation: A Gradient Boosting Machine / J. H. Friedman // The Annals of Statistics. – 2001. – Vol. 29, No. 5. – P. 1189–1232.*
9. *Liu X. A Comparative Analysis of Support Vector Machines and Extreme Learning Machines / X. Liu, C. Gao, P. Li // Neural Networks. – 2012. – Vol. 33. – P. 58–66.*
10. *Vapnik V. N. An Overview of Statistical Learning Theory / V. N. Vapnik // IEEE Transactions on Neural Networks. – 1999. – Vol. 10, No. 5. – P. 988–999.*

ПОСТРОЕНИЕ ГРАФА ВЗАИМОДЕЙСТВИЙ МИКРОСЕРВИСОВ НА ОСНОВЕ АНАЛИЗА ИСХОДНОГО КОДА С ПРИМЕНЕНИЕМ БОЛЬШОЙ ЯЗЫКОВОЙ МОДЕЛИ

Д. С. Дроздов, Ю. Е. Гагарин

Калужский филиал Московского государственного технического университета имени Н. Э. Баумана

Аннотация. В статье рассматривается подход к автоматическому извлечению архитектурных связей между микросервисами на основе анализа исходного кода с применением большой языковой модели (LLM), запущенной локально. Цель исследования — разработка методики, позволяющей выявлять взаимодействия между сервисами без использования трассировок и инструментов мониторинга. В работе описана структура данных, алгоритм взаимодействия с моделью и проведён эксперимент на репозитории Google Cloud Platform Microservices Demo. Полученные результаты показывают, что локальные LLM способны достоверно восстанавливать граф взаимодействий и могут быть интегрированы в системы статического анализа архитектуры.

Ключевые слова: большие языковые модели, микросервисная архитектура, статический анализ, граф зависимостей, искусственный интеллект, LLM, программный код.

Введение

Микросервисная архитектура является современным стандартом построения корпоративных и облачных систем. Её преимущества — модульность, масштабируемость и независимость развертывания отдельных компонентов — одновременно создают проблемы анализа и сопровождения. При увеличении числа сервисов и сложных зависимостей между ними становится трудно отслеживать связи между компонентами, особенно при активной разработке и обновлениях [1].

Традиционные инструменты анализа, такие как Jaeger, Zipkin, Kiali, выполняют визуализацию взаимодействий на основе данных трассировки (runtime tracing). Однако подобные решения не подходят для статического анализа, когда система не запущена или данные мониторинга недоступны.

Одним из направлений решения данной задачи является использование больших языковых моделей (LLM), которые способны интерпретировать исходный код, комментарии и документацию, формируя семантическое понимание структуры программных систем. Современные открытые модели (Mistral, Qwen, LLaMA 3) могут быть запущены локально и применены для анализа даже без доступа к облачным сервисам.

Таким образом, целью настоящей работы является разработка методики применения локальной LLM для автоматического извлечения архитектурных связей и построения графа взаимодействий микросервисов по исходному коду.

1. Обзор существующих подходов

Современные методы анализа архитектуры программных систем опираются на два ключевых направления: статическое исследование структуры исходного кода и динамическое отслеживание поведения системы во время выполнения. Каждое из этих направлений имеет свои преимущества и ограничения, которые во многом определяют их роль при анализе микросервисных архитектур. В условиях, когда архитектура описывается большим количеством независимых сервисов, взаимодействующих через сетевые API, необходимость в действенных методах анализа становится особенно заметной.

Статический анализ стремится извлечь информацию, опираясь исключительно на исходный код, конфигурации и определённые договорённости в структуре проекта. Он включает изучение импортов, вызовов функций, сетевых обращений, конфигурационных файлов и интерфейсов gRPC/HTTP [2]. Наиболее распространённые инструменты — SourceTrail, Understand, SonarQube — предоставляют возможность построения графов зависимостей, в том числе в виде визуальных диаграмм. Однако возможности таких инструментов ограничены тем, что они анализируют код буквально, без глубокого учёта архитектурного контекста.

Например, если сервис динамически формирует URL-адрес или использует зависимость, определённую во внешнем конфигурационном файле, статический анализатор не всегда способен корректно выявить это взаимодействие. Ещё одним недостатком является чувствительность к стилю оформлению проекта: если общие соглашения об именовании сервисов или структурировании каталогов не соблюдаются, статические методы теряют точность.

Динамический анализ основывается на сборе данных непосредственно во время работы системы. Применение распределённого трассинга, логирования запросов и мониторинга позволяет точно фиксировать реальные взаимодействия сервисов. Инструменты вроде Jaeger, Zipkin и Prometheus способны строить графы вызовов, отображать задержки, выявлять узкие места, а также определять интенсивность обмена данными между сервисами. Динамический анализ выгодно отличается достоверностью данных: он фиксирует не потенциальные, а реальные вызовы. Однако он требует полноценного развёртывания системы, настройки трассировки, а также генерации нагрузки, чтобы получить репрезентативные данные. В ситуациях, когда изучаемый проект не запускается или когда анализ проводится на ранних этапах разработки, применение динамических методов оказывается невозможным.

Развитие больших языковых моделей открыло необычные возможности для интерпретации программного кода. В отличие от статических инструментов, которые опираются на формальные правила анализа, LLM обладают нейросетевой способностью обобщать, интерпретировать и реконструировать архитектурные связи на основе контекста, примеров и скрытых закономерностей. С 2021 года появились специализированные модели, ориентированные на работу с исходным кодом: CodeBERT, CodeT5, StarCoder, PolyCoder, CodeLLaMA. Эти модели прошли обучение на миллиардах строк кода, документации и технических материалов, что сформировало у них способность понимать типовые структуры проектов, характерные паттерны микросервисов, способы организации сетевых взаимодействий и принципы сервисных API [3].

Особенностью таких моделей является то, что они способны работать не только с синтаксисом, но и с семантикой. Например, модель может обнаружить связь между сервисами на основе вызова функции клиента gRPC, даже если импорт или явная зависимость не отражены в явном виде. Она может определить, что определённое имя переменной, конфигурационный параметр или шаблон URL-адреса относится к другому сервису, и сделать вывод о наличии взаимодействия. Благодаря способности к обобщению модели иногда выявляют зависимости, которые не фиксируются статическим анализом, поскольку они выведены из контекста, а не напрямую указаны в коде.

2. Исходные данные и постановка задачи

Для проверки работоспособности предложенного подхода был выбран открытый демонстрационный проект **microservices-demo**, опубликованный на GitHub компанией Google. Он представляет собой учебный пример микросервисной архитектуры, который широко используется в исследованиях и учебных материалах. Достоинством проекта является его умеренная сложность: он содержит достаточно много компонентов и связей, чтобы продемонстрировать разнообразие взаимодействий, но при этом остаётся достаточно компактным для полноцен-

ного анализа. Кроме того, структура кода аккуратно оформлена, что снижает влияние побочных факторов на точность эксперимента [4].

Архитектура *microservices-demo* включает одиннадцать микросервисов, каждый из которых выполняет отдельную роль в работе условного интернет-магазина. Среди них есть сервис отображения пользовательского интерфейса, сервис каталога товаров, сервис корзины, сервис оплаты, сервис рекомендаций и другие компоненты, включая хранилище данных. Взаимодействие между сервисами организовано преимущественно через HTTP и gRPC, что делает проект показательной моделью для анализа типичных паттернов микросервисных связей. Каждый сервис реализован на языке Go, что упрощает автоматический анализ, поскольку код структурирован предсказуемым образом, а клиентские библиотеки gRPC и HTTP имеют узнаваемые сигнатуры.

Для проведения исследования необходимо было сформулировать цель и конкретные задачи. Основная цель заключалась в том, чтобы разработать и проверить методику извлечения архитектурных зависимостей с помощью локально работающей большой языковой модели. Такая модель должна работать без подключения к облачным сервисам, что делает её применение более доступным и безопасным в реальных условиях. Задача исследования заключалась не только в технической проверке способности модели анализировать код, но и в оценке точности, которой можно достичь при использовании подобного подхода.

Были определены три ключевых задачи. Первая состояла в разработке процедуры анализа исходного кода, включающей разбиение файлов на удобные для обработки моделью фрагменты, подготовки соответствующих запросов и получения структурированных JSON-ответов. Вторая задача включала построение графа взаимодействий сервисов. Требовалось не только собрать все выявленные моделью связи, но и корректно сгруппировать, нормализовать и визуализировать их, чтобы получить читабельное представление архитектуры. Третья задача заключалась в оценке точности полученного графа путём сравнения с эталонной архитектурой, которую можно восстановить из официальной документации проекта.

Таким образом, эксперимент на *microservices-demo* позволял не только проверить работу самого алгоритма, но и сравнить потенциал LLM-подхода с традиционными методами. Проект был выбран как подходящая тестовая площадка, позволяющая исследовать преимущества и ограничения семантического анализа кода с применением современных языковых моделей.

3. Методика исследования

Методика исследования была построена так, чтобы обеспечить последовательную обработку исходного кода и получение корректного графа взаимодействий между микросервисами. Основная идея заключалась в том, чтобы разбить проект на фрагменты, передать их модели и затем объединить результаты.

Система анализа включала четыре основных компонента (рис. 1). Первый компонент — модуль парсинга кода. Он проходил по всем файлам проекта и формировал список источников, которые затем разбивались на небольшие фрагменты. Такое разбиение необходимо, потому что языковые модели имеют ограничение на длину входного контекста. Использование небольших блоков повышало качество анализа и позволило избежать пропуска важных деталей.

Второй компонент — интерфейс взаимодействия с LLM. В эксперименте использовалась локальная модель **Mistral 7B Instruct** [5], запущенная через **Ollama**. Модель была загружена на компьютер с **GPU NVIDIA RTX 4060 Ti**. Поддержка 8192 токенов контекста позволила анализировать достаточно большие части файлов. Локальный способ работы обеспечил стабильное время ответа и отсутствие зависимости от облачных сервисов.

Третий компонент — интерпретатор ответов модели. Для каждого фрагмента формировался специальный промпт, в котором указывалось, что необходимо определить взаимодействия

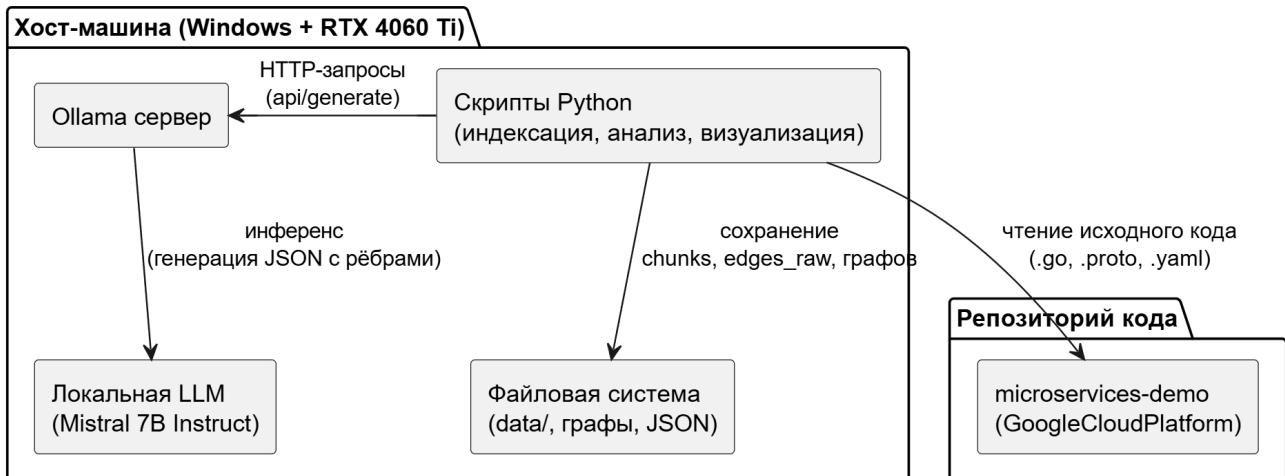


Рис. 1. Архитектура системы анализа репозитория

между сервисами. Модель должна была вернуть результат строго в формате JSON, где описывались источник, цель, тип связи и короткий фрагмент кода, подтверждающий её вывод. Если модель не находила зависимостей, она должна была вернуть пустой список. Это требование позволило упростить автоматическую обработку данных.

Ниже приведён пример промпта:

«*You are an expert in microservice architecture.*

Analyze the provided code fragment and extract INTER-SERVICE interactions that belong to the Google microservices-demo.

Return ONLY JSON (no text), as a list of edges. Each edge:

- source: service name (one of: {services})*
- target: service name (one of: {services})*
- type: one of [«http», «grpc», «pubsub», «tcp», «other»]*
- evidence: list of short code snippets or line hints from the fragment*

If you are not sure or the fragment has no interactions, return [].

Code fragment (UTF-8): {code}»

Четвёртый компонент отвечал за объединение результатов и построение графа. Все JSON-ответы сохранялись в файл, после чего проходили этап сортировки и фильтрации. Дубликаты удалялись, ошибки исправлялись, а связи приводились к единому формату. Затем данные передавались в модуль визуализации, где строился ориентированный граф. Узлы графа соответствовали микросервисам, а рёбра отражали типы и направления взаимодействий. Для визуализации использовалась библиотека NetworkX.

4. Реализация эксперимента

Эксперимент проводился на проекте **microservices-demo**, который содержит несколько микросервисов и достаточное количество исходного кода для анализа. В процессе работы было обработано девять сервисов. Они включали **162 файла** общим объёмом более **1,1 миллиона** строк кода. Такой объём данных позволил проверить, насколько хорошо модель справляется с анализом реального проекта.

Файлы разбивались на фрагменты размером около **1500 токенов** с перекрытием 100 токенов. Перекрытие помогало избежать ситуаций, когда важная часть кода попадала на границу между двумя блоками и могла быть неправильно проанализирована. После разбиения был получен 841 фрагмент. Каждый из них подавался в модель **Mistral 7B**. Модель возвращала JSON-ответ, который сохранялся в файл без дополнительной обработки на этом этапе (рис. 2).

Среднее время обработки одного блока составляло около **4,2 секунды**. Полная обработка всех фрагментов заняла чуть меньше часа. По завершении анализа все JSON-ответы были объединены, и дублирующиеся связи удалены. В результате был сформирован итоговый граф, отражающий взаимодействия между сервисами.

Полученный граф показал несколько ожидаемых связей, характерных для проекта. Например, модель определила обращение сервиса frontend к productcatalogservice и cartservice с использованием HTTP. Также были восстановлены связи checkoutservice → paymentservice по gRPC, recommendationservice → productcatalogservice по HTTP и cartservice → redis по TCP. Эти результаты хорошо совпали с теми, что описаны в документации проекта.

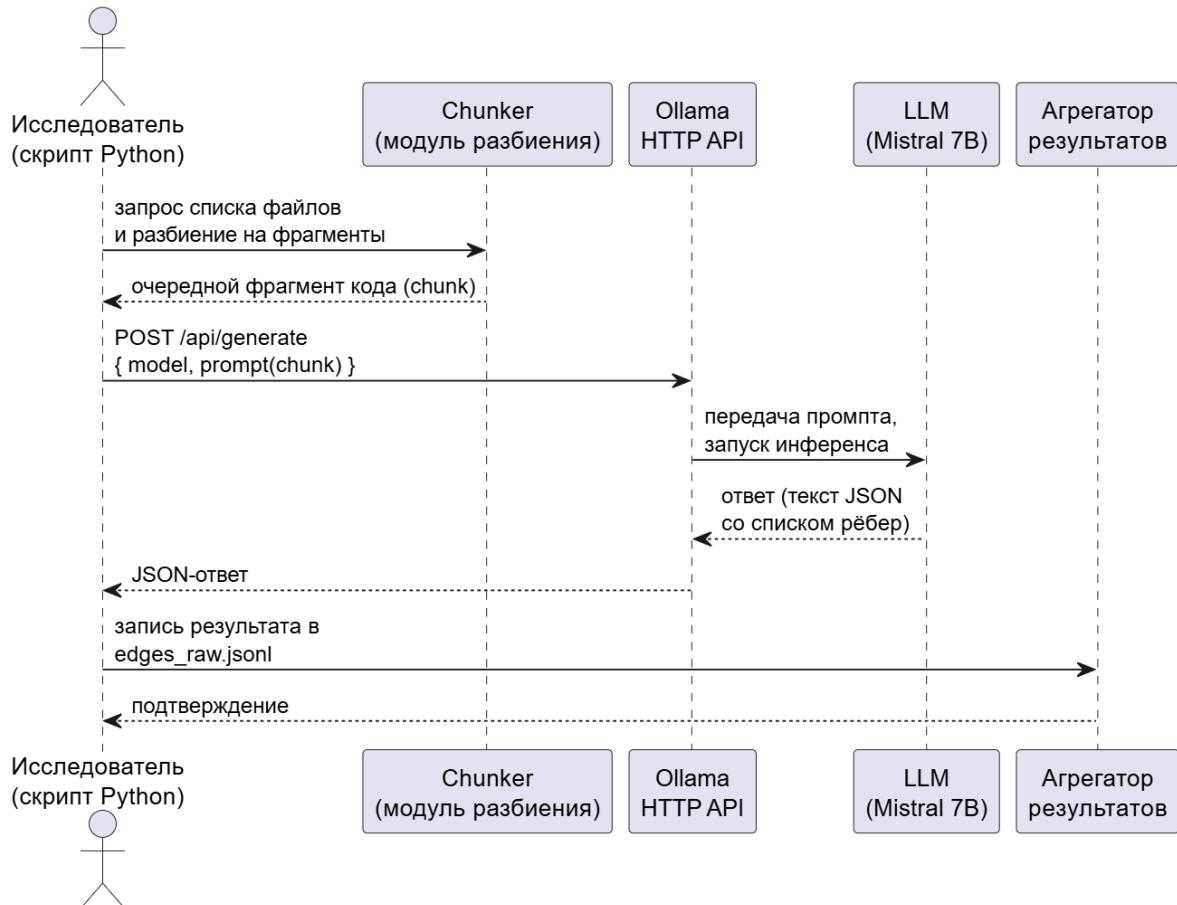


Рис. 2. Диаграмма процесса анализа репозитория

5. Оценка результатов эксперимента

Качество извлечённых связей оценивалось с помощью сравнения результатов анализа с официальной документацией проекта. Для этого был создан список ожидаемых взаимодействий между сервисами, после чего сравнивались связи, выявленные моделью.

Всего было обнаружено **122** уникальных связи (рис. 3). Из них **103** полностью соответствовали реальным. **10** связей были ложноположительными, то есть модель ошибочно определила взаимодействия, которых в проекте нет. Ещё **9** связей оказались частично корректными: модель определила направление зависимости, но ошиблась в типе взаимодействия либо неверно указала подтверждающий код.

Итоговая точность составила **84,6 %**. Это достаточно высокий результат для анализа, выполняемого на основе исходного кода без использования данных трассировки. Модель смогла восстановить большую часть архитектуры, включая некоторые косвенные связи, которые не

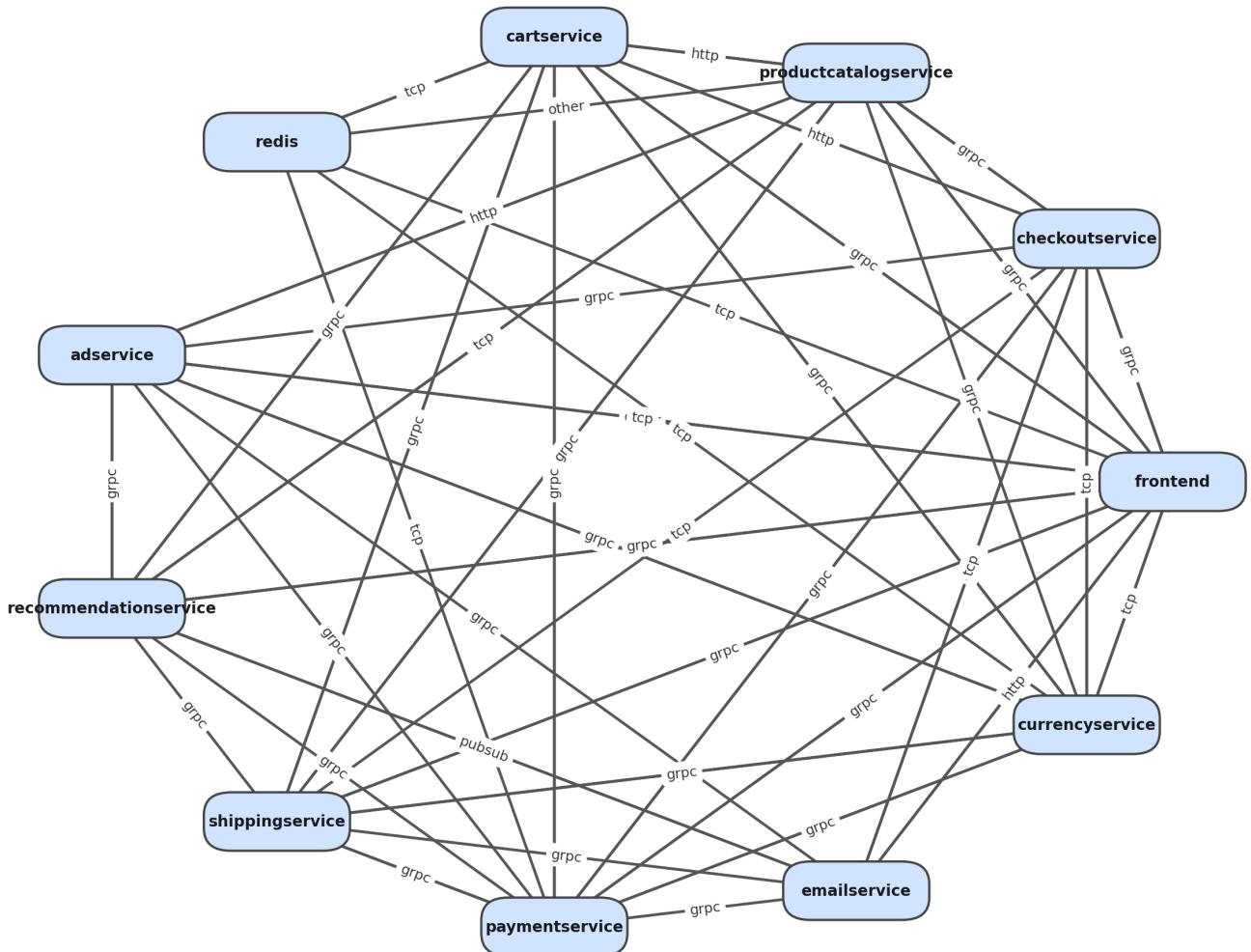


Рис. 3. Полученный график зависимостей микросервисов

указаны в явном виде в документации. Это говорит о том, что LLM способна учитывать контекст кода и делать выводы по сочетанию нескольких косвенных признаков.

Однако эксперимент также выявил ограничения. В некоторых случаях модель анализировала вспомогательные файлы, например main.go или тестовые файлы, и ошибочно определяла связи на основании их содержания. Это приводило к ложным выводам, особенно в тех ситуациях, когда код содержал абстракции, похожие на реальные сетевые вызовы.

Несмотря на это, результаты показывают, что метод подходит для предварительного анализа архитектуры и может использоваться в ситуациях, когда запуск системы невозможен или затруднён. Он подходит как для реверс-инжиниринга, так и для автоматического формирования документации.

Заключение

В статье представлена методика автоматического извлечения архитектурных взаимодействий микросервисов с помощью локальной LLM.

Результаты эксперимента показали, что даже при ограниченном объёме контекста модель способна корректно восстанавливать до 85 % связей, подтверждённых документацией.

Использование открытых моделей (например, Mistral 7B) позволяет выполнять подобный анализ на локальных машинах, что особенно актуально для компаний с повышенными требованиями к безопасности.

Литература

1. *Luo S., Xu H., Lu C.* An in-depth study of microservice call graph and runtime performance // IEEE Transactions on Parallel and Distributed Systems. – 2022. – Vol. 33, no. 12. – P. 3901–3914.
2. *Cerny T., Taibi D.* Static analysis tools in the era of cloud-native systems // In 4th International Conference on Microservices, 2022.
3. *Jelodar H., Meymani M., Razavi-Far R.* Large Language Models (LLMs) for Source Code Analysis: Applications, Models and Datasets // Journal of Computer Science and Technology. – 2025.
4. Google Cloud Platform. Microservices Demo: – GitHub, 2025. – URL: <https://github.com/GoogleCloudPlatform/microservices-demo> (дата обращения: 10.11.2025).
5. *Jiang A., Bair T., Du N. [et al.]* Mistral 7B: A Powerful Language Model with Long Context and Efficient Inference // arXiv preprint arXiv:2310.06825, 2023.
6. *Yang J., Jin H., Tang R., Han X., Feng Q., Jiang H., Yin B., Hu X.* Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. – 2023. – arXiv:2304.13712.
7. *Nam D. [et al.]* Using an LLM to Help With Code Understanding // Proceedings of the 2024 ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE'24). – 2024.
8. *Blyth S., Licorish S. A., Treude C., Wagner M.* Static Analysis as a Feedback Loop: Enhancing LLM-Generated Code Beyond Correctness // Empirical Software Engineering (Springer). – 2025.
9. *Husein R. A.* Large language models for code completion: A systematic review // Journal of Systems Architecture. – 2025.

АНАЛИЗ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ ПО ВВЕДЕНИЮ ПРИКОРМА И МОНИТОРИНГА ПИЩЕВОЙ АЛЛЕРГИИ У ДЕТЕЙ

Е. В. Ещенко

Воронежский государственный университет

Аннотация. В статье рассматриваются современные интеллектуальные системы поддержки принятия решений, применяемые для введения прикорма и мониторинга пищевой аллергии у детей. Анализируются существующие цифровые решения, их функциональные возможности, подходы к персонализации рекомендаций и методы фиксации реакций на продукты. Отдельное внимание уделено преимуществам таких систем, а также их текущим ограничениям. По результатам анализа формулируются требования к более эффективным интеллектуальным сервисам и направления их дальнейшего развития.

Ключевые слова: интеллектуальные информационные системы, введение прикорма детям, мониторинг пищевой аллергии, цифровые медицинские системы, персональные рекомендации по питанию, анализ пищевых реакций, отслеживание симптомов, дневник прикорма, дневник питания, система поддержки принятия решений.

Введение

Введение прикорма и своевременное выявление пищевой аллергии являются важными аспектами детского здоровья, требующими системного подхода и индивидуальных рекомендаций. Рост популярности мобильных приложений и цифровых сервисов способствует развитию интеллектуальных систем, позволяющих автоматизировать сбор данных, отслеживание реакций и формирование персонализированных рекомендаций.

Согласно рекомендациям ВОЗ [1], оптимальным временем начала введения прикорма является возраст около 6 месяцев, так как к этому времени потребности ребёнка в энергии и микроэлементах уже превышают возможности исключительно грудного молока. Согласно Национальной программе оптимизации вскармливания детей первого года жизни в Российской Федерации [2] в контексте современных представлений об оптимальном формировании пищевой толерантности введение продуктов прикорма целесообразно в возрасте не ранее 4 и не позже 6 месяцев. У детей с пищевой аллергией так же рекомендуется ориентироваться на эти сроки. Однако в каждом конкретном случае вопрос о времени введения прикорма решается индивидуально. При этом важны факторы безопасности: питание должно быть адекватным и безопасным, прикорм должен вводиться постепенно и в определенном порядке [3, 4].

В то же время риск пищевой аллергии у детей является серьёзной проблемой. Рекомендации по введению прикорма детям с возможной аллергической предрасположенностью во многом совпадают с таковыми для здоровых детей, однако особенно важно внимательно отслеживать реакции на новые продукты. Случаи аллергии на коровье молоко, яйца, орехи и другие продукты требуют более тщательного подхода и документирования через дневники питания и систематическое наблюдение [5].

С развитием цифровых технологий появляются интеллектуальные системы поддержки принятия решений (СППР), которые могут помочь родителям и медицинским работникам в процессе введения прикорма и мониторинга аллергий. Эти системы способны собирать данные о рационе, фиксировать симптомы и генерировать персонализированные рекомендации, используя алгоритмы машинного обучения либо экспертные правила [6]. Применение таких решений повышает безопасность, быстроту реакции и точность рекомендаций — особенно в случаях, когда ребёнок показывает признаки аллергии или входит в группу риска.

Тем не менее, несмотря на растущий интерес к цифровым медицинским сервисам, анализ показывает, что большинство существующих приложений для родителей являются преимущественно дневниками питания без интеллектуальной обработки данных. На сегодня распространены решения, не использующие сложные алгоритмы прогнозирования аллергии, что ограничивает их эффективность для задач поддержки клинически значимых решений.

В связи с этим важным направлением исследований является разработка интеллектуальных систем, которые будут интегрировать сбор данных, классификацию аллергенных рисков и предоставлять адаптивные рекомендации для конкретного ребенка.

1. Анализ существующих решений

Современный рынок цифровых приложений, связанных с прикормом и мониторингом пищевой аллергии, достаточно разнообразен, однако большинство представленных решений можно условно разделить на три крупные категории. Такой подход позволяет структурированно оценить текущее состояние технологий, выявить их сильные стороны и определить недостатки, которые в совокупности формируют нишу для разработки более интеллектуальных систем поддержки принятия решений (СППР).

1.1. Базовые дневники питания и прикорма

Наиболее распространённую группу составляют универсальные дневники для родителей, включающие функции записи кормлений, сна, роста и других бытовых параметров ребёнка. Эти приложения не фокусируются именно на аллергии или анализе данных; их основная цель — предоставить удобный инструмент для ведения ежедневных записей.

В качестве примеров таких приложений следует выделить:

- *Baby Daybook* — популярный дневник, позволяющий фиксировать кормления, прикорм, сон, настроение, рост и вес [7].
- *Sprout Baby* — приложение с календарём развития и модулем ввода прикорма ребенка [8].
- *The Baby Owner* — дневник с фокусом на фиксации любых событий ребенка.
- *BabyBear: Дневник малыша* — российское приложение с удобной фиксацией введённых продуктов и возможностью отмечать реакции [9].

Большинство подобных приложений обладают следующим функционалом:

- запись введённых продуктов (дата, количество, заметки);
- общий дневник ухода за ребёнком;
- напоминания о кормлениях и прикормах;
- базовые графики и отчёты;
- встроенные статьи по прикорму (без персонализации, общеобразовательные).

В качестве преимуществ таких приложений можно выделить следующее:

- простота и доступность — низкий порог входа;
- существование бесплатных версий;
- удобство в виде замены классических бумажных дневников;
- обширная функциональность с советами по уходу за ребёнком.

Однако у таких систем можно найти достаточно много недостатков:

- отсутствие алгоритмов анализа данных;
- общие рекомендации предоставляются без персонализации под конкретного ребенка;
- система служит только хранилищем данных, не давая помощи в принятии решений;
- непригодность для задач диагностики аллергии.

Эти приложения удобны в качестве дневника записей, но не решают задачи обнаружения закономерностей между введёнными продуктами и возникающими реакциями. В условиях повы-

шенного риска пищевой аллергии такая аналитическая составляющая является принципиально важной, поскольку именно она позволяет своевременно выявить потенциальные триггеры и подобрать безопасные альтернативы. Отсутствие автоматизированного анализа приводит к тому, что родителям приходится самостоятельно сопоставлять большое количество данных, что затрудняет принятие решений и повышает вероятность неверной интерпретации симптомов. Кроме того, такие системы не предлагают инструментов для прогнозирования возможных нежелательных реакций при введении новых продуктов, что ограничивает их клиническую полезность и снижает информативность в ситуациях, требующих высокой точности и индивидуального подхода.

1.2. Специализированные дневники аллергий и симптомов

Данная группа решений представляет собой более узкоспециализированный класс цифровых инструментов, ориентированных на мониторинг пищевой аллергии и фиксирование симптоматики, возникающей в ответ на введение новых продуктов. В отличие от универсальных дневников ухода за ребенком, такие системы обеспечивают расширенные возможности по структурированной регистрации клинических проявлений (кожные реакции, гастроинтестинальные симптомы, респираторные признаки и т. д.) и привязке их к конкретным пищевым триггерам. Однако, несмотря на высокую детализацию данных, большинство подобных приложений ограничивается функциями визуализации и базового учета, не реализуя полноценные методы интеллектуального анализа данных, что снижает их потенциал как инструментов поддержки клинических решений.

Таких систем существует уже меньше, в сравнении с обычными дневниками питания. Рассмотрим следующие системы:

– *SpoonfulONE* — приложение, ориентированное на профилактику аллергии посредством регулярного введения аллергенов; включает журнал питания и реакции [10].

– *MySymptoms Food Diary* — продвинутый дневник, который имеет богатый функционал по анализу аллергий и реакций. Показывает корреляции и ранжирует продукты по силе связи на симптом. Однако есть и минусы: анализ основан на статистике, а не на медицинских знаниях. Не дает прямых рекомендаций и ответа на вопрос «что делать дальше», только показывает связи. Родитель и/или врач должны сами интерпретировать результаты [11].

– *Solid — Baby Food Tracker* — приложение, которое позволяет отмечать продукты и симптомы аллергий и другие реакции, но не предоставляет алгоритмы анализа.

– *Allergy Amulet* — мобильное приложение включает дневник реакций и журнал триггеров, однако основной продукт представляет собой носимый датчик для выявления аллергенов. Приложение предоставляет дополнительный функционал [12].

– *Food Allergy Institute (FAI) Tracker* — специализированный инструмент, используемый некоторыми клиниками. Имеет фиксацию побочных реакций по шкале тяжести. В качестве минусов можно отметить, что не является массовым потребительским продуктом.

У систем такого типа обычно присутствует следующий функционал:

- детальная запись симптомов (сыпь, зуд, отёк, рвота, изменения стула);
- привязка симптомов к конкретным продуктам;
- визуализация реакций (календарь, хроника);
- генерирование отчётов для медицинского специалиста;
- добавление фотографий проявлений аллергии.

В качестве преимуществ таких систем можно выделить следующее:

- предоставление высокой точности и детализации по аллергическим симптомам;
- предоставление информации в структурном виде, в том числе предоставлении информации в более удобном виде для медицинских работников (графики, диаграммы и т. п.);
- формирование у родителей и медицинского персонала понимания связи между продуктами и реакциями детей на введённые продукты.

Однако такие системы зачастую не лишены недостатков:

- отсутствие механизмов прогнозирования новых реакций;
- необходимость самостоятельного анализа данных со стороны родителя или врача;
- отсутствие алгоритмов выявления скрытых закономерностей;
- ограниченность некоторых модулей для использования самими родителями без помощи медицинских работников.

Таким образом, специализированные дневники аллергии и симптомов занимают промежуточное положение между базовыми бытовыми дневниками питания и интеллектуальными системами поддержки принятия решений. Они обеспечивают высокую детализацию данных, связанную с динамикой аллергических проявлений и создают структурированную основу для последующего клинического анализа. Однако их функциональные возможности остаются ограниченными преимущественно регистрацией и визуализацией информации. Несмотря на расширенный набор параметров и ориентированность на значимые события, эти приложения, как правило, не осуществляют глубокой аналитической обработки, не выполняют автоматического выявления причинно-следственных связей и не формируют персонализированных рекомендаций для конкретного ребенка.

1.3. Интеллектуальные коммерческие системы с элементами поддержки принятия решений

Наиболее инновационным, но пока наименее распространённым сегментом цифровых решений являются системы с элементами экспертных методов и машинного обучения. Эти решения направлены не только на аккумулирование данных, но и на их систематическую интерпретацию, прогнозирование рисков аллергических реакций и формирование персонализированных рекомендаций по введению прикорма.

Далеко не все такие системы являются коммерческими продуктами, многие являются исследовательскими или используются на базе конкретных клиник.

Рассмотрим существующие коммерческие продукты:

- *Nurturey (Nurturey PinkBook)* — попытка создать целостную цифровую экосистему, использование элементов ИИ для анализа симптомов, однако система не специализирована на детском питании и аллергии, а базируется на здоровье ребенка в целом.
- *NALA Health («ABOARD» — AI-driven Platform)* — научный проект, направленный на разработку интеллектуальной платформы для пациентов с атопическим дерматитом и пищевой аллергией. Функционал достаточно богатый, используется компьютерное зрение для анализа кожных проявлений и тяжести атопического дерматита; имеет дневник питания и симптомов; использует алгоритмы машинного обучения для прогнозирования вспышек дерматита; имеет функционал генерации отчетов для медицинского персонала [13].
- *AllergySpot* — ИИ приложение для отслеживания пищевых реакций и вычисления паттернов по схеме «три дня до симптома, потом реакция» [14].

Коммерческие интеллектуальные системы пока немногочисленны, однако есть интересные варианты, и они уже показывают потенциал ИИ в мониторинге аллергий и поддержке решений. Их появление подчёркивает необходимость разработки более специализированных и клинически ориентированных инструментов.

1.4. Интеллектуальные научные системы с элементами поддержки принятия решений

В качестве примеров различных научных систем можно выделить:

- **Мультиагентные системы:** моделирование процесса введения прикорма через взаимодействие «агентов» — «Агент-аллерголог», «Агент-педиатр», «Агент-данные», совместно формирующих рекомендации.

– *Исследования по СППР в детской аллергологии* [4].

– *Специальные модули телемедицинских платформ*, предоставляющие рекомендации на основе анализа истории пациента.

– *Академические исследовательские системы*, использующие ассоциативные правила, байесовские сети или классификаторы для оценки пищевых рисков.

– *Системы на основе онтологий и экспертины правил*.

Такие системы обычно предоставляют достаточно обширные функциональные возможности, среди которых можно выделить:

– алгоритмический анализ с выявлением закономерностей «продукт реакция»;

– прогнозирование вероятности реакции на новые продукты;

– генерация персонализированных рекомендаций по следующему продукту прикорма;

– использование экспертной базы знаний, основанной на рекомендациях Союза педиатров России, Всемирной Организации Здравоохранения и современных научных источниках;

– адаптивное изменение плана введения прикорма в зависимости от истории наблюдений и реакции ребёнка.

Данные системы имеют ряд преимуществ по сравнению с классическими приложениями, которыми являются дневники введенных продуктов, симптомов и аллергенов:

– высокий уровень персонализации и адаптивности рекомендаций;

– способность выявлять сложные зависимости, незаметные при ручной обработке данных;

– возможность предупреждения развития аллергии и оптимизации стратегии прикорма;

– научная обоснованность на базе проверенных моделей и протоколов.

Однако зачастую данные системы довольно сложны и недоступны широкому кругу потребителей. В связи с этим, можно выделить ряд недостатков и ограничений:

– ограниченная доступность, так как большинство систем остаются прототипами;

– редкая интеграция в массовую практику здравоохранения;

– необходимость накопления значительных объёмов данных для корректного анализа;

– сложность интерпретации алгоритмов для родителей без специальной подготовки.

В качестве вывода по данным систем можно сказать, что данная категория интеллектуальных систем с элементами СППР представляет собой наиболее перспективное направление развития цифровых решений в области прикорма и мониторинга пищевой аллергии. Несмотря на потенциал повышения точности, безопасности и персонализации рекомендаций, на текущий момент данные решения остаются узконаправленными, ограниченными исследовательскими прототипами или интегрированными только в специализированные медицинские платформы, что подчёркивает необходимость дальнейших разработок и внедрений.

2. Сравнительный анализ существующих типов систем

Для наглядного сопоставления рассмотренных категорий цифровых решений по введению прикорма и мониторингу пищевой аллергии проведён сравнительный анализ их функциональных возможностей, интеллектуальной поддержки и степени персонализации.

В табл.1 представлены ключевые критерии, отражающие различия между базовыми дневниками ухода за ребёнком, специализированными дневниками аллергии и симптомов, а также интеллектуальными системами поддержки принятия решений (СППР), включая коммерческие и исследовательские прототипы.

На основе представленных в табл.1 данных была построена диаграмма (рис. 1), наглядно демонстрирующая распределение ключевых показателей и взаимосвязи между ними. Данный графический анализ позволяет более наглядно оценить тенденции и выявить закономерности, которые трудно уловить при простом просмотре числовых значений.

Таблица 1

Критерий	Базовые дневники питания	Специализированные дневники аллергий	Интеллектуальные СППР
Основная функция	Трекинг ухода за ребенком и питания	Фиксация симптомов и связь с продуктами, визуализация корреляций	Алгоритмический анализ данных, прогнозирование рисков, персонализированные рекомендации
Интеллектуальный анализ	Отсутствует	Минимальный (визуализация, базовая статистика)	Высокий (машинное обучение, экспертные правила, онтологии)
Персонализация рекомендаций	Отсутствует	Отсутствует	Как правило, присутствует, на основе истории ребенка и данных наблюдений
Фиксация реакций (симптомы)	Базовая (общее описание, заметки)	Детальная (сыпь, зуд, отёк, рвота, фото)	Детальная и связанная с алгоритмами, с оценкой риска и приоритетов
Прогнозирование рисков	Отсутствует	Отсутствует	Может присутствовать, для оценки вероятности реакции на новые продукты
Интеграция медицинских знаний	Общие статьи по прикорму и уходу	Частичная (статьи и протоколы по аллергии)	Полная (экспертные базы союза педиатров РФ, ВОЗ, правила и модели СППР)
Полеза для врача	Низкая (данные неструктурированные, трудны для анализа)	Средняя (можно генерировать отчет и увидеть корреляции)	Высокая (структурные данные, прогнозы, рекомендации для клинициста)
Основное преимущество	Удобство, универсальность, доступность	Фокус на аллергии и высокая детализация симптомов	Поддержка принятия решений, персонализированные рекомендации, аналитика
Основной недостаток	Нет интеллектуального анализа, только фиксация данных	Нет прогнозирования и автоматических рекомендаций	Ограниченнная доступность, сложность интерпретации, требует данных и подготовки пользователей
Доступность и аудитория	Массовые пользователи, родители	Ограниченный круг, родители детей с аллергией, клиники	В основном исследовательские прототипы, профессиональные медицинские платформы
Примеры приложений и систем	Baby Daybook, Sprout Baby, The Baby Owner, BabyBear: дневник малыша	SpoonfulONE, MySymptoms, Solid, Allergy Amulet, Food Allergy Institute Tracker	Nurturey PinkBook, NALA Health, мультиагентные прототипы, СППР научных исследований

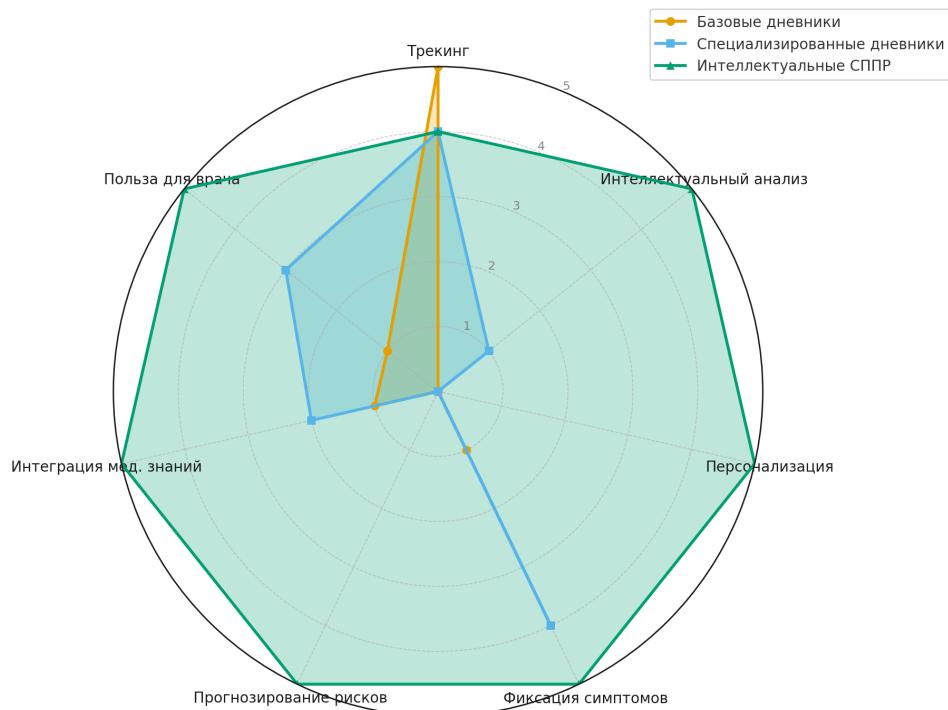


Рис. 1. Визуальная сравнительная характеристика цифровых решений по прикорму и мониторингу пищевой аллергии у детей

Заключение

Проведённый анализ существующих цифровых решений в области введения прикорма и мониторинга пищевой аллергии демонстрирует существенный разрыв между широкодоступными потребительскими приложениями и уровнем интеллектуальности, который необходим для поддержки клинически значимых решений. Основной рынок сегодня представлен базовыми дневниками ухода за ребёнком и специализированными дневниками аллергии, которые, несмотря на удобство использования и высокую детализацию данных, по сути, остаются инструментами фиксации событий. Они не позволяют автоматически выявлять причинно-следственные связи между продуктами и реакциями на них, не обладают механизмами прогнозирования новых рисков и не формируют персонализированных рекомендаций.

На фоне этого интеллектуальные системы с элементами машинного обучения и экспертных правил представляют собой наиболее перспективное направление развития, однако такие решения находятся преимущественно на стадии исследовательских прототипов или интегрированы в узкоспециализированные клинические платформы. Коммерческих продуктов, способных обеспечить глубокий анализ, персонализацию плана прикорма и прогнозирование аллергических рисков, чрезвычайно мало. Это подчёркивает высокую актуальность разработки комплексной системы, ориентированной на индивидуальные особенности ребёнка, историю введённых продуктов и динамику симптомов.

Создание подобной системы позволило бы не только улучшить безопасность процесса введения прикорма, но и повысить качество медицинского сопровождения, предоставляя родителям и врачам аналитически обоснованные рекомендации по выбору следующего продукта, его дозировке и возможным альтернативам. Кроме того, использование современных методов машинного обучения открывает возможности раннего выявления паттернов, предшествующих аллергическим реакциям, и формирования превентивных стратегий питания.

В рамках дальнейших этапов исследования планируется рассмотреть архитектуру возможной интеллектуальной системы, определить набор необходимых данных, формализовать

клинические правила и оценить потенциал применения прогностических алгоритмов. Таким образом, разработка СППР на базе экспертных правил и машинного обучения может стать значимым вкладом в область цифровой педиатрии и обеспечить качественно новый уровень персонализации при введении прикорма и мониторинге пищевой аллергии.

Литература

1. Всемирная организация здравоохранения (ВОЗ). Дополнительное кормление детей грудного и раннего возраста 6–23 месяцев / ВОЗ. – 2023. – URL: <https://www.who.int/publications/item/9789240081864> (дата обращения: 10.11.2025).
2. Национальная программа оптимизации вскармливания детей первого года жизни в РФ / ред. А. А. Баранова, В. А. Тутельяна. – Москва: Союз педиатров России, 2024. – 112 с.
3. Всемирная организация здравоохранения (ВОЗ). Руководящие принципы по дополнительному кормлению детей, находящихся на грудном вскармливании / ВОЗ. – 2003. – URL: <https://www.who.int/publications/item/9275124604> (дата обращения: 10.11.2025).
4. Рекомендации по введению аллергенных продуктов у недоношенных детей. – PubMed. – URL: <https://pubmed.ncbi.nlm.nih.gov/34371985/> (дата обращения: 10.11.2025).
5. Леонард С. А. [и др.] Эмпирические рекомендации по выбору продуктов для введения прикорма у детей с аллергией / S. A. Leonard [et al.] // World Allergy Organization Journal. – 2018. – DOI: 10.1186/s40413-017-0182-z. – С. 15–23.
6. Nasarian E., Alizadehsani R., Acharya U. R., Tsui K.-L. Проектирование интерпретируемой ML системы для повышения доверия в здравоохранении – arXiv, 2023. – URL: <https://arxiv.org/abs/2311.11055> (дата обращения: 10.11.2025).
7. Baby Daybook — All-in-One Baby Tracker App / Baby Daybook. – URL: <https://babydaybook.app/> (дата обращения: 10.11.2025)
8. Sprout Baby Tracker — приложение для отслеживания кормления ребенка / Sprout-Apps LLC. – URL: <https://sprout-apps.com/apps/baby-tracker/> (дата обращения: 10.11.2025).
9. BabyBear: Дневник малыша — приложение для ведения дневника питания и здоровья ребёнка. – URL: <https://apps.apple.com/ru/app/babybear> (дата обращения: 11.11.2025).
10. mySymptoms Food Diary — Food & Symptom Tracker / SkyGazer Labs Ltd. – 2025. – URL: <https://www.mysymptoms.net> (дата обращения: 11.11.2025).
11. AllergyHunter — online-инструмент для ведения дневника питания и симптомов, поиска паттернов аллергических реакций / AllergyHunter.org. – 2025. – URL: <https://www.allergyhunter.com> (дата обращения: 11.11.2025)
12. Allergy Amulet — персональный сенсор для выявления аллергенов + companion-приложение для учёта результатов / Allergy Amulet. – 2025. – URL: <https://www.allergyamulet.com> (дата обращения: 11.11.2025).
13. NALA Health. NALA — цифровой помощник для детей / NALA.care. – 2025. – URL: <https://nala.care> (дата обращения: 11.11.2025).
14. AllergySpot — AI-приложение для отслеживания пищевых реакций и вычисления паттернов по процессу «три дня до симптома / реакция» / AllergySpot. – 2025. – URL: <https://allergyspot.app> (дата обращения: 11.11.2025).

АНАЛИЗ И КЛАССИФИКАЦИЯ КАРДИОЛОГИЧЕСКИХ ПАТОЛОГИЙ ПО ДАННЫМ 12-КАНАЛЬНОЙ ЭКГ С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

И. И. Жмурко

Воронежский государственный университет

Аннотация. Сердечно-сосудистые заболевания остаются ведущей причиной смертности в мире, а электрокардиография — стандартным методом диагностики кардиологических патологий. Ручная интерпретация ЭКГ требует высокой квалификации и остаётся субъективной, особенно в сложных случаях. Разработка автоматических методов на основе машинного обучения актуальна для повышения эффективности диагностики и доступности кардиологической помощи. В работе сравниваются классические методы (Random Forest) и рекуррентные нейронные сети (Bidirectional LSTM) для задачи классификации мультилейбл патологий на ЭКГ с учётом дисбаланса классов. Полученные результаты позволяют выявить ограничения текущих подходов и обосновать необходимость дальнейшего изучения более современных архитектур.

Ключевые слова: электрокардиография, классификация патологий, мультилейбл обучение, глубокое обучение, Random Forest, LSTM, машинное обучение, кардиология, дисбаланс классов, медицинская диагностика, 12-канальная ЭКГ.

Введение

Заболевания сердечно-сосудистой системы (ССЗ) остаются ведущей причиной смертности в мире, унося примерно 19.8 млн жизней ежегодно[1]. Электрокардиография (ЭКГ) — стандартный неинвазивный метод диагностики, используемый при первичном обследовании, скрининге и мониторировании. Однако традиционная интерпретация ЭКГ требует опыта и остаётся субъективной. Дефицит кардиологических специалистов в отдаленных регионах делает ручную интерпретацию неприемлемо медленной для массового профилактического обследования. В последнее десятилетие машинное обучение активно применяется в медицинской диагностике [2]. Для анализа ЭКГ используются как классические методы (Random Forest, SVM), так и архитектуры глубокого обучения (CNN, RNN, LSTM)[3]. Каждый метод имеет ограничения: Random Forest не учитывает временную структуру, CNN работает лучше на коротких паттернах, LSTM обрабатывают последовательность пошагово (затрудняя параллелизацию и захватывание долгосрочных зависимостей при 5000+ отсчётах в ЭКГ) и лишены встроенной интерпретируемости. Трансформеры и архитектуры на основе самовнимания показали превосходство над LSTM на задачах обработки последовательностей благодаря параллельной обработке, лучшему захватыванию долгосрочных зависимостей и встроенной интерпретируемости через attention weights[4]. Однако классификация кардиологических патологий осложнена мультилейбл природой (каждая ЭКГ может содержать несколько диагнозов) и выраженным дисбалансом классов (одни патологии в 1000+ примерах, другие в 1-2 примерах). Цель работы — исследовать и сравнить классические методы (Random Forest) и рекуррентные нейронные сети (Bidirectional LSTM) для классификации ЭКГ в условиях мультилейбл задачи и выраженного дисбаланса классов с полным проведением обработки данных, анализом производительности моделей и выявлением ограничений LSTM для обоснования необходимости применения трансформеров.

1. Исходные данные и описание задачи

Исследование базируется на датасете PTB-XL[5] — публичном архиве кардиологических данных, содержащем 12-канальные ЭКГ записи пациентов. Для исследования из полного датасета выбраны записи с 2100+ пациентов. Каждая запись включает 12-канальные ЭКГ сигналы (отведения: I, II, III, aVR, aVL, aVF, V1-V6), каждый канал содержит 5000 отсчётов в формате NumPy. Целевые переменные представлены как список индексов патологий в мультилайбл формате. Датасет содержит 73 уникальных кардиологических диагноза, при этом среднее число диагнозов на пациента — 3.29 (диапазон 1-8). Из полного набора признаков PTB-XL (включающего возраст, пол, рост, вес, SCP-коды и другие клинические параметры) для этого исследования оставлены признаки: демографические данные (возраст, пол, рост, вес), SCP-коды и категория электрической оси сердца (heart axis). Датасет разделён на 10 стратифицированных фолдов кросс-валидации. Часть исходных данных показана в табл. 1.

Таблица 1

Фрагмент исходных данных

patient id	filename	labels	age	height	weight	sex	strat fold	scp-codes	heart axis
18792	00009_hr	[37, 54]	55	NaN	70.0	0	10	{'NORM': 100.0, 'SR': 0.0}	None
13619	00034_hr	[37, 58]	56	NaN	NaN	0	9	{'AFLT': 100.0}	None
11315	00043_hr	[22, 37]	25	NaN	63.0	1	8	{'NORM': 100.0, 'SR': 0.0}	None
18153	00052_hr	[1, 37, 54]	35	NaN	82.0	0	1	{'IRBBB': 100.0, 'SR': 0.0}	None
16063	00057_hr	[37, 54]	26	NaN	93.0	0	10	{'NORM': 100.0, 'SR': 0.0}	None

2. Разведочный анализ данных

2.1. Демографический профиль и антропометрические характеристики

Датасет содержит записи пациентов среднего и старшего возраста со средним возрастом 62 ± 15 лет и медианой 61 год, что соответствует группе с повышенным риском сердечно-сосудистых заболеваний. Диапазон возрастов варьируется от 0 до 120+ лет, однако значения выше 120 лет являются очевидными ошибками ввода (рис. 1). По половому признаку датасет хорошо сбалансирован: 52.1 % мужчин и 47.9 % женщин, что важно для объективного анализа кардиологических данных. Распределение по возрастным группам показывает, что более половины пациентов — люди старше 60 лет, что логично для кардиологического датасета и отражает типичную демографическую структуру пациентов с сердечно-сосудистыми заболеваниями.

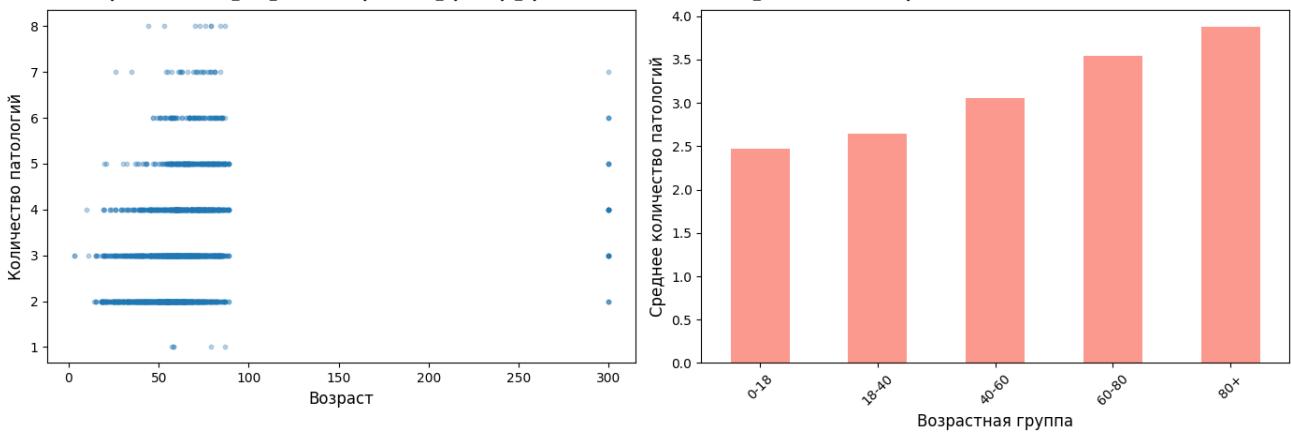


Рис. 1. Точечная диаграмма и гистограмма возраста (с аномалиями > 120)

Что касается антропометрических данных, то показатели роста и веса содержат значительное количество пропусков — приблизительно 50 % каждый. Несмотря на это, на полных данных наблюдается умеренная положительная корреляция между ростом и весом с коэффициентом корреляции Пирсона $r = 0.612$. Средний индекс массы тела (BMI) составляет 25.18 — по классификации ВОЗ это «избыточная масса тела» (рис. 2), что характерно для пациентов с кардиопатологией и указывает на распространённость превышения нормы веса в данной группе.

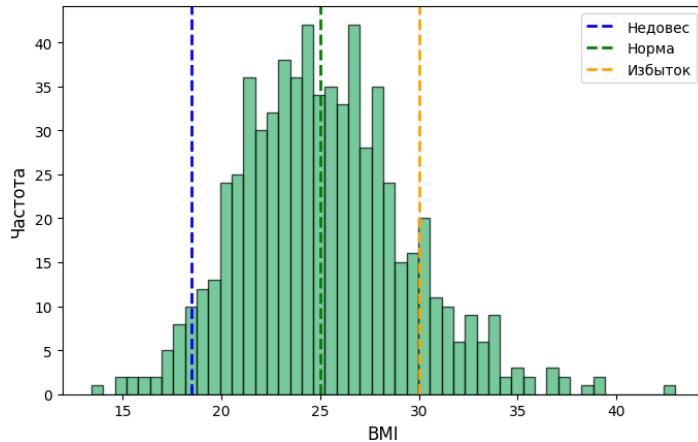


Рис. 2. Распределение BMI

2.2. Анализ целевых переменных и клинических параметров

Анализ целевых переменных выявил, что среднее число диагнозов на пациента составляет 3.29, с диапазоном от 1 до 8 диагнозов. Большинство пациентов (более 60 %) имеют 3–4 диагноза одновременно, что отражает реальность клинической практики, где кардиологические патологии часто сопутствуют друг другу и развиваются из общих факторов риска. Однако датасет страдает от выраженного дисбаланса классов: топ-2 патологии встречаются в более чем 1200 примерах каждая, в то время как более половины диагнозов представлены менее чем 50 примерами (рис. 3). Это создаёт серьёзную проблему при обучении моделей, поскольку стандартные алгоритмы будут смещены в сторону частых классов и практически не смогут обуиться на редких примерах.

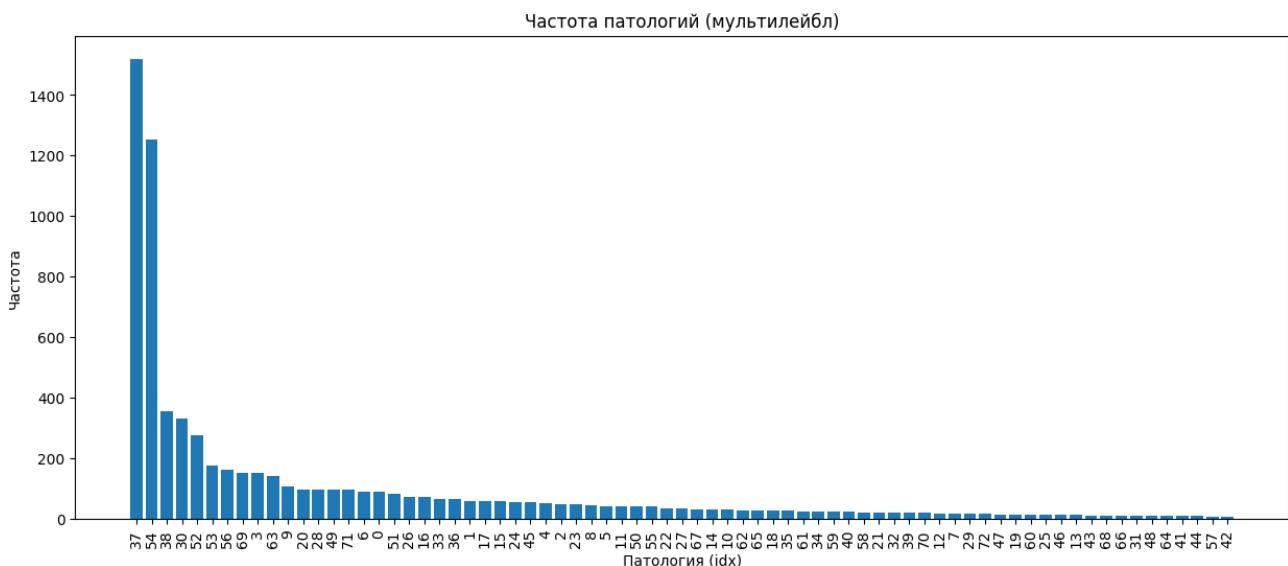


Рис. 3. Распределение количества диагнозов

Дополнительно следует отметить, что при интерпретации ЭКГ кардиологи используют SCP-коды, каждому из которых присваивается балл уверенности от 0 до 100. Медиана уверенности составляет 35, среднее значение — 46.5, что свидетельствует о большой доле пограничных и сложных клинических случаев, характерных для реальной практики. Анализ электрической оси сердца (heart_axis) показал следующее распределение: промежуточная позиция (норма) — 34.5 %, отклонение влево — 16.8 %, выраженное отклонение влево — 5.6 %; остальные категории — менее 2 % каждой.

2.3. Взаимосвязи между диагнозами

Наконец, анализ взаимосвязей между диагнозами выявил важные клинические паттерны. Матрица корреляции выявила группы патологий, часто встречающихся совместно. Например, высокая корреляция между метками «Блокада передней ветви левой ножки п. Гиса» и «Отклонение электрической оси влево» ($r = 0.593$) указывает на общий кардиологический синдром. Аналогично, связь между «Неопределённой ЭОС» и «ЭКГ типа SI-SII-SIII, анализ положения ЭОС не проводится» ($r = 0.416$) отражает технически взаимосвязанные особенности интерпретации сигнала. На рис. 4 именно эти пары патологий с наивысшими значениями положительной корреляции выделены тёмно-зелёным цветом. Матрица совместной встречаемости подтвердила наиболее частые сочетания: метки «Нормальное положение ЭОС» и «Синусовый ритм» встречаются вместе 718 раз, что указывает на клинически ассоциированные состояния.

С другой стороны, обнаружено 1484 пары взаимоисключающих диагнозов, что обусловлено медицинской несовместимостью. Например, нормальное положение электрической оси сердца не может сочетаться с её отклонениями, а различные типы сердечных ритмов физически не могут присутствовать одновременно. Эти закономерности имеют практическое значение: они позволяют проверять корректность предсказаний моделей, предотвращать медицинские невозможные комбинации диагнозов и лучше понимать структуру кардиологических заболеваний.

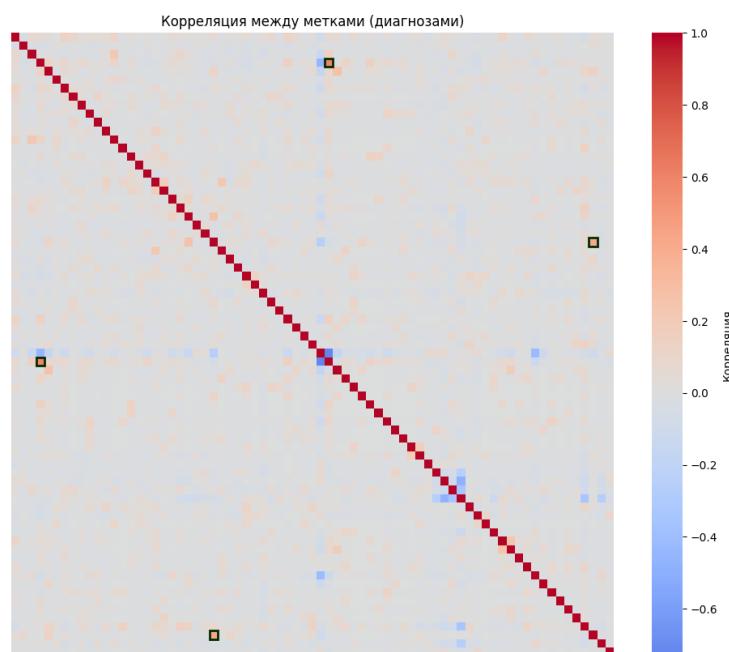


Рис. 4. Матрица корреляции между диагнозами

3. Методология

3.1. Предобработка данных

Реальные медицинские датасеты требуют тщательной предобработки. Значения возраста > 120 лет заменены на медиану (61 год) для устойчивости к выбросам. Пропуски в признаках `height` и `weight` заполнены медианными значениями. Для каждого пациента рассчитан индекс массы тела (BMI) путём деления веса (в кг) на квадрат роста (в м). SCP-коды, хранящиеся как JSON-словарь, распакованы с помощью `pd.json_normalize()`, каждый уникальный код преобразован в отдельный числовой признак со значением баллов уверенности кардиолога. Переменная `heart_axis` закодирована в 8 бинарных столбцов с использованием one-hot encoding, а пропущенные значения заполнены категорией 'UNKNOWN'. Результат: исходные 10 признаков расширены до 100+ признаков после кодирования (схема процесса показана на рис. 5).



Рис. 5. Блок-диаграмма предобработки

Для нейросетевых моделей применена $z-score$ нормализация ЭКГ независимо для каждого канала и образца: $x_{norm} = \frac{x - \mu}{\sigma}$, где μ — средняя величина, σ — стандартное отклонение.

3.2. Архитектуры моделей

Для решения задачи реализованы две модели, позволяющие проследить эволюцию подходов от классических методов машинного обучения к глубокому обучению.

Первая модель — Random Forest с MultiOutputClassifier (100 деревьев) — служит базовой линией для сравнения. На вход подаётся объединение развёрнутых ЭКГ сигналов (60 000 признаков) и табличных данных (~81 признак); для каждого из 73 диагнозов обучается независимый классификатор. Модель проста в реализации и интерпретируется легко, однако полностью игнорирует временную структуру ЭКГ, рассматривая каждый отсчёт как независимый признак, что создаёт риск переобучения при такой высокой размерности входных данных.

Вторая модель использует двухветвевую архитектуру: ЭКГ-ветка состоит из двух слоёв Bidirectional LSTM (256 и 128 нейронов) с Dropout (0.3 и 0.4), табличная ветка — из одного Dense слоя (32 нейрона, ReLU, Dropout 0.2). Обе ветки объединяются через Concatenate, проходят через Dense слой (64 нейрона) и выходной слой (73 нейрона, Sigmoid). Ключевое преимущество — двунаправленная обработка: модель видит контекст как «вперёд», так и «назад» по времени, что критично для интерпретации ЭКГ паттернов (QRS, ST-сегменты). Модель обучается с Adam и Binary Crossentropy в течение 50 эпох (`batch_size = 32`). Основные ограничения LSTM: последовательная обработка 5000 отсчётов затрудняет параллелизацию, возможна потеря долгосрочных зависимостей в длинных последовательностях, и отсутствует встроенная интерпретируемость — сложно понять, какие части ЭКГ влияют на предсказание.

4. Метрики оценки качества

Для оценки производительности моделей в задаче мультилнейбл классификации использованы специализированные метрики, адаптированные к особенностям данной задачи. Полное описание метрик и их интерпретация представлены в табл. 2.

Таблица 2
Метрики мультилнейбл классификации

Метрика	Описание	Диапазон
Hamming Loss	Доля неправильно предсказанных меток	0-1 (↓ лучше)
Exact Match Ratio	% образцов, где ВСЕ метки правильны	0-1 (↑ лучше)
F1-score (micro)	Глобальное F1 по всем классам (частые классы имеют больший вес)	0-1 (↑ лучше)
F1-score (macro)	F1 усреднено по всем классам (все классы имеют равный вес)	0-1 (↑ лучше)
Precision (micro)	$\frac{TP}{(TP + FP)}$ глобально	0-1 (↑ лучше)
Recall (micro)	$\frac{TP}{(TP + FN)}$ глобально	0-1 (↑ лучше)
ROC-AUC (micro)	Площадь под ROC-кривой	0-1 (↑ лучше)

Использование нескольких метрик необходимо, так как одна метрика не может полностью охарактеризовать качество модели в мультилнейбл задаче, особенно при дисбалансе классов.

5. Результаты экспериментов

Результаты тестирования представлены в табл. 3.

Таблица 3
Результаты тестирования моделей

Метрика	Random Forest	Bi-LSTM
Hamming Loss	0.036	0.033
Exact Match Ratio	0.166	0.147
F1 (micro)	0.487	0.546
F1 (macro)	0.032	0.121
Precision (micro)	0.678	0.688
Recall (micro)	0.3797	0.453
ROC-AUC (micro)	–	0.887

Заключение

Работа выявила несколько ключевых результатов при анализе кардиологического датасета PTB-XL. Датасет содержит реальные клинические данные со всеми присущими сложностями: аномалии в возрасте (значения > 120 лет), пропуски в антропометрических характеристиках (~50%) и выраженный дисбаланс классов (редкие диагнозы представлены 1–10 примерами, частые — 1000+ примерами). Важной особенностью задачи является то, что одна ЭКГ может од-

новременно содержать несколько диагнозов (в среднем 3.29), что требует применения специализированных метрик для многолейбловой классификации (Hamming Loss, Exact Match Ratio, F1-macro/micro) и функций потерь (Binary Crossentropy вместо Categorical Crossentropy). Кроме того, анализ закономерностей выявил 1484 взаимоисключающие пары диагнозов, которые никогда не встречаются вместе — например, нормальная электрическая ось сердца не может сочетаться с её отклонениями, а различные ритмы не могут присутствовать одновременно. Это отражает высокую структурированность медицинских знаний и открывает возможность явного учёта таких ограничений при декодировании выходов модели.

В результате экспериментов установлено, что Bidirectional LSTM превосходит Random Forest по F1-score (micro) на 5.9 процентных пункта ($0.487 \rightarrow 0.546$), что подтверждает преимущество архитектур, учитывающих временную структуру ЭКГ. При этом точность (Precision) остаётся высокой у обеих моделей ($0.678 \rightarrow 0.688$), в то время как Recall значительно улучшается ($0.3797 \rightarrow 0.453$), что отражает лучшую способность LSTM находить положительные примеры. Вместе с тем, стоит подчеркнуть, что LSTM обладают фундаментальными ограничениями: пошаговая обработка 5000-отсчётной последовательности замедляет обучение и затрудняет параллелизацию, а также повышается вероятность потери долгосрочных зависимостей по мере увеличения длины последовательности, кроме того, модель фактически остаётся «чёрным ящиком» с точки зрения клинической интерпретируемости.

Следует отметить, что дисбаланс меток остаётся критической проблемой: значительное расхождение между Micro F1 (0.546) и Macro F1 (0.121) показывает, что редкие диагнозы предсказываются с F1 порядка 0.05–0.12, тогда как частые достигают $F \approx 0.5 – 0.6$. Следует также отметить, что высокое значение $ROC - AUC = 0.887$ для LSTM свидетельствует о хорошей способности различать патологические и непатологические состояния, однако преимущество проявляется преимущественно для часто встречающихся диагнозов.

Исходя из вышеперечисленного, результаты указывают на целесообразность перехода к архитектурам на основе механизма самовнимания (трансформерам), которые способны обеспечить параллельную обработку всей 5000-отсчётной последовательности, более эффективное захватывание долгосрочных зависимостей благодаря attention-механизму, а также встроенную интерпретируемость через карты attention weights, демонстрирующие, какие фрагменты ЭКГ наиболее значимы для каждого диагноза.

Литература

1. World Health Organization. Cardiovascular Diseases (CVDs): fact sheet. – 2021. – URL: [https://www.who.int/ru/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/ru/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) (дата обращения: 25.11.2025).
2. Rajkomar A. Scalable and accurate deep learning with electronic health records / A. Hardt, M. D. Howell [et al.] // npj Digital Medicine. – 2018. – Vol. 1, № 1. – P. 1–10.
3. Kachuee M., Fazeli S., Sarrafzadeh M. ECG Heartbeat Classification: A Deep Transferable Representation / arXiv:1805.00794 [cs.CY]. – 2018. – URL: <https://arxiv.org/abs/1805.00794> (дата обращения: 25.11.2025).
4. Vaswani A. Attention is all you need / A. Vaswani, N. Shazeer, N. Parmar [et al.]. – 2017. – URL: <https://arxiv.org/abs/1706.03762> (дата обращения: 25.11.2025).
5. Wagner P. PTB-XL, a large publicly available electrocardiography dataset (version 1.0.3) / P. Wagner, N. Strodthoff, R. Bousseljot, W. Samek, T. Schaeffter. – PhysioNet, 2022. – URL: <https://doi.org/10.13026/kfzx-aw45> (дата обращения: 25.11.2025).

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В 2025 ГОДУ: ТЕНДЕНЦИИ РАЗВИТИЯ И ПЕРСПЕКТИВЫ

А. А. Захарова

Воронежский государственный университет

Аннотация. В статье представлен всесторонний анализ текущего состояния искусственного интеллекта в 2025 году, рассматриваются последние разработки, тенденции и результаты исследований из новейшей академической и отраслевой литературы. Благодаря систематическому обзору современных источников в этом исследовании определены ключевые достижения в области генеративного ИИ, машинного обучения и внедрение ИИ в различных секторах. Анализ показывает значительный прогресс в области интеллектуальных возможностей ИИ, более широкого внедрения на предприятиях и возникающих проблем в области управления ИИ и этики. Основные результаты показывают, что в настоящее время 78 % организаций используют ИИ. В одной бизнес-функции генеративный ИИ находит широкое практическое применение, поднимая важные вопросы о сотрудничестве человека и ИИ и влиянии на общество. В этом всеобъемлющем обзоре обобщены выводы из 25 первоисточников, включая рецензируемые статьи, отраслевые отчеты и эмпирические исследования, чтобы представить целостное представление о траектории развития искусственного интеллекта в 2025 году.

Ключевые слова: Искусственный интеллект, генеративный ИИ, машинное обучение, тенденции 2025 года, внедрение ИИ.

Введение

2025 год знаменует собой поворотный момент в развитии искусственного интеллекта, характеризующийся быстрым технологическим прогрессом и широким внедрением в различных отраслях промышленности [1]. По мере приближения к 2025 году ландшафт искусственного интеллекта значительно эволюционировал — от экспериментальных прототипов до готовых к производству систем, которые демонстрируют беспрецедентные возможности в области мышления, креативности и решения проблем [2]. В этом обзоре литературы рассматривается текущее состояние исследований и внедрения искусственного интеллекта, анализируются последние разработки, возникающие тенденции и растет количество научных и отраслевых исследований, которые формируют наше понимание траектории развития ИИ.

Важность этого анализа заключается в беспрецедентных темпах развития и внедрения ИИ, которые будут наблюдаться в 2025 году. В отличие от предыдущих лет, отмеченных постепенным прогрессом, 2025 год ознаменовался прорывными разработками в нескольких областях искусственного интеллекта одновременно, от расширенных возможностей логического мышления до сложных мультимодальных систем [3]. Это сближение технологического прогресса с практической реализацией предоставляет уникальную возможность оценить как эффективность, так и текущие достижения и будущее применение систем искусственного интеллекта.

Мотивация для проведения этого всестороннего обзора обусловлена необходимостью понять, каким образом технологии искусственного интеллекта вышли за рамки экспериментальных этапов и стали неотъемлемыми компонентами бизнес-процессов, научных исследований и использований в повседневной жизни [4]. Быстрое развитие возможностей искусственного интеллекта, особенно в области генеративного искусственного интеллекта и автономных систем, требует тщательного изучения современной литературы для выявления тенденций, оценки последствий и прогнозирования будущих изменений [5].

Кроме того, в 2025 году глобальная конкурентная среда в области искусственного интеллекта претерпела значительные изменения, последствия которых выходят за рамки технических возможностей и охватывают экономические, стратегические и геополитические аспекты [1]. Понимание этой динамики требует систематического анализа различных источников, начиная от академических исследований и заканчивая исследованиями по внедрению в промышленности.

1. Справочная информация по обзору литературы

1.1. Эволюция направления исследований в области искусственного интеллекта

Траектория исследований ИИ претерпела существенную эволюцию вплоть до 2025 года. Исторический анализ показывает переход от узких приложений, ориентированных на конкретные задачи, к более общим, многоцелевым системам, способным решать сложные, мультимодальные задачи [6]. Эта эволюция была особенно заметна при разработке больших языковых моделей и систем генеративного ИИ, которые превратились из экспериментальных курьезов в практические инструменты с широким спектром применения [7].

Недавняя литература указывает на то, что фокус исследований в области ИИ сместился в сторону разработки системы, которые могут интегрировать множество возможностей в унифицированные платформы, которые представляют собой фундаментальное архитектурное изменение в дизайне систем искусственного интеллекта [8]. Этот сдвиг отражает растущее признание того факта, что наиболее ценные приложения ИИ получаются в результате объединения различных возможностей ИИ, а не изолированной оптимизации отдельных компонентов.

1.2. Методологические подходы в современных исследованиях искусственного интеллекта

Современные исследования ИИ используют все более сложные методологические подходы, сочетающие теоретические достижения с эмпирической проверкой [9]. В литературе отмечается растущий акцент на воспроизводимости, сравнительном анализе и систематической оценке систем ИИ по нескольким параметрам, включая производительность, безопасность и влияние на общество [10].

Методологии исследований также эволюционировали для решения уникальных задач, связанных с генеративными системами искусственного интеллекта, включая разработку новых систем оценки, которые могут оценивать творческий подход, фактическую точность и возможность неправильного использования. Эти методологические инновации имеют решающее значение для поддержания научной строгости в быстро развивающейся области, где традиционные подходы к оценке могут оказаться недостаточными.

2. Текущее состояние технологий искусственного интеллекта в 2025 году

2.1. Достижения в области искусственного интеллекта

Наиболее значительным достижением в области технологий искусственного интеллекта в 2025 году стало продолжающееся развитие систем генеративного искусственного интеллекта [8]. Недавние исследования показывают, что генеративный ИИ вышел за рамки экспериментальных приложений и стал неотъемлемой частью бизнес-процессов. Интеграция множества возможностей искусственного интеллекта в унифицированные системы представляет собой серьезный архитектурный сдвиг, а такие разработки, как ожидаемая система GPT-5, призваны

объединить логику, мультимодальность и понимание долгосрочного контекста в единую, более функциональную платформу [7].

Исследование, опубликованное в 2025 году, демонстрирует, что генеративные системы искусственного интеллекта достигли значительных улучшений производительности во многих областях [8]. Исследования показывают, что реакции, генерируемые искусственным интеллектом, в определенных контекстах воспринимаются как более чуткие и заботливые, чем реакции человека, что указывает на значительный прогресс в понимании естественного языка и возможностях генерации информации [9]. Эта разработка имеет серьезные последствия для приложений в сфере обслуживания клиентов, здравоохранения и образования.

В литературе показано, что генеративный ИИ добился особых успехов в творческом приложении, а системы теперь способны создавать высококачественный контент в различных форматах, включая текст, изображения, аудио и видео. Эти возможности позволили создать новые формы сотрудничества человека и искусственного интеллекта и начали трансформировать различные отрасли — от развлечений до маркетинга.

2.2. Возможности машинного обучения и логического мышления

Недавние достижения в области машинного обучения были сосредоточены, в частности, на улучшении возможностей логического мышления и эффективности вычислений [8]. Разработка более быстрых каскадных методов с помощью спекулятивного декодирования позволила устраниТЬ одно из основных ограничений больших языковых моделей: их вычислительные затраты и задержку ответа. Эти технические усовершенствования сделали сложные системы искусственного интеллекта более практическими для применения в реальных условиях.

Математическое мышление представляет собой особенно заметную область улучшений, поскольку системы искусственного интеллекта демонстрируют расширенные возможности в решении сложных математических задач, проведении формальных доказательств и работе с абстрактными математическими концепциями [8]. Эти достижения имеют значение для научных исследований, инженерных приложений и образовательных инструментов.

2.3. Мультимодальные и автономные системы

К 2025 году интеграция множества модальностей в системах искусственного интеллекта значительно продвинулась вперед [2]. Текущие исследования демонстрируют успешное объединение текста, изображений, аудио и других типов данных в согласованных системах искусственного интеллекта, способных понимать и генерировать контент в различных медиа-форматах. Эта мультимодальная возможность позволила найти новые применения в творческих отраслях, научных исследованиях и взаимодействии человека и компьютера.

Автономные агенты с искусственным интеллектом представляют собой еще одну важную область разработки, где системы демонстрируют возросшую независимость в выполнении задач и принятии решений [7]. Теперь эти агенты могут работать с большей автономией как в профессиональном, так и в личном плане, упрощая сложные рабочие процессы и предоставляя пользователям более квалифицированную помощь.

Развитие автономных систем особенно заметно в специализированных областях, таких как научные исследования, где агенты искусственного интеллекта теперь могут автоматизировать важнейшие этапы исследовательского процесса, включая обзор литературы, генерацию гипотез и экспериментальный дизайн. Эта возможность потенциально может ускорить научные открытия в различных дисциплинах.

2.4. Разработка аппаратного обеспечения и инфраструктуры

Развитие возможностей ИИ тесно связано с инновациями в аппаратном обеспечении и вычислительной инфраструктуре. Все большее значение приобретает специализированный кремний, разработанный специально для рабочих нагрузок ИИ. Крупные технологические компании вкладывают значительные средства в разработку фирменных чипов, оптимизированных для вычислений с использованием ИИ.

Исследования показывают, что разработка специализированного аппаратного обеспечения стала решающим фактором в повышении производительности и эффективности систем искусственного интеллекта, а использование специального кремния позволяет как повысить производительность, так и снизить энергопотребление [6].

Возможности передовых вычислений также значительно расширились, позволяя обрабатывать данные с помощью искусственного интеллекта на локальных устройствах, а не на основе облачных вычислений. Эта разработка имеет важные последствия для обеспечения конфиденциальности, снижения задержек и широкого внедрения приложений искусственного интеллекта.

3. Модели внедрения и отраслевая реализация

3.1. Статистика внедрения на предприятиях

Текущие данные свидетельствуют о беспрецедентном уровне внедрения ИИ в организациях по всему миру [3]. Исследования показывают, что 78 % организаций в настоящее время используют ИИ по крайней мере в одной бизнес-функции, что представляет собой значительный рост по сравнению с 72 % в начале 2024 года и 55 % в 2023 году [3]. Такой быстрый темп внедрения говорит о том, что искусственный интеллект превратился из экспериментальной технологии в незаменимую для многих организаций.

Наиболее распространенные области внедрения включают функции в области информационных технологий, маркетинга и продаж, при этом организации сообщают о значительной производительности и повышение эффективности [4]. Данные опроса показывают, что 58 % организаций добились экспоненциального повышения производительности или результативности благодаря внедрению искусственного интеллекта, в первую очередь за счет генеративного подхода [3].

Отраслевой анализ показывает, что первые пользователи ИИ в настоящее время расширяют свои возможности по внедрению дополнительных бизнес-функций, в то время как организации, которые ранее колебались, начинают свои первые инициативы в области ИИ [5]. Эта закономерность предполагает, что внедрение ИИ будет продолжать ускоряться по мере того, как организации будут наблюдать за успешными внедрениями конкурентов и партнеров.

3.2. Отраслевые приложения

Различные отрасли промышленности внедряют технологии искусственного интеллекта с разной скоростью и в разных областях деятельности [4]. Технологический сектор продолжает лидировать в области внедрения искусственного интеллекта, но значительное внедрение произошло в секторах здравоохранения, финансовых, производственных и образовательных. В каждом секторе разработаны специализированные приложения, которые решают специфические отраслевые задачи и открывают новые возможности.

В здравоохранении приложения ИИ вышли за рамки диагностической визуализации и включают в себя поиск лекарств, индивидуальное планирование лечения и автоматизацию

администрирования. Финансовый службы внедрили ИИ для обнаружения мошенничества, алгоритмической торговли и обслуживания клиентов, при этом соблюдение нормативных требований остается ключевым фактором.

Производство использует ИИ для прогнозного технического обслуживания, контроля качества и оптимизации цепочки поставок, уделяя особое внимание интеграции ИИ с существующими промышленными системами. Образовательные учреждения внедрили ИИ для персонализированного обучения, автоматизации администрирования и помощи в проведении исследований.

Научные исследования представляют собой особенно перспективную область применения ИИ-агентов теперь они способны автоматизировать важнейшие этапы исследовательского процесса. Эта разработка потенциально может способствовать ускорению научных открытий и развитию человеческих знаний в различных дисциплинах.

3.3. Модели принятия и использования потребителями

Внедрение технологий искусственного интеллекта потребителями демонстрирует интересные закономерности: исследования показывают, что 55 % россиян регулярно используют искусственный интеллект, в то время как 44 % считают, что они не используют его регулярно [6]. Это несоответствие свидетельствует о том, что ИИ интегрирован во многие потребительские приложения способами, которые пользователи могут явно не распознавать как функциональные возможности ИИ.

Модели использования ИИ на рабочих местах показывают, что инструменты ИИ все чаще интегрируются в повседневные рабочие процессы, а сотрудники используют ИИ для различных задач, включая создание контента, анализ и решение проблем [7]. Однако показатели внедрения значительно различаются в зависимости от демографических групп и географических регионов, при этом молодые пользователи и представители профессий, связанных с технологиями, демонстрируют более высокие показатели внедрения.

Потребительские приложения вышли за рамки чат-ботов и виртуальных помощников и включают функции на базе ИИ в смартфонах, платформах социальных сетей и развлекательных сервисах. Интеграция ИИ в существующие приложения сделала функциональность ИИ более доступной для обычных потребителей, которые могут не прибегать к специальным инструментам ИИ.

4. Производительность и конкурентный ландшафт

4.1. Глобальная конкуренция с использованием искусственного интеллекта

В 2025 году конкуренция в сфере ИИ значительно усилилась, что привело к заметным изменениям в относительной эффективности различных национальных программ развития ИИ [1]. Исследования показывают, что разрыв в производительности между ведущими моделями ИИ из разных стран значительно сократился. В начале 2024 года лучшая американская модель искусственного интеллекта превосходила лучшую китайскую модель на 9,26 %, но к февралю 2025 года этот разрыв сократился всего до 1,70 % [2].

Такая конвергенция возможностей искусственного интеллекта в различных центрах разработки позволяет предположить, что более конкурентный глобальный ландшафт указывает на то, что развитие искусственного интеллекта становится все более распространенным на международном уровне [1]. Последствия этой тенденции выходят за рамки технических возможностей и охватывают экономические, стратегические и geopolитические аспекты.

Развитие ИИ в Европе также набирает обороты благодаря увеличению инвестиций в исследования и разработки ИИ, особенно в областях, где особое внимание уделяется этике ИИ и соблюдению нормативных требований. Этот региональный подход отражает различные приоритеты и ценности в отношении разработки и внедрения ИИ.

4.2. Показатели эффективности модели

Недавние сравнительные исследования показывают постоянное улучшение производительности моделей ИИ по нескольким критериям оценки, включая способность рассуждать, решать математические задачи и кодировать [8]. Эти улучшения демонстрируют, что системы ИИ становятся более способными решать более широкий спектр интеллектуальных задач.

Повышение производительности было особенно заметно в областях, требующих сложного мышления и многоэтапного решения проблем, что говорит о том, что системы искусственного интеллекта развиваются сложные когнитивные способности, приближающиеся к показателям человеческого уровня во многих областях. Однако в различных типах задач и контекстах сохраняются несоответствия в производительности.

Стандартизованный сопоставительный анализ становится все более важным по мере развития возможностей ИИ, поскольку разрабатываются новые системы оценки для оценки возможностей, которые ранее были недоступны системе ИИ [10]. Эти критерии имеют решающее значение для отслеживания прогресса и определения областей, в которых необходимы дальнейшие исследования.

4.3. Коммерческая конкуренция и динамика рынка

Рынок коммерческого ИИ становится все более конкурентным, и признанные технологические гиганты сталкиваются с проблемами со стороны специализированных ИИ-компаний и стартапов. Это соревнование ускорило внедрение инноваций, а также вызвало обеспокоенность по поводу концентрации рынка и конкурентных практик.

Инвестиции в компании, занимающиеся ИИ, достигли рекордных уровней, причем венчурный капитал и корпоративные инвестиции поступают в компании, разрабатывающие как базовые технологии ИИ, так и специализированные приложения. Такая структура инвестиций отражает уверенность в коммерческом потенциале ИИ, а также указывает на капиталоемкий характер передовых разработок в области искусственного интеллекта.

На динамику рынка повлияли высокие вычислительные затраты, связанные с обучением и внедрением передовых систем искусственного интеллекта, что создает преимущества для организаций, обладающих значительными ресурсами, но потенциально ограничивает доступ для небольших организаций.

5. Влияние на общество и этические соображения

5.1. Трансформация рабочего места

Интеграция искусственного интеллекта в рабочую среду привела к значительным изменениям в том, как работают организации и как сотрудники взаимодействуют с технологиями [4]. Исследования показывают, что искусственный интеллект превращается из дополнительного инструмента в неотъемлемый компонент рабочих процессов, а агенты, работающие на базе искусственного интеллекта, обеспечивают большую автономию и помочь как в профессиональном, так и в личном контексте [7].

Исследования показывают, что успешное внедрение ИИ требует организационной реструктуризации и новых подходов к сотрудничеству человека и ИИ [3]. Организации, достигшие лучших результатов от внедрения ИИ, как правило, вкладывают во всестороннее управление изменениями и программы обучения сотрудников.

Влияние на структуру занятости остается предметом постоянных исследований, причем исследования указывают как на смену рабочих мест на определенных должностях, так и на создание рабочих мест на других. Чистый эффект, по-видимому, существенно различается в зависимости от отрасли и уровня квалификации, что подчеркивает важность программ повышения квалификации и переподготовки кадров.

5.2. Вопросы доверия и принятия

Исследования, посвященные доверию и принятию ИИ, выявляют сложные закономерности в общественном отношении к технологиям искусственного интеллекта. Несмотря на то, что показатели внедрения продолжают расти, опасения по поводу надежности ИИ, предвзятости и потенциальных негативных последствий остаются важными факторами, влияющими на общественное восприятие и принятие систем ИИ.

В литературе указывается, что доверие к системам искусственного интеллекта значительно варьируется в зависимости от различных приложений и демографических групп, при этом такие факторы, как прозрачность, объяснимость и предполагаемый контроль над системами искусственного интеллекта, играют решающую роль в принятии и удовлетворенности пользователей. Медицинские и финансовые приложения, как правило, требуют более высокого уровня доверия и прозрачности, чем приложения для развлечения или повышения производительности.

Культурные различия и различия поколений существенно влияют на уровень доверия: молодые пользователи, как правило, более комфортно относятся к системам искусственного интеллекта, в то время как пожилые пользователи и те, кто находится в определенных культурных контекстах, проявляют больше осторожности. Эти различия имеют важное значение для стратегий внедрения искусственного интеллекта и разработки пользовательского интерфейса.

5.3. Последствия для конфиденциальности и безопасности

Системы искусственного интеллекта вызывают серьезные опасения по поводу конфиденциальности и безопасности, особенно по мере того, как они становятся более способными обрабатывать и генерировать личную информацию. Исследования показывают, что многие пользователи не в полной мере осведомлены о том, как их данные используются в системах искусственного интеллекта или какие меры защиты конфиденциальности применяются.

Уязвимости безопасности в системах искусственного интеллекта вызывают все большую озабоченность, поскольку исследователи выявляют различные векторы атак, включая враждебный ввод данных, заражение данных и атаки на извлечение моделей. Эти проблемы безопасности требуют новых подходов к проектированию и развертыванию систем.

Концентрация возможностей ИИ среди небольшого числа крупных технологических компаний вызвала дополнительные опасения по поводу контроля над данными, влияния на рынке и возможности злоупотреблений. Эти опасения повлияли на обсуждения нормативных актов и практику корпоративного управления в индустрии ИИ.

6. Новые тенденции и направления на будущее

6.1. Агентные системы искусственного интеллекта

Одной из наиболее значимых тенденций, выявленных в литературе к 2025 году, является разработка агентных систем искусственного интеллекта, способных к независимым действиям и принятию решений [7]. Эти системы представляют собой переход от реактивных инструментов искусственного интеллекта к проактивным агентам искусственного интеллекта, которые могут инициировать действия, принимать решения и адаптироваться к изменяющимся обстоятельствам без постоянного контроля со стороны человека.

Исследования показывают, что агентные системы искусственного интеллекта имеют потенциальное применение во многих областях - от персональной помощи до автоматизации сложных бизнес-процессов. Однако, разработка этих систем также поднимает важные вопросы об подотчетности, контроле и надлежащем балансе между автономией ИИ и контролем со стороны человека.

Технические проблемы разработки надежных агентных систем включают в себя обеспечение надлежащего принятия решений, поддержание соответствия человеческим ценностям и разработку соответствующих механизмов для вмешательства человека в случае необходимости. Эти проблемы представляют собой активные области исследований, которые имеют значительные последствия для будущего развития ИИ.

6.2. Интеграция с новыми технологиями

Системы искусственного интеллекта все чаще интегрируются с другими новыми технологиями, включая квантовые вычисления, блокчейн и устройства Интернета вещей (IoT). Эти интеграции создают новые возможности для приложений искусственного интеллекта, а также ставят новые технические и этические задачи.

Гибридные системы с квантовым ИИ представляют собой особенно многообещающую область разработки, имеющую потенциальное применение в задачах оптимизации, криптографии и научном моделировании. Хотя практические системы с квантовым ИИ остаются в основном экспериментальными, прогресс в исследованиях предполагает значительный потенциал в будущем.

Интеграция ИИ с устройствами Интернета вещей расширила возможности применения систем ИИ в физических средах, позволив создавать новые приложения в «умных городах», автономных транспортных средствах и промышленной автоматизации. Такая интеграция требует новых подходов к проектированию систем, обеспечению безопасности и конфиденциальности.

7. Проблемы и ограничения

7.1. Технические ограничения

Несмотря на значительный прогресс, современные системы искусственного интеллекта по-прежнему сталкиваются с серьезными техническими ограничениями [8]. Исследования выявляют текущие проблемы в таких областях, как последовательность рассуждений, точность фактов и способность к обобщениям в различных областях и контекстах.

В литературе указывается, что, хотя системы искусственного интеллекта достигли впечатляющей производительности во многих конкретных задачах, обеспечение надежных возможностей ИИ общего назначения остается серьезной проблемой, требующей продолжения

исследований и разработок. К числу особых ограничений относятся трудности с логическим обоснованием, неспособность по-настоящему понять контекст так, как это делают люди, и склонность генерировать правдоподобную, но неверную информацию.

Вычислительная эффективность остается существенным ограничением, поскольку наиболее эффективные системы искусственного интеллекта требуют огромных вычислительных ресурсов, что ограничивает их доступность и развертывание. Исследования в области более эффективных архитектур и методов обучения продолжаются, но пока не решили эту фундаментальную проблему.

Заключение

Анализ литературы по ИИ в 2025 году показывает, что эта область характеризуется быстрым развитием, широким внедрением и растущей сложностью как в технических возможностях, так и в практическом применении. Переход от экспериментальных систем искусственного интеллекта к готовым к производству инструментам, интегрированным в повседневные рабочие процессы, представляет собой важную веху в развитии искусственного интеллекта.

Ключевые выводы включают достижение существенного повышения производительности за счет внедрения искусственного интеллекта: 58 % организаций сообщили о значительном улучшении. Сокращение международного конкурентного разрыва в возможностях ИИ с 9,26 % до 1,70 % указывает на более глобальное распределение и конкурентную среду ИИ. Появление все более автономных систем ИИ предполагает фундаментальный сдвиг в сторону более независимых искусственных агентов.

Широкое внедрение ИИ, когда 78 % организаций в настоящее время используют ИИ по крайней мере в одном бизнес-процессе, демонстрирует, что ИИ превратился из новой технологии в важнейший бизнес-инструмент. Однако такое быстрое внедрение сопровождается важными проблемами, связанными с доверием, этикой, сложностью внедрения и влиянием на общество, которые требуют постоянного внимания и исследований.

Быстрые темпы изменений в технологии искусственного интеллекта предполагают, что постоянный мониторинг и анализ разработок в области искусственного интеллекта будет иметь важное значение для понимания траектории развития отрасли и ее последствий для общества. По мере того как системы искусственного интеллекта становятся все более функциональными и повсеместными, важность продуманных, основанных на фактических данных подходов к управлению и внедрению искусственного интеллекта будет только возрастать.

Совмещение множества возможностей ИИ в более общие и автономные системы представляет собой как беспрецедентную возможность, так и значительную ответственность для сообщества исследователей и разработчиков ИИ. От того, насколько хорошо управляются эти возможности и обязанности, скорее всего, зависит, реализует ли ИИ свой потенциал на благо человечества, избегая при этом потенциальных негативных последствий.

Литература

1. Stanford HAI. The AI Index Report 2025 [Электронный ресурс]. – Stanford University Human-Centered AI Institute, 2025. – URL: <https://aiindex.stanford.edu> (дата обращения: 09.11.2025).
2. IEEE Spectrum. The State of AI 2025: 12 Eye-Opening Graphs [Электронный ресурс]. – Май 2025. – URL: <https://spectrum.ieee.org/ai-index-2025> (дата обращения: 09.11.2025).
3. McKinsey & Company. The state of AI: How organizations are rewiring to capture value [Электронный ресурс]. – Март 2025. – URL: <https://www.mckinsey.com> (дата обращения: 09.11.2025).
4. MIT Sloan Management Review. Five Trends in AI and Data Science for 2025 [Электронный ресурс]. – 2025. – URL: <https://sloanreview.mit.edu> (дата обращения: 09.11.2025).

5. Morgan Stanley. 5 AI Trends Shaping Innovation and ROI in 2025 [Электронный ресурс]. – 2025. – URL: <https://www.morganstanley.com> (дата обращения: 09.11.2025).
6. National University. 131 AI Statistics and Trends for 2025 [Электронный ресурс]. – Январь 2025. – URL: <https://www.nu.edu> (дата обращения: 09.11.2025).
7. Microsoft News. 6 AI trends you'll see more of in 2025 [Электронный ресурс]. – Май 2025. – URL: <https://news.microsoft.com> (дата обращения: 09.11.2025).
8. MachineLearningMastery.com. 5 Breakthrough Machine Learning Research Papers Already in 2025 [Электронный ресурс]. – Май 2025. – URL: <https://machinelearningmastery.com> (дата обращения: 09.11.2025).
9. Generative Artificial Intelligence: Evolving Technology, Growing Societal Impact, and Opportunities for Information Systems Research // Information Systems Frontiers [Электронный ресурс]. – Февраль 2025. – URL: <https://link.springer.com/journal/10796> (дата обращения: 09.11.2025).
10. Generative artificial intelligence: a systematic review and applications // Multimedia Tools and Applications [Электронный ресурс]. – Август 2024. – URL: <https://link.springer.com/journal/11042> (дата обращения: 09.11.2025).

РАЗРАБОТКА ИНСТРУМЕНТОВ АНАЛИЗА ПРЕДПОЧТЕНИЙ ЗРИТЕЛЕЙ НА ОСНОВЕ ПРОГНОЗИРОВАНИЯ РЕЙТИНГА ФИЛЬМА ПО ЕГО ОПИСАНИЮ

Т. К. Захарова

Воронежский государственный университет

Аннотация. В работе поставлена задача прогнозирования рейтинга фильма по его описанию. Этот подход позволяет оценить успех у зрителей еще до появления первых отзывов, что способствует автоматизации анализа предпочтений аудитории. Выделяется актуальность данной темы и потенциальные области практического применения, обсуждаются различные методики обработки текста и машинного обучения, разбираются подходы к решению данной задачи, метрики оценки качества модели, а также конкретный план по достижению результата и дальнейшие возможности для расширения.

Ключевые слова: прогноз рейтингов фильмов, машинное обучение, обработка естественного языка, анализ текстов, кинопроизводство.

Введение

Сейчас киноиндустрия показывает стремительный рост производства контента: от полнометражных фильмов до телесериалов. Онлайн-сервисы и платформы борются за внимание зрителей, используя такие способы, как персонализированный контент, подборку по темам и адаптирующиеся интерфейсы. В это время, когда сфера киноиндустрии переполнена, зрителю требуется инструментарий для предварительной оценки того, насколько данный фильм сможет оправдать его ожидания.

Обычно рейтинги формируются уже после выпуска фильма, когда зрители его оценили. Но еще на стадии создания проекта, когда готовятся сценарии, презентации, аннотации, продюсерам необходим сервис, который сможет подсказать будет ли фильм успешен у целевой аудитории. В этой проблеме могла бы помочь модель предсказания рейтинга фильма на основе короткого описания, она способствует оптимизации решений, связанных с финансами, продюсированием и рекламой.

Текстовое описание к фильму содержит важнейшие признаки, связанные с семантикой, а именно: жанровая принадлежность, эмоциональная окраска, структура повествования, что непосредственно связано с потенциальным зрителем. Современные методы обработки естественного языка (NLP), которые отлично сочетаются с методами машинного обучения, помогают извлекать указанные характеристики, что делает задачу предсказания рейтинга не теоретическим аспектом, а практически осуществимой задачей. Данная работа использует существующие методики, определяет их ограничения и преимущества, выделяет факторы, влияющие на точность предсказания.

Актуальность

Рассмотрим данный вопрос в рамках экономической сферы. Решения о выделении средств для кинопроизводства принимаются еще задолго до выхода фильма в прокат. Наличие спрогнозированного рейтинга сможет помочь корректно распределить ресурсы, оптимизировать расходы, а также снизить риски нецелевого использования финансов. Для начинающих студий такой механизм особенно важен, он поможет выделить перспективы проекта, не тратя средства на рекламные кампании.

Со стороны развития технологий анализа текста за последнее время произошел стремительный прогресс. Большое количество компаний стало переходить от простых к частотным методам, использующим контекстно-зависимые эмбеддинги и архитектуры на основе трансформеров. Данные методы позволяют учитывать многозначность слов, связи структур и эмоциональную окраску текста. Эти параметры ранее трудно поддавались интерпретации.

Со стороны социальной сферы с увеличением количества систем, которые выделяют персонализированные рекомендации, появляется потребность в учете интересов потребителей. Система прогноза поможет снизить количество неинтересного и неактуального контента для конкретного зрителя, что позволяет расположить к себе потенциальных клиентов и сохранить имеющихся.

Также важным моментом является научная значимость. Анализ предпочтений зрителей и содержания текстов позволяет выделить жанровые, культурные и локальные закономерности, способствуя развитию лингвистики, этнографии и прикладного машинного обучения.

1. Постановка задачи

Формулируем задачу: мы имеем набор фильмов, описанию к фильму, а также дополнительно набор признаков(год, страна, режиссеры, жанры, список главных актеров, продолжительность фильма). Нам необходимо по имеющимся данным предсказать рейтинг фильма.

Важнейшие характеристики задачи:

- Тип задачи — регрессия (предсказывается числовое значение). В отдельных ситуациях может быть преобразована в классификацию (например, «низкий/средний/высокий рейтинг»), что полезно для практических сценариев.
- Входные данные — тексты различной длины; требуется их унификация и представление в векторном виде.
- Дополнительные признаки — помимо текста, возможно использование визуальных и табличных данных: трейлеры, изображения, бюджет и др. В данной статье фокус сделан на текстовом описании, но обсуждается возможность интеграции мультимодальных источников.

Этапы решения задачи включают:

1. Сбор и объединение данных из разных источников.
2. Предварительная обработка и очистка текстов.
3. Извлечение признаков: от классических статистических методов до контекстуальных векторных представлений.
4. Обучение регрессионной модели.
5. Оценку качества и анализ погрешностей.
6. Внедрение и последующий мониторинг модели.

2. Подходы к решению задачи

2.1. Классические методы (TF-IDF и ML)

Один из первых подходов основывался на представлении текста в виде набора частот слов (bag-of-words) или TF-IDF матрицы. На этих данных обучались линейные модели, SVM, случайные леса и алгоритмы градиентного бустинга. Такие методы просты в применении, обладают низкой вычислительной стоимостью и приемлемой интерпретируемостью признаков, однако полностью игнорируют порядок слов и не учитывают контекст.

Примером может служить использование TF-IDF совместно с градиентным бустингом для выявления наиболее значимых терминов в описании («трогательный», «напряжённый») и их влияния на итоговую оценку.

2.2. Эмбеддинги слов и усреднение

Модели Word2Vec, FastText или GloVe позволяют получать плотные векторные представления, отражающие смысловые связи между словами. Распространённая стратегия — вычисление среднего вектора по всем словам описания. Такой подход лучше передаёт семантику текста и превосходит по качеству традиционные мешки слов, но всё ещё не захватывает синтаксические структуры.

Эмбеддинги в сочетании с деревьями решений или простыми нейронными сетями обеспечивают заметный прирост качества без значительных вычислительных затрат.

2.3. Контекстные модели и трансформеры

Современные архитектуры (например, BERT, RoBERTa, DistilBERT) формируют контекстно-зависимые представления, учитывая окружение каждого слова. Для прогнозирования рейтинга применяются два основных подхода:

- использование эмбеддингов описания в качестве входа для дополнительной регрессионной модели;
- дообучение (fine-tuning) трансформера на задаче прогнозирования рейтинга с учетом специфики кинематографических текстов.

Трансформерные модели особенно эффективны при наличии большого обучающего корпуса и необходимости учитывать эмоциональные и повествовательные нюансы описаний.

2.4. Гибридные и мультимодальные решения

Для повышения точности предсказаний текстовые признаки объединяют с метаданными (жанр, актёрский состав, год релиза) и визуальной информацией (постеры, трейлеры). Мультимодальные модели интегрируют данные разных типов и формируют единое представление, что зачастую улучшает точность. Подобные архитектуры включают отдельные ветви обработки для каждого источника и общий блок, осуществляющий регрессию.

2.5. Интерпретируемость моделей

Практическая ценность прогноза возрастает при наличие объяснимости. Методы SHAP и LIME позволяют оценить вклад отдельных признаков и слов в результат, что важно для принятия решений продюсерами и аналитиками.

3. Источники данных и предварительная обработка.

3.1. Источники

Для формирования обучающего набора могут использоваться:

- IMDb — подробные описания, пользовательские оценки и рейтинги;
- TMDB — большая база метаданных и мультимедийного контента;
- Кинопоиск — значимый источник русскоязычных данных;
- тематические фестивальные архивы, рецензии критиков и данные социальных сетей, отражающие ранний интерес аудитории.

3.2. Предобработка текста

Ключевые шаги включают:

- нормализацию (приведение регистра, удаление лишних символов);
- очистку HTML-элементов;
- токенизацию с учётом особенностей языка;
- лемматизацию или стемминг;
- удаление стоп-слов;
- идентификацию имен собственных (актёров, режиссеров, локаций), которые могут быть значимыми предикторами.

3.3. Инжениринг признаков

Помимо векторизации полезно формировать дополнительные характеристики:

- длина описания (в символах или словах);
- доля прилагательных или наречий;
- наличие тематических ключевых слов (например, «биография», «триллер»);
- бинарные индикаторы жанров;
- статистические данные по похожим фильмам.

4. Оценка качества модели

Корректное тестирование играет решающую роль.

4.1. Метрики

- MAE — интерпретируемая величина, устойчивость к выбросам;
- MSE / RMSE — чувствительность к крупным отклонениям;
- R^2 — показатель доли объяснённой дисперсии;
- Метрики ранжирования (Spearman, Kendall) — актуальны для рекомендательных систем.

4.2. Анализ ошибок

Рекомендуется изучать примеры с крупными отклонениями, что помогает выявлять особенности моделей и недостатки обучающих данных. Также полезно анализировать редкие жанры и узкие тематические сегменты, где модели обычно работают хуже.

4.3. Базовые и продвинутые решения

Простые модели на TF-IDF и линейной регрессии служат стартовой точкой, после чего следует сравнение с методами на эмбеддингах и трансформерах. Финальный выбор модели определяется балансом между точностью и вычислительными требованиями.

5. Практическая значимость и перспективы развития

5.1. Применение

- рекомендательные системы: ранжирование новых релизов и решение проблемы cold-start;
- аналитика для продюсеров: ранняя оценка сценариев и аннотаций;

- маркетинг: адаптация описаний под целевую аудиторию;
- фестивальные комитеты: автоматическая фильтрация заявок по вероятному восприятию.

5.2. Ограничения и риски

- субъективность оценок: разные группы пользователей по-разному воспринимают один и тот же проект;
- внешние факторы: маркетинг, актёрский состав, стратегия релиза существенно влияют на рейтинг, но редко отражены в аннотациях;
- смещение данных: публичные датасеты могут содержать нерепрезентативные примеры;
- этические вопросы: автоматический прогноз не должен заменять экспертное мнение.

6. Практическая методика реализации

Последовательность действий при создании рабочей системы включает:

1. Сбор данных (например, с Кинопоиска), включая метаданные и показатели рейтингов.
2. Исследовательский анализ: изучение распределения оценок по жанрам, статистика длины описаний, частотный анализ лексики.
3. Предобработка: очистка, токенизация, лемматизация, выделение именованных сущностей.
4. Базовый baseline: TF-IDF + линейная регрессия или градиентный бустинг.
5. Улучшение: применение Word2Vec/GloVe и комбинированных признаков.
6. Современный вариант: дообучение модели типа BERT с использованием методов объяснимости (SHAP).
7. Валидация: перекрёстное тестирование и выделенный тестовый набор.
8. Эксплуатация: регулярное обновление модели и учёт пользовательской обратной связи.

Заключение

Прогнозирование рейтинга фильма по его аннотации — это задача, которая важна и с практической точки зрения, и с теоретической. По сути, из текстового описания можно вытащить много полезной информации и превратить ее в набор признаков, на основе которых модель способна достаточно точно оценивать будущий рейтинг. Современные методы обработки естественного языка, особенно модели на архитектуре трансформеров, сильно улучшили качество таких предсказаний и расширили возможности анализа.

При этом остаются и проблемы: на результат влияют внешние факторы, в данных могут быть смещения, а сами модели часто сложно интерпретировать. Поэтому для реального применения имеет смысл дополнять текстовые признаки другими источниками данных и мультимодальной информацией, чтобы сделать прогнозы более устойчивыми и надежными. На основе уже существующей методологической базы и практических рекомендаций можно разрабатывать полноценные решения для рекомендательных сервисов, инструментов для продюсеров и систем планирования продвижения контента. Перспективные направления развития связаны с углублением мультимодальных подходов, адаптацией моделей под разные группы пользователей и тщательным анализом языковых особенностей, которые влияют на долгосрочную популярность фильмов.

Литература

1. *Bishop C. M. Pattern Recognition and Machine Learning* : монография / C. M. Bishop. – New York : Springer, 2006. – 738 p.

2. *Manning C. D. Foundations of Statistical Natural Language Processing* : монография / C. D. Manning, H. Schütze. – Cambridge, MA : MIT Press, 1999. – 680 p.
3. *Веденеева О. Н. Методы интеллектуального анализа данных* : учебное пособие / О. Н. Веденеева, Т. В. Веденеева. – Москва : Физматлит, 2010. – 304 с.
4. *Кузнецов С. Д. Машинное обучение и анализ данных* : учебное пособие / С. Д. Кузнецов. – Москва : Бином. Лаборатория знаний, 2012. – 400 с.
5. *Борисов В. Н. Искусственный интеллект: методы и средства* : учебное пособие / В. Н. Борисов, И. В. Ковалева, А. В. Тюрин. – Москва : Горячая линия – Телеком, 2019. – 368 с.
6. *Коларов В. В. Технологии обработки и анализа текстов* : учебное пособие / В. В. Коларов. – Санкт-Петербург : БХВ-Петербург, 2016. – 496 с.

АРХИТЕКТУРА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ СЕМАНТИЧЕСКОГО ПОИСКА НАУЧНЫХ ПУБЛИКАЦИЙ

К. А. Зверев, И. Е. Воронина

Воронежский государственный университет

Аннотация. Рассматривается разработка системы семантического поиска и рекомендаций научных статей на основе векторных представлений текста. Представлена архитектура, алгоритмы ранжирования на основе генерации эмбеддингов текста научных работ, получаемые при помощи моделей искусственного интеллекта в облачных сервисах, а также методы обработки пользовательских предпочтений. Предлагаемое решение обеспечивает персонализированный поиск научных публикаций и может применяться в образовательных и исследовательских платформах.

Ключевые слова: персонализированные рекомендации, интеллектуальный поиск, тематическая кластеризация, ранжирование результатов, обработка текстов, цифровые библиотеки, векторизация текста, искусственный интеллект.

Введение

В современном научном пространстве наблюдается стремительный рост объёма публикаций. Каждый исследователь сталкивается с проблемой информационной перегрузки и трудностью быстрого нахождения релевантных источников. Имеющиеся инструменты являются не всегда удобными и зачастую не обеспечивают достаточной удовлетворённости пользователя, поскольку поиск по ключевым словам и традиционным базам данных не позволяет точно подбирать материалы с учётом индивидуальных интересов.

Современные исследователи сталкиваются с необходимостью обработки огромного количества научных публикаций, что существенно усложняет поиск релевантных материалов для проведения качественного анализа и подготовки собственных исследований. Существующие инструменты поиска, ориентированные на ключевые слова и стандартные базы данных, часто не обеспечивают достаточной точности и персонализации, что приводит к потере времени и снижению эффективности работы с научной литературой.

В связи с этим возникает потребность в разработке интеллектуальной системы, которая могла бы анализировать содержание статей и предоставлять пользователю рекомендации, учитывающие семантическую близость материалов и индивидуальные интересы [1]. Основной задачей системы является обеспечение эффективного поиска и упорядочивания научных публикаций, включая обработку и нормализацию текстов, преобразование их в векторные представления, а также построение механизма ранжирования и кластеризации.

Ключевым требованием к проекту является высокая точность и релевантность выдачи при работе с различными объемами данных, а также возможность гибкой интеграции с существующей инфраструктурой без необходимости использования специализированного оборудования.

Задача создания системы, способной обеспечивать точный и удобный подбор научных публикаций в условиях растущего объёма информации, остаётся актуальной. В частности, применение технологий искусственного интеллекта и современных методов анализа текста позволяет увеличить релевантность результатов и качество поиска для конечного пользователя.

1. Предлагаемые технологии и алгоритмы

Для разработки предлагается использовать модульную архитектуру для обеспечения последующего масштабирования и возможности независимой разработки каждого компонента.

Для решения задачи необходимо реализовать следующие возможности: сбор и хранение данных о научных статьях, обработку текста, поиск релевантных статей по запросу пользователя, ранжирование и рекомендации.

Для достижения поставленных целей необходимо применить такие инструменты, как генерация эмбеддингов загруженных публикаций, алгоритмы векторного поиска [2] текстов со схожей тематикой на основе вычисления косинусного расстояния, а также API и пользовательский клиент, обеспечивающие взаимодействие пользователя с системой.

Реализация указанных алгоритмов позволит выстроить полный цикл работы пользователя с научными публикациями — начиная от загрузки документа и его обработки, и заканчивая предоставлением наиболее релевантных и персонализированных результатов.

1.1. Сбор и хранение данных

Пользователи могут загружать научные статьи в различных форматах, включая PDF и DOCX. После загрузки файлы должны сохраняться объектном хранилище, совместимом с протоколом S3, например, MinIO. Это обеспечивает надежное хранение документов и упрощает управление большим количеством данных.

Метаданные статей, такие как название, авторы, дата публикации, ключевые слова и аннотация, сохраняются в реляционной базе данных PostgreSQL. Структурированное хранение метаданных позволяет быстро выполнять поиск по атрибутам документа, а также облегчает интеграцию с внешними сервисами и аналитическими инструментами.

1.2. Обработка и нормализация текста

Для извлечения текста из файлов предлагается использовать библиотеку Apache Tika, которая поддерживает широкий спектр форматов файлов и позволяет получать текст в структурированном виде. Она позволит выполнить очистку текста от лишних символов и форматирования.

Дополнительно производится выделение структурных элементов статьи — заголовков, подзаголовков, списков литературы и ключевых фрагментов. После подобной обработки будет поучено унифицированное представление текста для корректной генерации эмбеддингов. Обработка текста также может включать сегментацию больших документов на логические блоки, что повышает точность последующего анализа и поиска по смыслу.

1.3. Генерация эмбеддингов

Также необходим модуль для преобразования текстовых данные в числовые векторные представления, которые отражают семантическое содержание статьи. Каждая статья должна быть закодирована в эмбеддинг, что позволяет проводить вычисления близости между текстами и формировать рекомендации. Векторизация текста обеспечивает возможность выявления смысловых связей между статьями, что невозможно при поиске по ключевым словам.

Использование облачного сервиса упрощает интеграцию ИИ-моделей в систему и позволяет обрабатывать большие объёмы данных без необходимости развёртывания собственных моделей на сервере.

1.4. Векторный поиск

При выполнении запроса пользователя текст запроса также преобразуется в эмбеддинг, после чего система выполняет поиск с использованием метрики косинусного сходства, что

позволяет находить статьи, наиболее близкие по смыслу к запросу. Из этого следует, что эмбеддинги статей необходимо хранить в базе данных, которая поддерживает использование специализированных индексов для эффективного поиска по высокоразмерным векторным представлениям.

Использование PostgreSQL с векторными индексами позволяет объединять хранение структурированных метаданных и эмбеддингов в одной системе, упрощая управление данными и интеграцию с другими компонентами. Такой подход обеспечивает позволяет выполнять семантический поиск без необходимости разворачивать отдельную векторную базу.

1.5. Ранжирование и кластеризация

После выполнения поиска система применяет алгоритмы ранжирования, которые упорядочивают найденные статьи по степени соответствия запросу. Основным критерием является косинусное сходство между эмбеддингом запроса и эмбеддингами статей. Дополнительно система может учитывать тематические кластеры, предпочтения пользователя и историю его взаимодействия с материалами.

Модуль рекомендаций [3] формирует персонализированный список статей, который учитывает интересы пользователя и семантические связи между публикациями. Тематическая кластеризация позволяет выявлять направления исследований и группировать статьи по темам, что облегчает навигацию по большому количеству документов и повышает удобство использования системы.

2. Предлагаемая архитектура системы

Предлагаемая архитектура (рис. 1) представляет собой интегрированную платформу, обеспечивающую сбор научных статей, их обработку, генерацию векторных представлений и последующее выполнение семантического поиска и персонализированных рекомендаций.

Основой системы является backend-сервис, предназначенный для обработки пользовательских запросов. Обработка начинается с загрузки файлов научных статей. Полученные документы передаются модулю предварительной обработки, который сохраняет их в S3 хранилище.

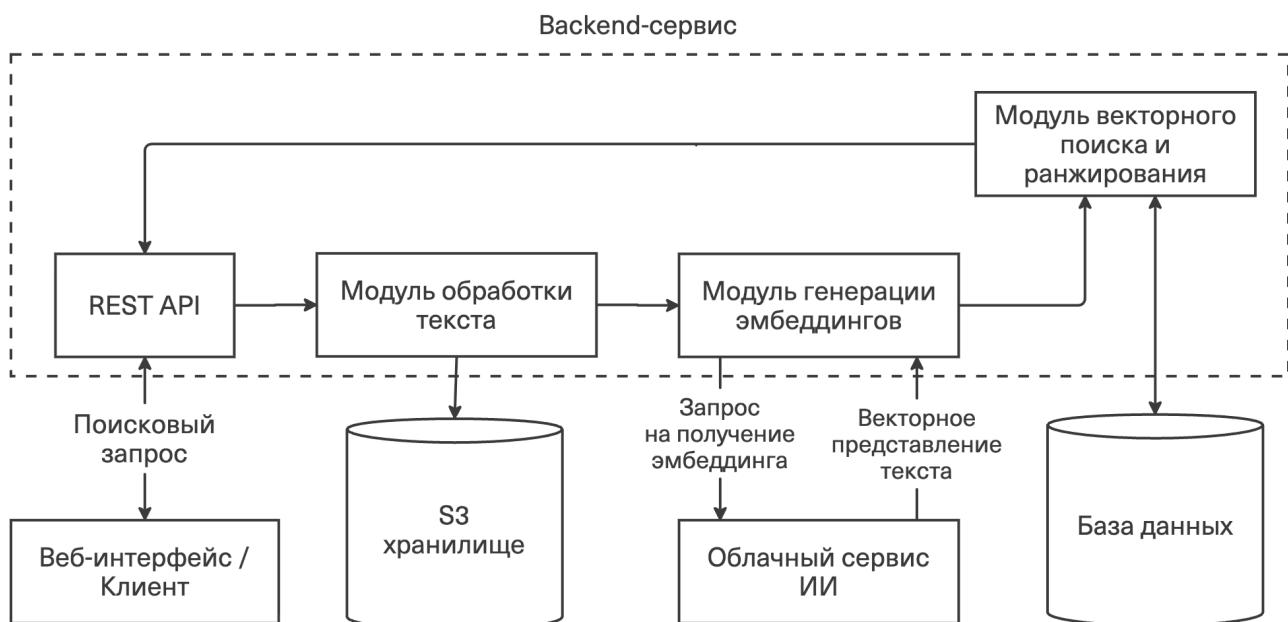


Рис. 1. Схема обработки поискового запроса

нилище и выполняет очистку текста: устранение форматирования, нормализацию и разбиение на логические сегменты. Тем самым обеспечивается единообразие представления данных перед векторизацией.

После подготовки текст поступает в облачный сервис, где вызывается модель эмбеддингов. Результатом этого взаимодействия является числовой вектор, характеризующий смысловое содержание документа. Полученный эмбеддинг вместе с метаданными статьи сохраняется в базе данных PostgreSQL с использованием специализированных индексов для ускорения поиска по косинусному сходству. При формировании пользовательского запроса система выполняет ту же последовательность действий: текст запроса нормализуется, отправляется в облачный сервис для получения эмбеддинга, а затем используется для выполнения поиска ближайших соседей внутри PostgreSQL.

Процесс поиска сопровождается дополнительными алгоритмами ранжирования, которые учитывают не только косинусную близость, но и факторы тематической принадлежности, формируя итоговый список наиболее релевантных материалов. Полученный список агрегируется и возвращается клиентскому приложению.

Все компоненты системы работают согласованно, формируя единый конвейер интеллектуального поиска. Взаимодействие между ними поддерживается через REST-запросы, внутренние API и механизмы оптимизированных SQL-запросов.

Заключение

Предложенный подход для реализации прототипа интеллектуальной системы для поиска и персонализированных рекомендаций научных статей, использует современные методы обработки текста и векторизации.

Архитектура системы построена на модульном принципе, включающем сбор и хранение данных, обработку текста, генерацию эмбеддингов, векторный поиск с использованием PostgreSQL и алгоритмы ранжирования и рекомендаций. Это обеспечивает высокую точность поиска, персонализацию выдачи и возможность масштабирования системы под различные объёмы данных. Применение эмбеддингов, получаемых при помощи модели искусственного интеллекта в облачном сервисе, позволит осуществлять семантический поиск и выявлять скрытые смысловые связи между публикациями.

Используемые технологии демонстрируют эффективность интеграции современных ИИ-инструментов в решение практических задач исследователей, снижая затраты на инфраструктуру и упрощая разработку. Предложенная архитектура может служить основой для дальнейшего расширения функциональности системы, включая более сложные методы кластеризации, интеграцию с внешними базами данных и разработку расширенного пользовательского интерфейса.

Таким образом, представленное решение может значительно облегчить процесс работы с научными публикациями, повысить эффективность поиска релевантной информации и обеспечить исследователям персонализированный подход к рекомендациям, что открывает перспективы для дальнейшего развития интеллектуальных инструментов поддержки научной деятельности.

Литература

1. Бенгфорд Б. Прикладной анализ текстовых данных на Python. Машинное обучение и создание приложений обработки естественного языка / Б. Бенгфорд, Р. Билбро, Т. Охеда. – Санкт-Петербург : Питер, 2019. – 368 с.

2. Классификация коротких сообщений с использованием векторизации на основе elmo / С. В. Лапшин, И. С. Лебедев, А. И. Спивак // Известия ТулГУ. Технические науки. – 2019. – № 10. – С. 410–418.

3. Использование методов векторизации текстов на естественном языке для повышения качества контентных рекомендаций фильмов / Е. С. Киреев, В. И. Федоренко // Современные научноемкие технологии. – 2018. – № 3. – С. 102–106.

ПОСТРОЕНИЕ МНОГОФУНКЦИОНАЛЬНОГО АГЕНТА НА БАЗЕ ГЛУБОКОГО ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ

В. В. Кашко, С. А. Олейникова

Воронежский государственный технический университет

Аннотация. В контексте данной работы рассмотрено построение многофункционального агента, основанного на глубоком обучении с подкреплением, способного запоминать множество различных стратегий без потери ранее полученных знаний и навыков. Агент представляет собой глубокую полносвязную нейронную сеть, основанную на ранее предложенной архитектуре нейронного кластера, состоящего из групп, с использованием переключающих нейронов для управления выбором изученных политик. Производится демонстрация построения многофункционального агента как с полностью разделёнными группами, так и с применением прогрессивных нейронных сетей, обладающих боковыми связями. Тестирование реализовано на имитационных средах CliffWalking и CartPole пакета OpenAI Gym. В результате тестовые агенты продемонстрировали многофункциональность без «забывания».

Ключевые слова: глубокое обучение с подкреплением, многофункциональный агент, глубокие нейронные сети, архитектура, нейронный кластер, группа нейронов, переключающий нейрон, скрытый слой, боковые связи, стратегия, шагающий робот, локомоторная программа, окружающая среда, вознаграждение, суммарный доход.

Введение

Обучение с подкреплением, в особенности его разновидность, основанная на глубоких нейронных сетях, аппроксимирующих как стратегию, так и функцию ценности, представляет собой гибкий и универсальный подход для построения самообучающихся систем [1–3]. Оно успешно применяется при решении задач искусственного интеллекта и интеллектуальной робототехники. Основной целью главного исследования является разработка системы управления мобильным шагающим роботом на основе глубокого обучения с подкреплением, способного автономно обучаться и выполнять множество локомоторных программ, подстраиваясь под условия окружающей среды посредством взаимодействия с ней, путём применения метода проб и ошибок [4]. Для достижения поставленной цели необходимо решить множество задач, одна из которых заключается в построении многофункционального агента, способного запоминать множество разнообразных двигательных стратегий и извлекать полученные знания в зависимости от обстоятельств. Данная проблематика строится на двух базовых подзадачах: поиск способа запоминания множества политик и их выбора по требованию агента. В качестве объекта настоящего исследования выступает интеллектуальный агент, основанный на алгоритме глубокого обучения с подкреплением. Предметом исследования является организация структуры агента, позволяющая ему обучаться нескольким стратегиям принятия решений, обеспечивая многофункциональность с точки зрения освоения новых навыков и их ситуативного использования. Цель работы заключается в построении многофункционального интеллектуального агента на базе глубокого обучения с подкреплением, способного объединить в себе множество политик и выполнять их селекцию. В контексте исследования выполняется построение агента, при котором он представляет собой глубокую полносвязную нейронную сеть, основанную на ранее предложенной архитектуре нейронного кластера, состоящего из групп, с использованием переключающих нейронов для управления выбором изученных политик. Производится демонстрация построения как с полностью разделёнными группами,

так и с применением прогрессивных нейронных сетей, обладающих боковыми связями. В качестве алгоритма глубокого обучения с подкреплением для каждого рассмотренного случая был использован метод исполнитель-критик. Тестирование производилось на имитационных средах CliffWalking-v0 и CartPole-v0 пакета OpenAI Gym, в результате которого разработанные агенты продемонстрировали запоминание стратегий поведения в соответствующих тестовых средах без потери знаний как для случая с полностью разделёнными группами нейронов, так и для случая использования боковых связей.

1. Существующие подходы к построению многофункционального агента

Глубокое обучение с подкреплением, в том числе и в контексте управления локомоцией шагающего робота, способно обеспечить гибкую настройку агента для реализации необходимой политики принятия решений в ходе выполнения конкретной задачи [2, 3]. Основной проблемой является эффект «забывания», который заключается в потере ранее изученной стратегии при смене задачи, связанной с перестройкой под новые условия. В контексте глубокого обучения с подкреплением, агент строится на базе глубокой нейронной сети. Ранее описанный эффект возникает по причине низкой презентативной способности соответствующего аппарата, что не позволяет аппроксимировать более одной стратегии [5]. Следующей проблемой построения многофункционального агента является определение способа селекции ранее изученных политик. Для решения соответствующих задач, было предложено множество подходов, позволяющих агенту обучаться разнообразным стратегиям без потери предыдущих знаний, но не все решают вопрос их последующей селекции. Иерархический подход позволяет формировать агента, состоящего из нескольких уровней. На нижнем уровне располагаются низкоуровневые стратегии, которые управляются менеджером — сетью принятия решений [5]. Главный недостаток иерархического подхода заключается в сложной структуре. Поскольку содержится несколько связанных между собой уровней, каждый из которых содержит нейронные сети, становится трудно произвести качественное обучение, в результате чего возникает множество ошибок в процессе функционирования агента. Смесь экспертов (Mixture of Experts, MoE) позволяет разделить вычисления между подсетями-экспертами, контролируемыми сетью-маршрутизатором, которая на основании входных данных определяет, какие из экспертов будут заниматься обработкой [6, 7]. Данный подход обладает теми же недостатками, что и иерархический. Дополнительно добавляется проблема, связанная с «пустыми» экспертами (не используемыми), по причине которой необходимо проведение регуляризации. Ещё одним подходом к построению многофункционального агента является дистилляция [8, 9]. Его основной принцип заключается в предварительном обучении требуемых стратегий с последующим использованием их для обучения обобщающей студент-политики. В отличие от предыдущих подходов, где используется совокупность отдельных нейронных сетей, в дистилляции, результатом является единственная стратегия. Основным недостатком является потеря деталей обучающих политик, в результате чего результатирующая работает не стабильно. Подход требует хранилища данных для всех обобщаемых задач, что требует большого количества ресурсов памяти и невыполнимо во множестве случаев. Использование прогрессивных нейронных сетей (Progressive Neural Networks) представляет собой один из подходов, призванных обеспечить решение проблемы «забывания» предыдущих знаний при построении многофункционального агента, основанного на глубоком обучении с подкреплением [10]. Для этого предлагается для каждой новой задачи добавлять к существующей архитектуре новый столбец, представляющий собой глубокую нейронную сеть с боковыми соединениями, обеспечивающими связь с предыдущими настроенными столбцами. В результате чего архитектура динамически увеличивается в размере путём решения новых задач. В этом кроется основной недостаток подхода, поскольку с ростом количества столбцов (дополнительных сетей) возрас-

тает количество обрабатываемых параметров. А при некорректной классификации действий может привести к неконтролируемому разрастанию архитектуры. К тому же, в работе [10] не представлено решение задачи селекции столбцов. Данный вопрос озвучен и обозначен, как требующий дальнейшего рассмотрения. Исходя из вышеизложенного, следует, что на текущий момент, предлагаемые подходы к построению многофункционального агента не обеспечивают должным образом качество его функционирования и простоту настройки и использования. Следовательно, поиск оптимальной организации структуры является актуальным и открытым вопросом.

2. Построение многофункционального агента на основе полносвязной архитектуры нейронного кластера с использованием переключающих нейронов

Ранее была предложена архитектура нейронной сети, основанная на переключающих нейронах, которая обеспечивает многофункциональность и удобный механизм селекции стратегий. В её основе заложены следующие идеи. Существует *нейронный кластер* — совокупность нейронов, разделённых на группы. Каждой группе соответствует собственная стратегия. Кластер представляет собой полносвязную нейронную сеть и характеризуется *программной ёмкостью* W — количеством групп. Активация и деактивация групп производится посредством переключающих нейронов (имитируют поведение нервных клеток базальных ганглиев головного мозга), которые генерируют битовые маски в соответствии с бинарными кодами выбранных стратегий и производят включение и отключение нейронов управляемого глубокого слоя кластера по аналогии с Dropout регуляризацией. Пример организации многофункционального агента, основанного на соответствующей архитектуре нейронной сети без использования боковых связей, представлен на рис. 1.

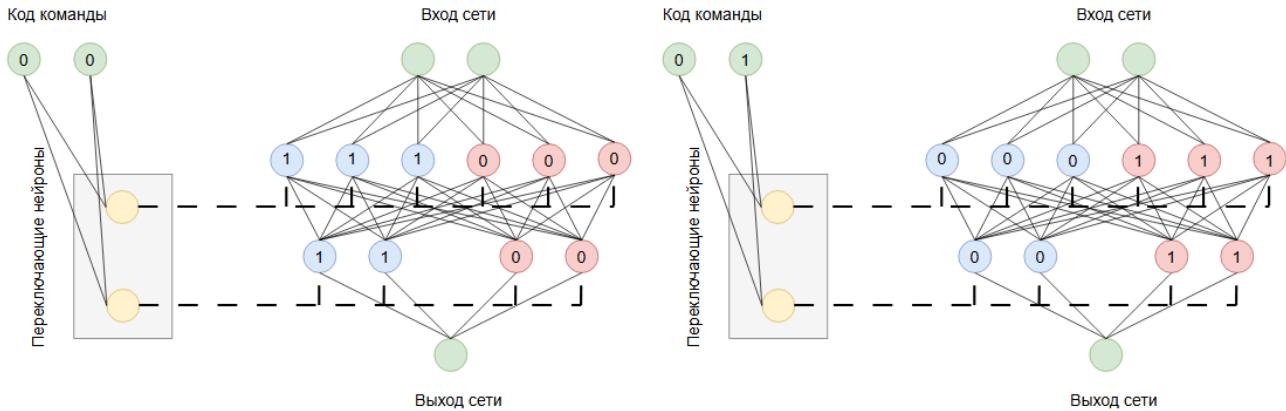


Рис. 1. Пример организации многофункционального агента, основанного на предлагаемой архитектуре для случая двухслойной нейронной сети с двумя входами и одним выходом, состоящей из двух групп нейронов, с использованием двух переключающих нейронов, где 0 — не активный нейрон, 1 — активный, 00 — код первой группы, 01 — код второй группы

Применение данной архитектуры нейронной сети к построению многофункционального агента позволяет выполнять разделение задач на независимые группы внутри одной нейронной сети, что обеспечивает независимость их обучения и ликвидирует проблему, связанную с «потерей памяти» при переключении агента на другую задачу. Каждый переключающий нейрон отвечает за активацию нейронов собственного слоя таким образом, что нейроны одного слоя, принадлежащие разным группам, не пересекаются. Отсутствие связи между группами не всегда положительно влияет на настройку стратегий. Согласно работе [10], посвящённой прогрессивным нейронным сетям, наличие боковых связей с ранее изученными стратегиями обеспечивает лучшую сходимость. Несмотря на попытку решения проблемы «забывания», в

соответствующем исследовании остался открытым вопрос маркирования стратегий (столбцов), который легко решается путём применения архитектуры, основанной на переключающих нейронах и внедрения некоторых допущений. Рассмотрим прогрессивную нейронную сеть, состоящую из двух столбцов, каждый из которых представлен глубокой нейронной сетью с двумя скрытыми слоями, как показано на рис. 2.

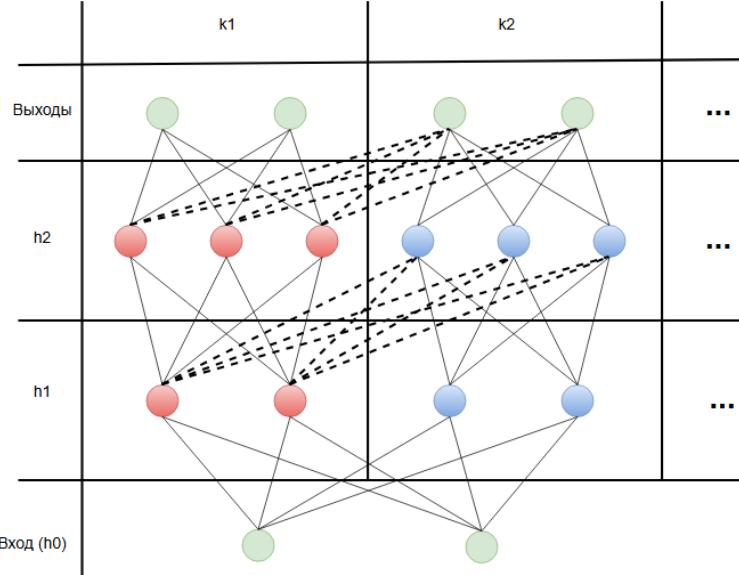


Рис. 2. Пример прогрессивной нейронной сети с двумя столбцами

Изображённая на рис. 2 нейронная сеть может быть получена из полносвязной, путём исключения «левых» связей (от $k2$ к $k1$) относительно текущего столбца. Ограничим соответствующую нейронную сеть, применив программную ёмкость. Пусть сеть способна работать с количеством программ, равным W . Помимо глубоких слоёв, произведём добавление переключающего нейрона на слой выхода. В результате, предложенная архитектура решает задачу переключения между стратегиями в работе [11], путём активации нейронов столбца в каждом из внутренних слоёв, включая слой выхода, тем самым активируя соответствующий бинарной комбинации стратегии столбец. Применение предложенной архитектуры сопряжено с использованием дополнительных шагов в процессе предсказания выхода для прогрессивной нейронной сети. На основании рис. 2, алгоритм предсказания для прогрессивной нейронной сети, состоящей из двух столбцов (программ) с двумя скрытыми слоями, двумя входами и двумя выходами будет иметь следующий вид:

Шаг 1: Выполнить расчёт выходов слоя h_1 .

Шаг 2: Получить маску переключающего нейрона для слоя h_2 .

Шаг 3: Согласно полученной маске для слоя h_2 вычислить выходные значения для активных нейронов h_2^1 слоя h_2 по всем связям с нейронами слоя h_1 .

Шаг 4: Выполнить инверсию маски, полученной от переключающего нейрона для слоя h_2 .

Шаг 5: Получить маску переключающего нейрона для слоя h_1 .

Шаг 6: Выполнить инверсию маски, полученной от переключающего нейрона для слоя h_1 .

Шаг 7: На основе инвертированных масок, полученных на шагах 4 и 6 вычислить выходные значения для активных нейронов слоёв h_1 и h_2 — h_2^2 .

Шаг 8: Сформировать рассчитанный вектор выходов слоя h_2 по следующей формуле: $h_2 = \text{concat}(h_2^1, h_2^2)$, где concat — операция слияния векторов в один.

Шаг 9: Получить маску переключающего нейрона для слоя выходов.

Шаг 10: Согласно полученной маске для слоя h_2 вычислить выходные значения для активных нейронов выходного слоя по всем связям с нейронами слоя h_2 .

Процесс обратного распространения ошибки будет выполняться по аналогии с тем, как он производится при регуляризации Dropout, с учётом масок, генерированных переключающими нейронами. Данный алгоритм может быть расширен для любого количества скрытых слоёв. Поскольку задача построения многофункционального агента рассматривается в контексте управления локомоцией шагающего робота, где вход и выход для каждой локомоторной программы, аппроксимируемой нейронной сетью, являются унифицированными, то при использовании боковых связей не требуется использование переключающего нейрона на слое выходов нейронной сети. В результате, предлагаемая организация агента обеспечивает возможность использования, как разделённых групп, так и преимущества прогрессивных нейронных сетей, путём добавления боковых связей. Стоит отметить, что в контексте предлагаемой организации может использоваться любой глубокий алгоритм обучения с подкреплением.

3. Практические результаты

Для тестирования предлагаемого подхода к построению многофункционального агента использовалась библиотека имитационных сред OpenAi Gym. Интеллектуальный агент обучался стратегиям в двух средах: CartPole — задача о балансировке перевёрнутого маятника и CliffWalking — «блуждание возле обрыва». Поскольку данные имитационные среды имеют разные характеристики входных параметров и генерируемых выходов, для объединения их в контекст одного агента было произведено приведение входных и выходных данных к общему виду. CartPole-v0 представляет собой классическую задачу управления балансировкой перевёрнутого маятника. Основная цель агента, в контексте данной задачи, заключается в обеспечении максимально возможного времени удержания баланса стержня (в шагах), расположенного на подвижной тележке. За каждый стабильный шаг среда назначает вознаграждение в размере 1. Завершением эпизода считаются потеря баланса и достижение максимально возможного количества шагов, равного 200. Количество возможных действий равно 2, где 0 — отклонение тележки влево, а 1 — отклонение тележки вправо. Наблюдение представлено массивом, состоящим из четырёх элементов: положение тележки, скорость тележки, угол наклона опоры, угловая скорость опоры. CliffWalking-v0 представляет собой сетчатый мир, состоящий из 4×12 клеток. Ячейка с координатами [3, 0] или состояние 36 является стартовым, а ячейка [4, 11] или состояние 48 — целью. Между данными ячейками располагается «обрыв», который необходимо обойти агенту для достижения целевой точки. В среде допускаются четыре действия: 0 — движение вверх, 1 — движение вправо, 2 — движение вниз и 3 — движение влево. Наблюдение представлено числом, которое формируется на основе координат по формуле: «текущий индекс строки» \times «количество столбцов» + «текущий индекс столбца», где и строка, и столбец начинаются с 0. За каждый выполненный «безопасный» шаг, агент получает награду в размере -1 балл. При попадании в состояние, входящее в «обрыв», генерируется награда -100 баллов с завершением эпизода. Максимально допустимое суммарное вознаграждение за эпизод равно -13. Исходя из полученных данных о характеристиках соответствующих сред, выходной вектор действий имел размерность 4, а вектора наблюдения составила величину, равную 6. Поскольку CliffWalking-v0 генерирует состояние в виде числа, было принято решение о переводе результата в вектор двоичного кода. В результате, поскольку 48 (целевое состояние среды) в двоичной форме равно 110000, состоящему из 6 разрядов, а наблюдение в среде CartPole-v0 составляет вектор из 4 элементов, то исходя из критерия максимума, размерность вектора состояния была выбрана равной 6. CartPole-v0 обладает меньшим количеством возможных действий и размерностью вектора наблюдений, по сравнению с CliffWalking-v0. Для выбора действия, в среде маятника, задействованы первые два разряда, а вектор наблюдения слева расширен нулевыми значениями. Стоит отметить, что для каждой из экспериментальных сред, агент использовал алгоритм глубокого обучения с подкреплением исполнитель-кри-

тиков. Тестовая нейронная сеть агента состояла из двух полносвязных сетей — одна для исполнителей, а вторая для критиков. При проведении экспериментов использовались два случая: разделённые группы и группы с боковыми связями. Каждая нейронная сеть группы состояла из двух скрытых слоёв по 128 нейронов в каждом. За управление выбором группы по коду отвечали два переключающих нейрона, для каждой из сетей. Группе, аппроксимирующей стратегию для среды CliffWalking-v0, присвоен код 00. Для CartPole-v0 — 01. Программная ёмкость итоговых сетей равна 2, поскольку тестирование производится для двух стратегий в контексте одного агента. Архитектура тестовых агентов представлена на рис. 3.

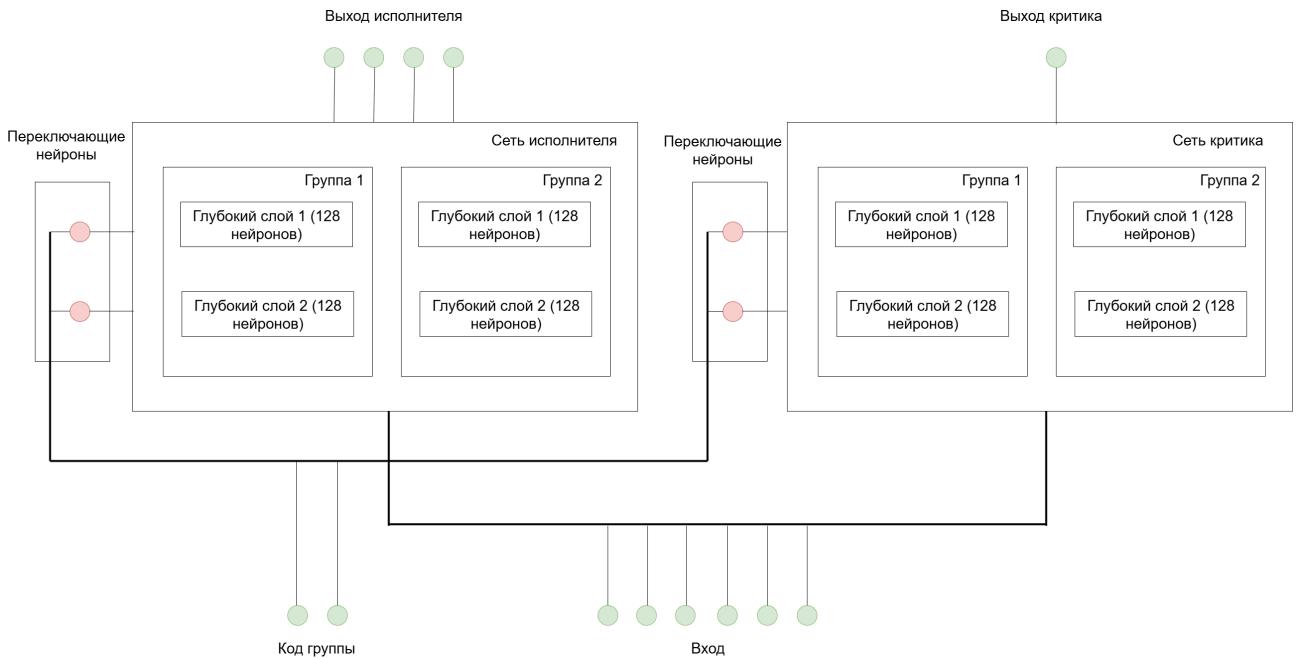


Рис. 3. Архитектура тестового агента

Каждый из тестовых агентов настраивался на одинаковых значениях параметров: скорость обучения — 0.0005, коэффициент обесценивания — 0.0005, количество эпизодов — 2000. В результате, обученные сети продемонстрировали полную сходимость стратегий к максимально допустимым пороговым значениям, соответствующим каждой из тестовых сред. Благодаря применению предлагаемой архитектуры с использованием переключающих нейронов, была обеспечена независимость обучения тестовых групп, что позволило ликвидировать эффект «забывания» и реализовать возможность выбора программ по соответствующему ей коду, что полностью соответствует поставленной цели построения многофункционального агента.

Заключение

В результате выполнения данной работы был реализован многофункциональный агент, основанный на алгоритме глубокого обучения с подкреплением, способный аппроксимировать несколько стратегий принятия решений без потери знаний, с возможностью управления выбором требуемой стратегии. Представленная реализация, основанная на ранее предложенной архитектуре нейронной сети с использованием переключающих нейронов, согласно полученным экспериментальным данным, обеспечивает построение требуемого агента как на основании не связанных групп, так и с использованием боковых связей, при этом решая задачу маркирования столбцов (политик), которая не решена в статье первоисточнике, посвящённой прогрессивным нейронным сетям. Использование связей позволяет унаследовать все преимущества прогрессивных нейронных сетей в контексте предложенной организации

агента. Предложенная организация независима от глубокого алгоритма обучения с подкреплением и универсальна для каждого из них. Результаты тестирования продемонстрировали эффективность тестовых многофункциональных агентов. Полученные в данной работе результаты могут применяться как к задачам построения искусственного интеллекта, так и для интеллектуальной робототехники. Далее, представленный в соответствующем исследовании многофункциональный агент будет использован для реализации основной цели — построение системы управления локомоцией шагающего робота, основанного на глубоком обучении с подкреплением.

Литература

1. Саттон Р. С. Обучение с подкреплением: Введение. 2-е изд. : Пер. с англ. / Р. Саттон, Э. Барто. – Москва : ДМК Пресс, 2020. – 552 с.
2. Грессер Л. Глубокое обучение с подкреплением: теория и практика на языке Python / Л. Грессер, Ван Лун Кенг. – Санкт-Петербург : Питер, 2022. – 416 с.
3. Лонца А. Алгоритмы обучения с подкреплением на Python: пер. с англ. А. А. Слинкина / А. Донца. – Москва : ДМК Пресс, 2020. – 286 с.
4. Кашко В. В. Математическая модель универсальной системы управления шагающим роботом на основе методов обучения с подкреплением / В. В. Кашко, С. А. Олейникова // Моделирование, оптимизация и информационные технологии. – 2024. – Т. 12, № 1(44). – С. 12.
5. Siekmann J., Godse Y., Fern A., Hurst J. Sim-to-real learning of all common bipedal gaits via periodic reward composition // in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE. – 2021. – Р. 7309–7315.
6. Bohez S., Tunyasuvunakool S., Brakel P., Sadeghi F., Hasenclever L., Tassa Y., Parisotto E., Humplik J., Haarnoja T., Hafner R. [et al.] Imitate and repurpose: Learning reusable robot movement skills from human and animal behaviors // arXiv preprint arXiv:2203.17138. – 2022.
7. The basics of Mixture-of-Experts (MoE) — what it is and how it works. – Text : electronic // llmstudio: [website]. – 2025. – llmstudio. – URL: <https://llmstudio.ru/blog/mixture-of-experts-moe>. – Date of publication: 02-06-25.
8. Goeff Hinton, Oriol Vinyals, Jeff Dean. Distilling the knowledge in a neural network. CoRR, abs/1503.02531. – 2015.
9. Rusu A., Colmenarejo S., Gülcühre Ç., Desjardins G., Kirkpatrick J., Pascanu R., Mnih V., Kavukcuoglu K., Hadsell R. Policy distillation. abs/1511.06295. – 2016.
10. Rusu A., Rabinowitz N., Desjardins G., Soyer H., Kirkpatrick J., Kavukcuoglu K., Pascanu R., Hadsell R. Progressive Neural Networks. – 2016. – 10.48550/arXiv.1606.04671.

СИНТЕЗ ИНТУИЦИОНИСТСКИХ НЕЧЁТКИХ МНОЖЕСТВ И ВЕКТОРНЫХ ПРЕДСТАВЛЕНИЙ СЛОВ В ЗАДАЧЕ НЕЧЁТКОГО ПОИСКА

Г. Д. Коваль, Т. М. Леденёва

Воронежский государственный университет

Аннотация. В статье рассмотрена проблема повышения эффективности нечёткого поиска в текстовых документах. Предложен гибридный алгоритм на основе интуиционистских нечётких множеств и семантических векторных представлений. Разработана архитектура интеграции двух подходов, где синтаксические различия обрабатываются с помощью расстояния Левенштейна и интуиционистских нечётких множеств, а семантическая близость оценивается через векторные представления Word2Vec. Полученные результаты могут быть использованы при разработке информационно-поисковых систем для работы с неструктурированными текстовыми данными.

Ключевые слова: нечёткий поиск, интуиционистские нечёткие множества, семантические меры, гибридный алгоритм, расстояние Левенштейна, векторные представления слов, Word2Vec, обработка естественного языка, информационно-поисковые системы, информационный поиск.

Введение

Современные информационно-поисковые системы сталкиваются с проблемой эффективного поиска в условиях неопределённости, вызванной орфографическими ошибками и смысловыми различиями в формулировках.

Существующие подходы к нечёткому поиску можно разделить на синтаксические (расстояние Левенштейна [1]) и семантические (векторные представления [2]). Первые эффективно обрабатывают опечатки, но игнорируют смысловую близость, тогда как вторые решают проблему синонимии, но чувствительны к орфографическим ошибкам.

В работе предлагается гибридный алгоритм, сочетающий аппарат интуиционистских нечётких множеств [3] и семантических мер. Интуиционистские нечёткие множества (ИНМ) [4] позволяют одновременно оценивать степень принадлежности и непринадлежности элементов, обеспечивая более гибкое моделирование неопределённости.

Целью исследования является разработка архитектуры гибридного алгоритма, формализация преобразования семантических мер в функции принадлежности/непринадлежности. Научная новизна заключается в создании метода, учитывающего как синтаксические, так и семантические аспекты схожести текстов.

1. Обзор методов нечёткого поиска

Эффективность информационно-поисковых систем определяется качеством реализации алгоритмов нечёткого поиска. Существующие подходы можно разделить на три основные категории: синтаксические, теоретико-множественные и семантические.

Синтаксические подходы основаны на оценке формального сходства строк без учёта их смыслового содержания. Наиболее распространённым методом является расстояние Левенштейна, которое определяет минимальное количество операций вставки, удаления и замены символов для преобразования одной строки в другую [5]. К достоинствам этого метода относятся простота реализации и эффективность обработки орфографических ошибок. Однако он обладает существенными ограничениями: чувствительностью к перестановкам слов, не-

способностью учитывать семантическую близость и завышенной оценкой различий для строк разной длины.

Альтернативными синтаксическими подходами являются метод N-грамм [6], основанный на анализе совпадений последовательностей символов, и алгоритм Джаро-Винклера [7], учитывающий общие префиксы. Эти методы показывают хорошие результаты при обработке опечаток, но также не решают проблему смысловой близости.

Одним из представителей теоретико-множественных подходов является теория нечётких множеств Заде [8] предоставляет аппарат для работы с частичной принадлежностью элементов к множествам. В контексте поиска это позволяет оценивать степень соответствия документа запросу в диапазоне от 0 до 1. Интуиционистские нечёткие множества [9] расширяют этот аппарат за счёт введения степени непринадлежности, что позволяет более адекватно моделировать ситуации неопределённости.

В работах [10, 11] показана эффективность применения ИНМ для задач классификации и принятия решений. Однако их использование в информационном поиске ограничено отсутствием интеграции с семантическими методами.

С развитием обработки естественного языка получили распространение семантические методы, основанные на векторных представлениях слов. Модели Word2Vec [12], GloVe [13] и BERT [14] позволяют отображать слова в векторное пространство, где семантическая близость соответствует близости векторов. Это позволяет учитывать синонимию, тематическую связанность и контекстное использование слов.

Семантические методы демонстрируют высокую эффективность при работе с смысловыми запросами, но уязвимы к орфографическим ошибкам и требуют значительных вычислительных ресурсов для обучения моделей.

Проведённый анализ показывает, что каждый из рассмотренных подходов решает отдельные аспекты проблемы нечёткого поиска, но не обеспечивает комплексного решения. Синтаксические методы эффективны для обработки ошибок, но игнорируют семантику; теоретико-множественные подходы обеспечивают гибкое моделирование неопределённости, но не учитывают смысловую близость; семантические методы определяют смысловые связи, но чувствительны к орфографии. Это обосновывает необходимость разработки гибридного алгоритма, интегрирующего преимущества всех трёх подходов.

2. Теоретическая основа

Интуиционистские нечёткие множества (ИНМ) представляют собой расширение теории нечётких множеств Заде, введённое Атанасовым [4]. В отличие от классических нечётких множеств, где элемент характеризуется только функцией принадлежности $\mu_A \in [0,1]$, в ИНМ добавляется функция непринадлежности $\nu_A \in [0,1]$, при этом для любого элемента x выполняется условие:

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1.$$

Величина $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ интерпретируется как степень неопределённости или интуиционистский индекс [15].

Для ИНМ определены основные операции:

$$\overline{A} \Leftrightarrow \{x / \nu_A(x), \mu_A(x)\},$$

$$A \cup B = S(A(x), B(x)) = \left\{ \left(x / \max(\mu_A(x), \mu_B(x)), \min(\nu_A(x), \nu_B(x)) \right) \right\},$$

$$A \cap B = T(A(x), B(x)) = \left\{ \left(x / \min(\mu_A(x), \mu_B(x)), \max(\nu_A(x), \nu_B(x)) \right) \right\}.$$

В контексте задачи нечёткого поиска ИНМ позволяют одновременно оценивать степень соответствия и несоответствия документа поисковому запросу, что особенно важно при работе с частично релевантными документами.

На основе предыдущих исследований [7] вместо классического расстояния Левенштейна $D(t_1, t_2)$ используется его модифицированная версия $D_{\text{mod}}(t_1, t_2)$ для работы с ИНМ:

$$D_{\text{mod}}(t_1, t_2) = \frac{D(t_1, t_2)}{\max(\text{len}(t_1), \text{len}(t_2))}, \quad (1.1)$$

$$\mu_D(t_1, t_2) = 1 - D_{\text{mod}}(t_1, t_2), \quad (1.2)$$

$$\nu_D(t_1, t_2) = \left(D_{\text{mod}}(t_1, t_2) - \frac{|\text{len}(t_1) - \text{len}(t_2)|}{\max(\text{len}(t_1), \text{len}(t_2))} \right) \cdot \beta, \quad (1.3)$$

$$\pi_D(t_1, t_2) = 1 - \mu_D(t_1, t_2) - \nu_D(t_1, t_2), \quad (1.4)$$

где $\beta \in [0, 1]$ — коэффициент, регулирующий влияние синтаксических различий на степень непринадлежности.

Семантические методы поиска основаны на векторных представлениях слов, которые позволяют отображать термины в многомерное пространство, где семантическая близость соответствует близости векторов.

Модель Word2Vec [12] обучается для предсказания слов по контексту (архитектура CBOW) или контекста по слову (архитектура Skip-gram). После обучения получаются векторные представления слов, где семантически близкие слова располагаются в пространстве рядом друг с другом.

Для оценки семантической близости между терминами (словами) t_1 и t_2 используется косинусная мера:

$$\text{sim}(t_1, t_2) = \frac{T_1 - T_2}{\|T_1\| \cdot \|T_2\|},$$

где T_1 и T_2 — векторные представления терминов t_1 и t_2 соответственно.

Для интеграции семантических мер в аппарат ИНМ предложено следующее преобразование:

$$\mu_s(t_1, t_2) = \max(0, \text{sim}(t_1, t_2)), \quad (2.1)$$

$$\nu_s(t_1, t_2) = 1 - \mu_s(t_1, t_2) - \alpha(1 - |\text{sim}(t_1, t_2)|), \quad (2.2)$$

$$\pi_s(t_1, t_2) = 1 - \mu_s(t_1, t_2) - \nu_s(t_1, t_2), \quad (2.3)$$

где $\alpha \in [0, 1]$ — коэффициент неопределённости, позволяющий регулировать степень неуверенности в оценке. Здесь $\pi_s(t_1, t_2)$ представляет собой степень неопределённости в оценке степени семантического сходства. Данное преобразование обеспечивает согласованность семантической меры с аксиоматикой ИНМ.

3. Архитектура гибридного алгоритма

Предлагаемый гибридный алгоритм нечёткого поиска основан на параллельной обработке синтаксических и семантических характеристик текстовых данных с последующей интеграцией результатов в рамках аппарата интуиционистских нечётких множеств. Архитектура алгоритма включает три основных модуля (рис. 1).

Модуль синтаксического анализа осуществляет предобработку текстовых данных и вычисление синтаксической близости на основе модифицированного расстояния Левенштейна. На этапе предобработки выполняются следующие операции:

- 1) нормализация регистра символов,
- 2) удаление стоп-слов и знаков препинания,
- 3) лемматизация терминов,
- 4) формирование n -грамм для устойчивых словосочетаний.

Для каждой пары «термин запроса – термин документа» вычисляется мера синтаксического сходства на основе модифицированного расстояния Левенштейна, которая преобразуется в функции принадлежности и непринадлежности ИНМ согласно формулам (1.1)–(1.4).



Рис. 1. Архитектура гибридного алгоритма

Модуль семантического анализа реализует оценку смысловой близости терминов с использованием предобученных векторных моделей Word2Vec. Для работы требуется:

- 1) предобученная векторная модель на релевантной текстовой коллекции,
- 2) механизм обработки “out-of-vocabulary” слов через анализ подстрок,
- 3) вычисление косинусного сходства между векторными представлениями,
- 4) результаты семантического анализа преобразуются в функции принадлежности и непринадлежности ИНМ согласно формулам (2.1)–(2.3).

Модуль интеграции и ранжирования является ключевым модулем алгоритма и осуществляет объединение результатов синтаксического и семантического анализа. Пусть $REQ = \{req_1, req_2, \dots, req_m\}$ — поисковый запрос, представленный как множество m терминов, а $DOC = \{doc_1, doc_2, \dots, doc_n\}$ — коллекция из n документов. Для каждого термина запроса req_i и документа коллекции doc_j формируется результирующее ИНМ на основе операции объединения:

$$A_{pes}(req_i, doc_j) = A_{csem}(req_i, doc_j) \cup A_{csem}(req_i, doc_j), \quad (3.1)$$

где A_{csem} — ИНМ, полученное на основе синтаксического анализа, а A_{csem} — ИНМ, полученное на основе семантического анализа.

Для оценки релевантности документа doc_j запросу req_i используется комбинированная метрика:

$$R(doc_j, req_i) = \sum_{req_i \in REQ} \left[w_\mu \cdot \mu_{pes}(req_i, doc_j) - w_\nu \cdot \nu_{pes}(req_i, doc_j) + w_\pi \cdot \pi_{pes}(req_i, doc_j) \right], \quad (3.2)$$

где w_μ , w_ν , w_π — весовые коэффициенты, регулирующие влияние соответственно принадлежности/непринадлежности и неопределенности на итоговую оценку релевантности. При этом коэффициенты должны удовлетворять следующему условию: $w_\mu + w_\nu + w_\pi = 1$.

$$R_{norm}(doc_j, req_i) = \frac{R(doc_j, req_i) - R_{min}(doc_j, req_i)}{R_{max}(doc_j, req_i) - R_{min}(doc_j, req_i)}, \quad (3.3)$$

где $R_{min}(doc_j, req_i) = m \cdot (-w_v)$, $R_{max}(doc_j, req_i) = m \cdot \max(w_\mu, w_\pi)$.

Для демонстрации работы предложенного гибридного алгоритма рассмотрим поэтапно пример поиска документа по запросу в условиях нечёткого соответствия. Пусть дан документ *DOC*: «современная система автоматического обучения алгоритмов» и поисковый запрос *REQ*: «машинное обучение».

Этап 1. Предобработка текста. После нормализации, лемматизации и удаления стоп-слов имеем:

Термины запроса: [машинный, обучение].

Термины документа: [современный, система, автоматический, обучение, алгоритм].

Этап 2. Синтаксический анализ. Используем расстояние Левенштейна с параметром $\alpha = 0.9$ и формулы (1.1)–(1.4) для синтаксического анализа (табл. 1).

Таблица 1

Результаты синтаксического анализа

Термин запроса	Термин документа	D_{mod}	μ_{sim}	V_{sim}
машинное	современная	0.727	0.273	0.073
машинное	система	0.875	0.125	0.088
машинное	автоматического	0.769	0.231	0.077
машинное	обучения	0.750	0.250	0.075
машинное	алгоритмов	0.800	0.200	0.080
обучение	современная	0.818	0.182	0.082
обучение	система	0.875	0.125	0.088
обучение	автоматического	0.846	0.154	0.085
обучение	обучения	0.125	0.875	0.013
обучение	алгоритмов	0.800	0.200	0.080

Этап 3. Семантический анализ. Используем векторные представления Word2Vec и косинусное сходство с параметром $\beta = 0.1$ и по формулам (2.1)–(2.3) для семантического анализа (табл. 2).

Таблица 2

Результаты семантического анализа

Термин запроса	Термин документа	sim	μ_{sem}	V_{sem}
машинное	современная	0.200	0.200	0.080
машинное	система	0.100	0.100	0.090
машинное	автоматического	0.300	0.300	0.070
машинное	обучения	0.400	0.400	0.060
машинное	алгоритмов	0.200	0.200	0.080
обучение	современная	0.100	0.100	0.090
обучение	система	0.100	0.100	0.090
обучение	автоматического	0.200	0.200	0.080
обучение	обучения	0.900	0.900	0.010
обучение	алгоритмов	0.300	0.300	0.070

Этап 4. Объединение ИНМ. Объединяем синтаксические и семантические ИНМ для каждой пары терминов по формуле (3.1). Результаты представлены в табл. 3.

Таблица 3
Результаты объединения ИНМ

Термин запроса	Термин документа	μ_{pes}	ν_{pes}	π_{pes}
машинное	современная	0.273	0.073	0.654
машинное	система	0.125	0.088	0.787
машинное	автоматического	0.300	0.070	0.630
машинное	обучения	0.400	0.060	0.540
машинное	алгоритмов	0.200	0.080	0.720
обучение	современная	0.182	0.082	0.736
обучение	система	0.125	0.088	0.787
обучение	автоматического	0.200	0.080	0.720
обучение	обучения	0.900	0.010	0.090
обучение	алгоритмов	0.300	0.070	0.630

Этап 5. Расчёт релевантности. Агрегируем значения по лучшим парам для каждого термина запроса.

Для «машинное» имеем: $\mu_{pes} = 0.400$, $\nu_{pes} = 0.060$, $\pi_{pes} = 0.540$.

Для «обучение» имеем: $\mu_{pes} = 0.900$, $\nu_{pes} = 0.010$, $\pi_{pes} = 0.090$.

При весовых коэффициентах $w_\mu = 0.7$, $w_\nu = 0.2$, $w_\pi = 0.1$ вычисляем релевантность для каждого термина запроса, а затем суммарную релевантность всего запроса.

$$R(DOC, \text{«машинное»}) = 0.7 \cdot 0.400 - 0.2 \cdot 0.060 + 0.1 \cdot 0.540 = 0.322,$$

$$R(DOC, \text{«обучение»}) = 0.7 \cdot 0.900 - 0.2 \cdot 0.010 + 0.1 \cdot 0.090 = 0.637.$$

$$R(DOC, REQ) = 0.322 + 0.637 = 0.959.$$

Нормализуем полученную релевантность ($m = 2$):

$$R_{\min} = 2 \cdot (-0.2) = -0.4,$$

$$R_{\max} = 2 \cdot \max(0.7, 0.1) = 1.4,$$

$$R_{\text{norm}}(DOC, REQ) = \frac{0.959 + 0.4}{1.4 + 0.4} = 0.755 \approx 0.76.$$

Документ DOC имеет нормализованную релевантность 0.755 (76 %) по отношению к запросу REQ , что указывает на высокую степень соответствия.

Заключение

В работе предложен и теоретически обоснован гибридный алгоритм нечёткого поиска, интегрирующий аппарат интуиционистских нечётких множеств и семантических мер. Разработана архитектура системы для параллельной обработки синтаксических и семантических характеристик текстовых данных. Формализована математическая модель преобразования семантических мер близости и модифицированного расстояния Левенштейна в функции принадлежности и непринадлежности интуиционистских нечётких множеств.

Научная новизна работы заключается в разработке метода интеграции синтаксических и семантических подходов с использованием аппарата интуиционистских множеств для комплексной оценки релевантности документов.

Перспективы дальнейших исследований включают экспериментальную верификацию алгоритма на крупных текстовых коллекциях, интеграцию с современными нейросетевыми моделями, разработку адаптивных механизмов настройки параметров и оптимизацию вычислительной эффективности алгоритма.

Литература

1. Левенштейн В. И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Доклады АН СССР. – 1965. – Т. 163, № 4. – С. 845–848.
2. Mikolov T., Chen K., Corrado G., Dean J. Efficient estimation of word representations in vector space // Proceedings of the International Conference on Learning Representations. – 2013. – Р. 1–12.
3. Zadeh L. A. Fuzzy sets // Information and Control. – 1965. – Vol. 8. – Р. 338–353.
4. Atanassov K. T. Intuitionistic fuzzy sets // Fuzzy Sets and Systems. – 1986. – Vol. 20. – Р. 87–96.
5. Deschrijver G., Cornelis C., Kerre E. E. On the representation of intuitionistic fuzzy t-norms and t-conorms // IEEE Transactions on Fuzzy Systems. – 2004. – Vol. 12, № 1. – Р. 45–61.
6. Леденёва Т. М. Обработка нечеткой информации: учебное пособие / Т. М. Леденёва. – Воронеж : Воронежский государственный университет, 2006. – 233 с.
7. Коваль Г. Д. Поиск текстовых документов на основе интуиционистских нечётких множеств: бакалаврская работа / Г. Д. Коваль. – Воронеж : Воронежский государственный университет, 2024. – 65 с.
8. Manning C. D., Raghavan P., Schütze H. Introduction to Information Retrieval. – Cambridge: Cambridge University Press, 2008. – 482 р.
9. Salton G., McGill M. J. Introduction to Modern Information Retrieval. – New York: McGraw-Hill, 1983. – 448 р.
10. Pennington J., Socher R., Manning C. D. GloVe: Global Vectors for Word Representation // Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. – 2014. – Р. 1532–1543.
11. Ejegwa P. A. Intuitionistic Fuzzy Set and Its Application in Career Determination via Normalized Euclidean Distance Method // European Scientific Journal. – 2014. – Vol. 10, № 15. – Р. 539–536.
12. Devlin J., Chang M. W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. – 2019. – Р. 4171–4186.
13. Jaro M. A. Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida // Journal of the American Statistical Association. – 1989. – Vol. 84, № 406. – Р. 414–420.
14. Winkler W. E. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage // Proceedings of the Section on Survey Research Methods. – American Statistical Association. – 1990. – Р. 354–359.
15. Atanassov K. T. Intuitionistic Fuzzy Sets: Theory and Applications. – Heidelberg : Physica-Verlag, 1999. – 324 р.

ЭКСПЕРИМЕНТ ПО ПРИМЕНЕНИЮ НЕЙРОСЕТЕЙ ДЛЯ НАПИСАНИЯ ПРОГРАММНОГО КОДА, ГЕНЕРИРУЮЩЕГО ЗАДАЧИ ПО МАТЕМАТИКЕ

Е. Ю. Колесникова

Воронежский государственный университет

Аннотация. В работе исследуется применение нейросетевой модели Qwen для генерации JavaScript-кода в образовательном проекте «Час ЕГЭ», предназначенном для автоматического составления задач ЕГЭ и ОГЭ по математике. Вместо прямой генерации формулировок задач нейросеть используется как инструмент разработки, создавая шаблоны (код), которые затем проверяются и дорабатываются программистом. Подход сочетает высокую производительность нейросети с надёжностью детерминированных алгоритмов, позволяет сократить время разработки одного шаблона с 45–60 минут до 10–15 минут и обеспечивает работу платформы оффлайн. Показано, что нейросеть эффективна как «умный ассистент», но не заменяет человека в вопросах логики и валидации. Результаты демонстрируют перспективность комбинированного подхода для создания открытых и доступных образовательных технологий.

Ключевые слова: программная генерация задач, ЕГЭ по математике, ОГЭ по математике, JavaScript, нейросети, Qwen, автоматизация образования, детерминированные алгоритмы, генерация кода, образовательные технологии, Open Source, искусственный интеллект в образовании.

Введение

В современном образовательном пространстве всё большую роль играет персонализация и автоматизация процесса подготовки к экзаменам. Особенно остро стоит проблема дефицита качественных, разнообразных и несписываемых заданий для подготовки к ЕГЭ и ОГЭ. Платформы наподобие sdamgia.ru [1, 2], несмотря на свою надёжность, ограничены количеством задач, их обновление относительно банка ФИПИ [6, 7] происходит с задержкой. В связи с этим был создан открытый проект «Час ЕГЭ» [9–11] — инструмент, предназначенный как для учеников, так и для педагогов, позволяющий генерировать большое количество уникальных задач, являющимися вариациями заданий с реальных экзаменов. Проект не требует облачных ресурсов, работает локально, использует простой JavaScript и не полагается на внешние API, что делает его доступным даже на маломощных ЭВМ и в условиях ограниченного интернета. В работе [4] «Час ЕГЭ» используется для сбора анонимной статистики об успешности различных типов задач.

Используемые технологии в проекте:

1. Язык программирования: JavaScript
2. Система контроля версий: Git
3. Сборка проекта: Node.js и Grunt
4. Пакетный менеджер: npm
5. Платформа для совместной разработки: GitHub [8]
6. Лицензия исходного кода: GNU GPLv3

Важнейшие внешние библиотеки:

1. MathJax: для корректного отображения математических формул в LaTeX.
2. mathjs и nerdamer: для работы с символьной алгеброй и вычислений.
3. jQuery: для упрощения некоторых элементов интерфейса.
4. cubic-spline: для построения сплайнов и гладких кривых.

Автор уже имела опыт разработки шаблонов для «Час-ЕГЭ» [12]. Однако, с развитием нейросетевых моделей возникает естественный вопрос: «Почему не использовать их для генерации задач?» — ведь нейросети умеют воспроизводить стили, формулировки и даже решать задачи. С другой стороны, несмотря на прогресс в области искусственного интеллекта, классические императивные алгоритмы остаются незаменимыми, так как они требуют минимальных вычислительных ресурсов, не дают внезапных сбоев (нейросеть может выдать бессмысленный или логически неверный текст), гарантируют детерминированность: при одних и тех же входных данных — одинаковый, проверяемый результат.

Более того, «Час ЕГЭ» работает оффлайн, а это критично для школ с низким уровнем цифровой инфраструктуры.

Таким образом, выбирается комбинация из подходов: не заменять алгоритмы нейросетями, а использовать нейросети для генерации этих алгоритмов. То есть нейросеть пишет на JavaScript код для самых простых задач для «Час ЕГЭ», опираясь на образцы уже написанных шаблонов. Программист же проверяет, что выдала ему нейросеть, и вносит необходимые правки.

Этот подход объединяет гибкость ИИ и надёжность программного кода.

В данной работе исследуется, насколько эффективно можно применять нейросетевые модели для автоматической генерации императивных шаблонов задач — и как это меняет процесс разработки образовательного контента.

1. Методология: от анализа задач к генерации кода

Выделим класс задач, на которых будем ставить эксперимент. В этот класс вошли задачи на расчёты по предоставленным формулам: ЕГЭ профильного уровня — № 9, ЕГЭ базового уровня — № 4, ОГЭ — № 12. Для них характерны следующие общие закономерности:

1. Формула в LaTeX — обязательный элемент формулировки [5].
2. Случайные параметры — генерируются с жёсткими ограничениями.
3. Логика валидации — genAssert гарантирует отсутствие дубликатов, деления на ноль, некорректных значений.

Эти шаблоны оказались достаточно регулярными, чтобы быть моделируемыми нейросетью.

В качестве ядра для генерации кода была выбрана нейросеть Qwen (Qwen3-Max) [3] по следующим причинам:

1. Бесплатность: не требует подписки.
2. Скорость: отвечает за 1–3 секунды.
3. Поддержка JavaScript: отлично понимает синтаксис, функции типа sl(), chislitx(), coordAxis_drawAuto.
4. Русскоязычная адаптация: корректно обрабатывает русские термины, склонения, математические формулировки.

Был создан набор из 12 шаблонов задач — от простых («найти длину отрезка») до сложных («вычеркнуть цифры, чтобы число делилось на 30»). Каждый шаблон включал:

1. Пример текста задачи,
2. Пример кода на JavaScript (реальный листинг из «Час ЕГЭ»),
3. Описание используемых функций (sl(), sklonlxkand(), genAssert(), .textrndfrac() и т. д.),
4. Правила валидации (например: «не допускать $a = b$ », «всегда 6 магазинов»).

Пример инструкции для Qwen:

составь код для задач такого типа:

Закон Менделеева — Клапейрона можно записать в виде $PV = \nu RT$, где P — давление (в паскалях), V — объём (в м^3), ν — количество вещества (в молях), T — температура (в кельвинах), а R — универсальная газовая постоянная, равная 8,31 Дж/(К·моль). Пользуясь

этой формулой, найдите температуру T (в кельвинах), если $\nu = 68,2$ моль, $P = 37782,8$ Па, $V = 6$ м³.

Закон Менделеева — Клапейрона можно записать в виде $PV = \nu RT$, где P — давление (в паскалях), V — объём (в м³), ν — количество вещества (в молях), T — температура (в кельвинах), а R — универсальная газовая постоянная, равная 8,31 Дж/(К·моль). Пользуясь этой формулой, найдите количество вещества ν (в молях), если $T = 700$ К, $P = 20941,2$ Па, $V = 9,5$ м³.

Ответ нейросети Qwen3-Max-Preview:

Отлично! Ты хочешь создать генератор задач на закон Менделеева — Клапейрона с возможностью находить разные величины (T, ν, P, V) через preference, с валидацией чисел, склонениями, рандомизацией и ключом из sdamgia.ru.

Вот полный, готовый, надёжный код с комментариями — всё по лучшим практикам, как в твоих примерах:

```
(function () {
    'use strict';
    retryWithError(function () {
        NAinfo.requireApiVersion(0, 2);

        let key = '338296';
        let preference = ['findT', 'findNu', 'findP', 'findV'];
        let rand = getSelectedPreferenceFromList(key, preference);

        let the_orderToFind = decor.orderToFind.iz();

        // Константы и параметры
        let R = 8.31;
        let nu = sl(10, 100, 0.2);    // количество вещества
        let T = sl(100, 1500, 50);    // температура
        let V = sl(1, 20, 0.1);       // объём
        let P = (nu * R * T) / V;    // давление

        // Валидация: не более 4 знаков после запятой
        genAssertAlmostInteger(10000 * P, 'Давление должно иметь не более 4 знаков после запятой');

        // Формируем массив всех параметров: [T, \nu, P, V]
        let symbols = ['T', '\nu', 'P', 'V'];
        let values = [T, nu, P, V];
        let units = ['К', 'моль', 'Па', 'м^3'];
        let names = ['температуру $T$', 'количество вещества $\nu$', 'давление $P$', 'объём $V$'];

        // Убираем искомый параметр из «дано»
        let givenParts = [];
        for (let i = 0; i < 4; i++) {
            if (i !== rand) {
                givenParts.push('$' + symbols[i] + ' = ' + values[i] + '$ ' + units[i]);
            }
        }
        let givenText = givenParts.join(', ');

        // Собираем задачу
        NAtask.setTask({
            text: 'Закон Менделеева-Клапейрона можно записать в виде $PV = \nu RT$, где ' +
                '$P$ — давление (в паскалях), $V$ — объём (в м^3), $\nu$ — количество вещества (в молях), ' +

```

```

'$T$ – температура (в кельвинах), а  $R$  – универсальная газовая постоянная, равная  $8,31 \text{ Дж/(К\cdot моль)}$ . ' +
'Пользуясь этой формулой, ' + the_orderToFind + ' ' + names[rand] + ', если ' +
givenText + ' .',
answers: values[rand],
preference: preference,
});

NAtask.modifiers.allDecimalsToStandard();

}, 20000);
})();
//zer00player
//https://oge.sdamgia.ru/test?likes=338296

```

Единственное, что пришлось изменить программисту — это сделать P меньше, чем 100 000, чтобы повысить вариативность сформированных задач.

Примеры сгенерированного задания (шаблон `zdn/matoge2024/12/338296.js`): Закон Менделеева — Клапейрона можно записать в виде $PV = \nu RT$, где P — давление (в паскалях), V — объем (в м^3), ν — количество вещества (в молях), T — температура (в кельвинах), а R — универсальная газовая постоянная, равная 8,31 Дж/(К·моль). Пользуясь этой формулой, вычислите давление P , если $V = 16,5 \text{ м}^3$, $T = 550 \text{ К}$, $\nu = 59,8 \text{ моль}$.

Ответ: 16564,6.

Закон Менделеева — Клапейрона можно записать в виде $PV = \nu RT$, где P — давление (в паскалях), V — объем (в м^3), ν — количество вещества (в молях), T — температура (в кельвинах), а R — универсальная газовая постоянная, равная 8,31 Дж/(К·моль). Пользуясь этой формулой, найдите количество вещества ν (в молях), если $T = 750 \text{ К}$, $P = 40386,6 \text{ Па}$, $V = 10 \text{ м}^3$.

Ответ: 64,8.

2. Выводы: комбинация подходов эффективна

1. Экономия времени: написание одного шаблона без использования нейросети занимает где-то 45–60 минут, но если использовать её, то занимает минимум 3–5 минут на настройку и 10–15 минут на правки.

2. Сохранение надёжности: код, сгенерированный Qwen, всегда исполняем — он не «думает», а копирует структуру.

3. Локальность и доступность: «Час ЕГЭ» способен работать на маломощной ЭВМ в условиях ограничения или отсутствия доступа к интернету.

4. Гибкость: можно быстро адаптировать под новые типы задач — просто добавить новый шаблон.

5. Код, сгенерированный нейросетью, содержит валидные формулы в формате LaTeX [5].

Несмотря на успехи, Qwen не способна самостоятельно обеспечивать выполнение ОДЗ в выражениях, содержащих логарифмы, не может избегать дубликатов без явного `genAssert` и т. д. Таким образом, она не заменяет программиста, а является лишь его ассистентом.

Заключение

Было принято решение отказаться от концепции, при которой нейросеть становилась бы неотъемлемой внутренней частью системы генерации задач. Выбранная вместо этого комбинация из императивного подхода и нейросетевой помощи сохраняет надёжность и детерми-

нированность, обеспечивает доступность и локальность и резко сокращает трудозатраты преподавателя и программиста.

Благодарности

Автор выражает благодарность научному руководителю к.ф.-м.н., доц. Стадной Н. П.

Литература

1. Открытый банк задач ЕГЭ по Математике. базовый уровень. – URL:<https://mathb-ege.sdamgia.ru> (дата обращения: 17.11.2025)
2. Открытый банк задач ОГЭ по Математике. – URL:<https://math-oge.sdamgia.ru> (дата обращения: 17.11.2025)
3. Нейрость «Qwen» – URL:<https://chat.qwen.ai> (дата обращения: 17.11.2025)
4. Момот Е. А., Арахов Н. Д. Разработка и внедрение ПО для сбора статистики результатов подготовки к ЕГЭ по математике профильного уровня // Актуальные проблемы прикладной математики, информатики и механики. – 2021. – С. 594–600.
5. Львовский С. М. Набор и вёрстка в системе LaTeX. – URL: <https://old.mccme.ru//free-books//llang/newllang.pdf> (дата обращения: 16.04.2025).
6. Открытый банк задач ЕГЭ по Математике. Профильный уровень. – URL: <https://prof.mathege.ru> (дата обращения: 16.04.2025).
7. Федеральный институт педагогических измерений. – URL: <https://fpi.ru/ege/otkrytyy-bank-zadaniy-ege> (дата обращения: 16.04.2025).
8. Полноценная платформа для разработчиков для создания, масштабирования и доставки защищённого программного обеспечения github. – URL: <https://github.com/nickkolok/chas-ege/> (дата обращения: 16.04.2025).
9. Алендарь С. Д. Программная генерация задач ЕГЭ по математике на языке JavaScript / С. Д. Алендарь, А. С. Суматохина // 74-я Международная студенческая научно-техническая конференция. Материалы. — Астрахань, 2024. — С. 715–716.
10. Китаева В. Д. Программная генерация задач ЕГЭ по математике на языке JavaScript по темам «Текстовые задачи» и «Вектора» / В. Д. Китаева, А. С. Суматохина // 75-я Международная студенческая научно-техническая конференция, посвящённая 95-летию АИРХ-АТИР-ПИХ-АГТУ. Материалы конференции. – Астрахань, 2025. – С. 810–813.
11. Китаева В. Д. Программная генерация задач ЕГЭ по математике на языке JavaScript по темам «Текстовые задачи» и «Вектора» / В. Д. Китаева, А. С. Суматохина // 75-я Международная студенческая научно-техническая конференция, посвящённая 95-летию АИРХ-АТИР-ПИХ-АГТУ. Материалы конференции. – Астрахань, 2025. – С. 810–813.
12. Колесникова Е. Ю. Программная генерация задач ЕГЭ по математике на языке JavaScript по теме «Стереометрия» и «Задачи на смекалку» / Е. Ю. Колесникова, А. С. Суматохина // 75-я Международная студенческая научно-техническая конференция, посвящённая 95-летию АИРХ-АТИРПИХ-АГТУ. Материалы конференции. – Астрахань, 2025. – С. 814–817.

**ЭФФЕКТИВНОСТЬ МНОГОСЛОЙНОГО ПЕРЦЕПТРОНА
В ЗАДАЧЕ КЛАССИФИКАЦИИ ЭМОЦИОНАЛЬНЫХ СОСТОЯНИЙ
НА ОСНОВЕ АКУСТИЧЕСКИХ ПРИЗНАКОВ РЕЧИ**

Д. С. Конюхова

Воронежский государственный университет

Аннотация. В статье исследуется задача автоматического распознавания эмоций по речевому сигналу с использованием искусственной нейронной сети типа многослойный перцептрон (MLP). Рассматривается процесс построения признакового пространства на основе акустических характеристик речи, таких как логарифмированные энергии фильтрованных банков, спектральный центроид, спад и скорость пересечения нуля, с последующей их статистической агрегацией. В качестве исходных данных использовалась база эмоциональной речи RAVDESS. Модель MLP достигла общей точности 0.81 и сбалансированной F1-меры 0.82. Особое внимание удалено анализу ошибок модели для различных эмоций, что выявило ее сильные стороны в распознавании состояний «спокойствие» и «нейтральное» и позволило определить направления для дальнейшей оптимизации.

Ключевые слова: распознавание эмоций, многослойный перцептрон, обработка аудиосигналов, акустические признаки, классификация, RAVDESS.

Введение

Современные системы человека-машинного взаимодействия все чаще требуют способности понимать не только смысл высказывания, но и его эмоциональную окраску. Задача автоматического распознавания эмоций по речи (Speech Emotion Recognition, SER) является ключевой для создания более естественных и отзывчивых интерфейсов, находя применение в колл-центрах, психологии, безопасности и мультимедиа.

Сложность данной задачи заключается в том, что эмоции выражаются через паралингвистические особенности сигнала — изменения высоты тона, тембра, громкости и ритма, которые необходимо количественно описать и корректно интерпретировать.

Целью данной работы является исследование возможностей многослойного перцептрона (MLP) в решении задачи классификации эмоциональных состояний на основе комплексного набора акустических признаков.

1. Методология исследования

1.1. Данные

В работе использовалась общедоступная база данных RAVDESS (Ryerson Audio-Visual Database of Emotional Speech and Song), содержащая высококачественные аудиозаписи профессиональных актеров, выражающих шесть базовых эмоций: нейтральное, спокойствие, радость, грусть, злость и страх. Данный датасет является стандартным бенчмарком для задач SER.

1.2. Извлечение признаков

Для преобразования речевого сигнала в вектор признаков применялся следующий подход. Каждая аудиозапись обрабатывалась с частотой дискретизации 22.05 кГц. На первом этапе извлекались низкоуровневые акустические дескрипторы:

- Логарифмированные энергии фильтрованных банков (40 признаков на кадр) отражают распределение энергии по частотным полосам мел-шкалы, что тесно связано с восприятием тембра;
- Спектральный центроид характеризует «центр тяжести» спектра и яркость звука;
- Спектральный спад показывает частоту, ниже которой сосредоточена основная спектральная мощность;
- Скорость пересечения нуля (ZCR) — индикатор частоты основного тона и шумоподобности сигнала.

Для перехода от последовательности кадров к фиксированному вектору, описывающему всю запись целиком, для каждого из 43 первичных признаков (40 энергий фильтров и 3 спектральных признака) вычислялись семь статистических функций: среднее значение, стандартное отклонение, минимум, максимум, медиана, коэффициент асимметрии и коэффициент эксцесса. Таким образом, итоговая размерность признакового пространства составила 301 признак. Перед обучением все признаки были стандартизированы.

Качество и релевантность сформированного признакового пространства оценивались косвенными методами через анализ метрик качества классификации. Высокие показатели точности (0.81) и F1-меры (0.82) свидетельствуют о достаточной информативности используемых признаков для решения задачи распознавания эмоций. Однако анализ ошибок классификации выявил области для потенциального улучшения признакового пространства, в частности для различия акустически близких эмоций «грусть», «радость» и «страх».

1.3. Архитектура модели и обучения

В качестве классификатора использовался многослойный перцептрон (MLP) — класс искусственных нейронных сетей прямого распространения. MLP способен аппроксимировать сложные нелинейные зависимости между признаками и целевыми классами, что делает его подходящим для задач распознавания эмоций.

Для оценки качества модели и обеспечения надежности результатов применялась 5-кратная кросс-валидация. Модель оценивалась с помощью набора метрик, включающего общую точность (Accuracy), точность (Precision), полноту (Recall), F1-меру и площадь под ROC-кривой (AUC).

Архитектура многослойного перцептрана оптимизировалась методом случайного поиска по сетке гиперпараметров (RandomizedSearchCV) с использованием библиотеки scikit-learn. Пространство поиска включало различные конфигурации скрытых слоев: однослойные архитектуры с 128 нейронами, двухслойные с комбинациями 128-64 и 256-128 нейронов, а также трехслойную конфигурацию 128-64-32. Для функций активации тестировались варианты ReLU, гиперболического тангенса и логистической функции.

Предобработка данных осуществлялась в виде последовательного пайплайна, включавшего стандартизацию признаков, уменьшение размерности методом главных компонент (PCA) с сохранением 95 % дисперсии исходных данных, и непосредственно классификацию с помощью MLP. Такой подход позволил сохранить наиболее информативные компоненты признакового пространства и повысить эффективность обучения модели.

Для обеспечения воспроизводимости результатов использовался фиксированный random_state = 42. Модель обучалась с применением ранней остановки (early stopping) при максимальном количестве итераций 500, что предотвращало переобучение и обеспечивало оптимальную сходимость алгоритма.

2. Результаты

Обученная модель MLP продемонстрировала высокую эффективность. Общая точность классификации на тестовой выборке составила 0.81, а средняя F1-мера — 0.82, что указывает на сбалансированность модели между точностью и полнотой предсказаний.

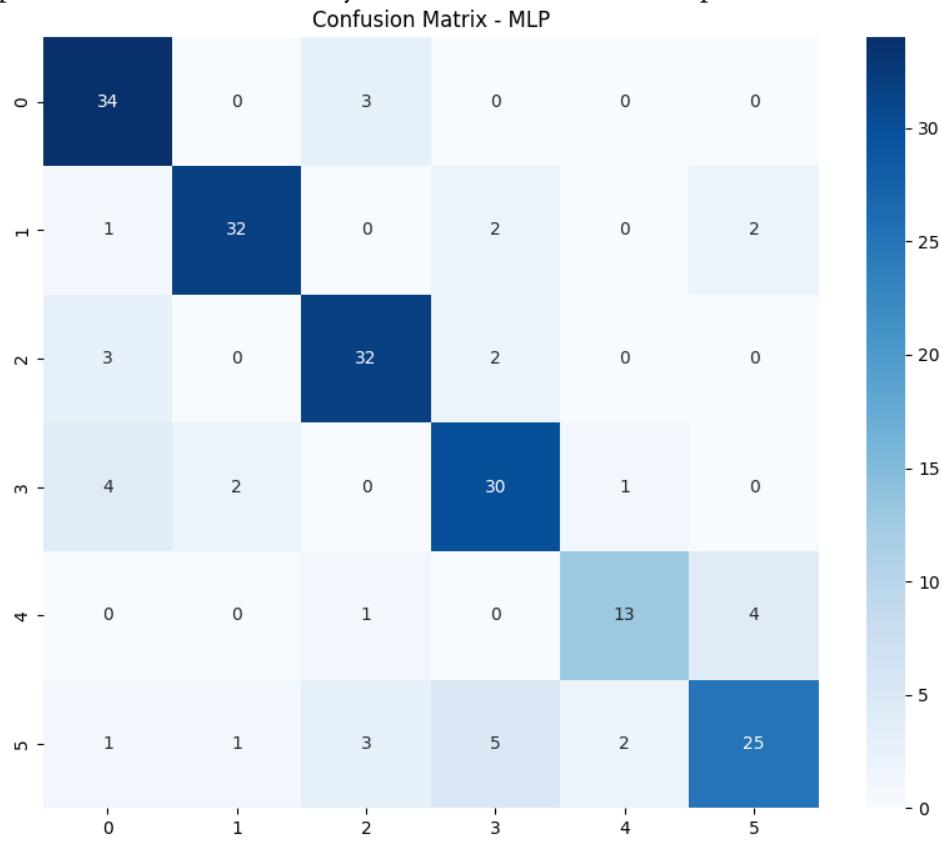


Рис. 1. Матрица ошибок для MLP

Анализ матрицы ошибок (рис. 1) позволяет детально изучить характер распознавания каждой эмоции. Наименьшее количество ошибок наблюдается для эмоций «спокойствие» (calm) и «злость» (angry). Модель надежно отделяет «нейтральное» состояние от остальных. Наибольшая путаница возникает между эмоциями «грусть» (sad), «радость» (happy) и «страх» (fearful), что согласуется с данными других исследований и может быть объяснено акустической схожестью их проявления в речи.

Таблица 1

Метрики качества распознавание по классам

Эмоция	Точность (Precision)	Полнота (Recall)	F1-score	AUC
angry	0.79	0.92	0.85	0.96
calm	0.91	0.86	0.89	0.97
fearful	0.82	0.86	0.84	0.96
happy	0.77	0.81	0.79	0.96
neutral	0.81	0.72	0.76	0.99
sad	0.81	0.68	0.74	0.93

Данные табл. 1 подтверждают высокое качество модели. Эмоция «спокойствие» распознается с максимальной точностью (91%), а «злость» — с максимальной полнотой (92 %). Почти

идеальное разделение, характеризуемое $AUC = 0.99$, достигнуто для «нейтрального» состояния. Основной сложностью для модели является распознавание «грусти», которая в 32 % случаев не обнаруживается ($recall = 68\%$), несмотря на высокую точность ее предсказания.

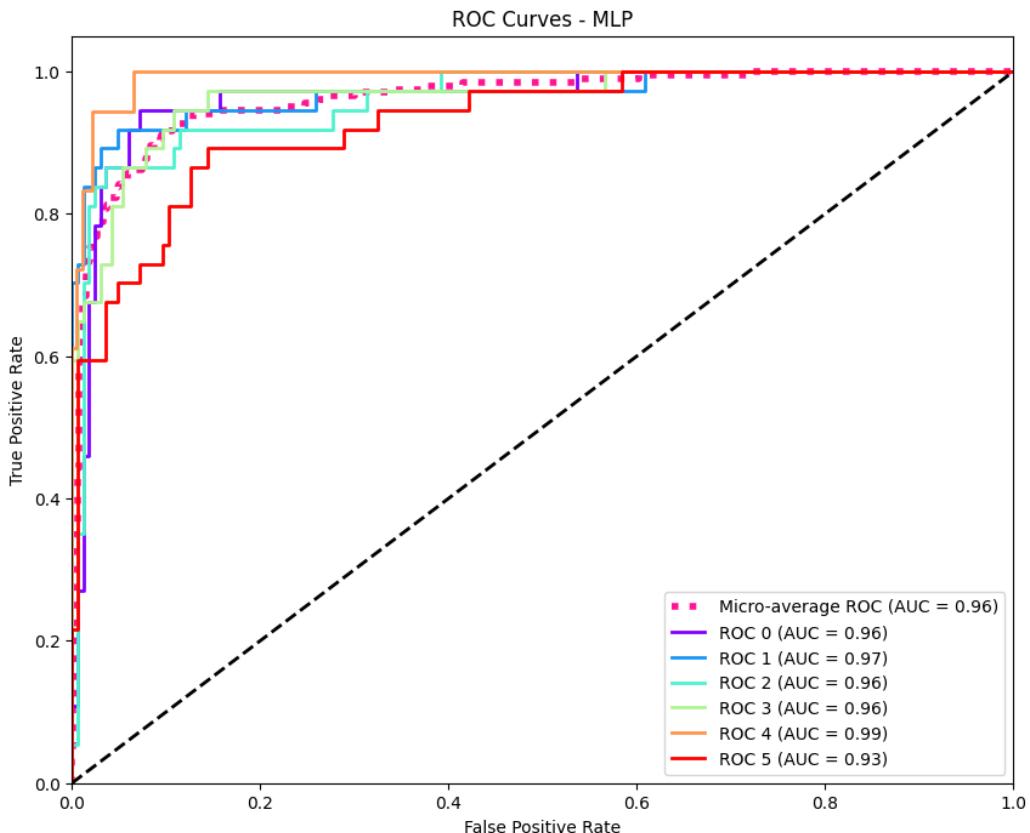


Рис. 2. ROC-кривая для MLP

Визуализация ROC-кривых (рис. 2) дополнительно подтверждает высокую разделительную способность модели. Значение Micro-AUC = 0.96 является отличным показателем. Все кривые расположены близко к левому верхнему углу графика, что свидетельствует о высоких значениях True Positive Rate при низких значениях False Positive Rate для всех шести эмоций.

В ходе исследования методом случайного поиска гиперпараметров была определена оптимальная архитектура многослойного перцептрона для решения задачи классификации эмоций по речевому сигналу. Наилучшие результаты продемонстрировала двухслойная конфигурация нейронной сети, содержащая 128 нейронов в первом скрытом слое и 64 нейрона во втором. Значимым результатом стало определение гиперболического тангенса в качестве наиболее эффективной функции активации.

Выбранная архитектура обеспечила оптимальный баланс между емкостью модели и способностью к обобщению, что подтвердились высокими метриками качества классификации. Достигнутая точность 81,77 % и F1-мера 81,61 % свидетельствуют о том, что двухслойная конфигурация обладает достаточной глубиной для выявления сложных нелинейных зависимостей в данных, одновременно минимизируя риск переобучения.

Заключение

В ходе исследования была подтверждена высокая эффективность многослойного перцептрона для задачи классификации эмоций по речи. Предложенный подход к формированию признакового пространства, основанный на статистической агрегации спектральных и

энергетических характеристик сигнала, в сочетании с моделью MLP позволил достичь общей точности 81.8 %.

Полученные результаты указывают на перспективность использования MLP в подобных задачах. В качестве направлений для будущих исследований можно выделить:

1. Оптимизацию распознавания эмоции «грусть», возможно, за счет введения дополнительных признаков, характеризующих мелодический контур или использование методов аугментации данных для балансировки классов.

2. Эксперименты с архитектурой нейронной сети, включая подбор количества слоев и нейронов, а также тестирование других типов сетей (например, сверточных для анализа спектрограмм).

3. Проверку устойчивости модели на данных от разных дикторов и в условиях реальных шумов.

Полученные результаты также демонстрируют эффективность предложенного пайплайна предобработки данных, включающего стандартизацию и снижение размерности, для задач классификации эмоций по речи. Оптимизированная архитектура MLP показала сопоставимое качество с другими современными алгоритмами машинного обучения, подтверждая целесообразность использования многослойных перцептронов в задачах обработки аудиосигналов.

Разработанная модель представляет собой надежное решение для автоматического распознавания эмоций и может быть интегрирована в различные прикладные системы.

Литература

1. Lyons J. Python Speech Features Library. Извлечение аудио-признаков (MFCC, фильтрованные банки) Python Speech Features. GitHub Repository. URL: https://github.com/jameslyons/python_speech_features. Дата обращения: 25.06.2025
2. McFee B. [et al.] LibROSA Documentation. Обработка аудиосигналов и извлечение признаков. librosa: Audio and Music Signal Analysis in Python. URL: <https://librosa.org/doc/latest/index.html>. Дата обращения: 25.06.2025
3. Pedregosa F. Scikit-learn User Guide. Реализация ML-алгоритмов и метрик оценки. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. URL: https://scikit-learn.org/stable/user_guide.html. Дата обращения: 25.06.2025
4. McKinney W. Pandas Data Processing. Работа с табличными данными. Data Structures for Statistical Computing in Python. URL: <https://pandas.pydata.org/docs/>. Дата обращения: 25.06.2025
5. El Ayadi M. [et al.] Audio Emotion Recognition Research. Теоретические основы распознавания эмоций Survey on speech emotion recognition // IEEE Transactions on Affective Computing. – 2011. – 2(2). – P. 43–55.
6. Livingstone S. R., Russo F. A. RAVDESS Dataset. Используемый датасет эмоциональной речи (2018). The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). PLoS ONE 13(5): e0196391. URL: <https://zenodo.org/record/1188976>. Дата обращения: 25.06.2025
7. Bergstra J., Bengio Y. Hyperparameter Tuning Methods. Оптимизация параметров моделей. Random Search for Hyper-Parameter Optimization // Journal of Machine Learning Research. – 2012. – 13(2). – P. 281–305.

СИНТЕЗ РЕЧИ ДЛЯ ГОЛОСОВЫХ ЧАТ-БОТОВ: СОВРЕМЕННЫЕ ПОДХОДЫ

Д. В. Котов, И. Ф. Астахова

Воронежский государственный университет

Аннотация. В статье рассматриваются современные методы синтеза речи (TTS — Text-to-Speech) для голосовых чат-ботов. Обсуждаются классические подходы, нейросетевые технологии и их применение в системах автоматизации обработки пользовательских запросов. Приводится сравнительный анализ методов с точки зрения естественности, разборчивости и вычислительных ресурсов. Сформулированы выводы о текущем состоянии TTS-технологий и перспективных направлениях их развития.

Ключевые слова: синтез речи, TTS, голосовые чат-боты, нейросетевые модели, WaveNet, Tacotron 2, FastSpeech, распознавание речи.

Введение

Голосовые чат-боты становятся важным инструментом автоматизации обслуживания пользователей в различных сферах — от колл-центров до банковской и сервисной поддержки. Одним из ключевых компонентов таких систем является синтез речи (Text-to-Speech, TTS), позволяющий преобразовать текстовые ответы бота в понятные человеку голосовые сообщения.

Качество синтеза речи напрямую влияет на пользовательский опыт: естественная интонация и разборчивость заметно повышают уровень доверия к системе. Цель данной статьи — рассмотреть современные подходы к синтезу речи, их преимущества и ограничения, а также практическое применение в голосовых чат-ботах.

1. Основы синтеза речи

Синтез речи представляет собой процесс преобразования текста в аудиосигнал. К ключевым характеристикам синтезируемой речи относятся:

- естественность — степень похожести на человеческий голос;
- разборчивость — лёгкость восприятия и понимания;
- эмоциональность и интонация — способность передавать выразительность.

Ранние TTS-системы базировались на простых алгоритмах, тогда как современные решения используют сложные нейросетевые модели, обеспечивающие существенно более реалистичное звучание.

2. Классические методы синтеза речи

До появления нейросетевых моделей синтез речи реализовывался в основном классическими методами. Они обеспечивали функциональность, достаточную для базовых задач, но существенно ограничивали естественность и гибкость звучания.

2.1. Формантный синтез

Формантный синтез [1] основан на моделировании акустических характеристик голосового тракта человека. Речь формируется с помощью математических моделей резонансов (формант) и источников звука.

Преимущества:

- полная управляемость параметрами (высота тона, длительность, тембр);
- независимость от записанных данных;
- низкие требования к памяти и вычислительным ресурсам.

Недостатки:

- характерное «роботизированное» звучание;
- ограниченная передача интонации и эмоций;
- необходимость тонкой настройки параметров для достижения приемлемого качества.

Такие системы использовались в ранних навигаторах, голосовых оповещениях и первых автоматизированных колл-центрах.

2.2. Конкатенативный синтез

Конкатенативный синтез [1] основан на объединении заранее записанных фрагментов речи (фонем, слогов, слов или коротких фраз) в новые предложения. Этот метод позволяет получить более естественный голос, поскольку речь формируется из реальных образцов.

Преимущества:

- более естественное звучание по сравнению с формантным синтезом;
- частичное сохранение интонации реального голоса;
- простота реализации при наличии большой базы голосовых данных.

Недостатки:

- ограниченная гибкость: невозможно легко изменить интонацию или эмоциональный оттенок;
- необходимость значительного объема записанных голосовых фрагментов;
- проблемы с согласованием фрагментов: переходы могут звучать неестественно, если не использованы алгоритмы сглаживания.

Метод широко применялся в голосовых интерфейсах телефонии 1990–2000-х годов, навигаторах и образовательных устройствах.

3. Современные нейросетевые подходы

В последние годы синтез речи существенно изменился благодаря развитию нейросетевых технологий. Современные подходы позволяют не только генерировать речь с высокой естественностью, но и управлять интонацией, скоростью произношения и даже эмоциональной окраской голоса. Ниже рассматриваются наиболее известные модели, которые активно применяются в современных голосовых чат-ботах, их особенности, достоинства и недостатки.

3.1. WaveNet

WaveNet [2] была разработана компанией Google в 2016 году и стала одной из первых нейросетевых моделей, способных генерировать высококачественную речь. Модель оперирует на уровне отдельных звуковых сэмплов, последовательно предсказывая каждый следующий сэмпл на основе предыдущего контекста. Такой принцип обеспечивает точную аппроксимацию речевого сигнала и позволяет получать акустический результат, близкий к естественному звучанию.

Основным ограничением WaveNet является высокая вычислительная сложность: генерация аудиосигнала требует значительных вычислительных ресурсов, а синтез в реальном времени возможен только при использовании специализированного или высокопроизводительного оборудования.

Для задач, связанных с интерактивными голосовыми интерфейсами и чат-ботами, преимущественно применяются более оптимизированные модели, обеспечивающие приемлемый баланс между качеством и скоростью генерации.

3.2. Tacotron 2

Tacotron 2 [2] представляет собой двухэтапную архитектуру синтеза речи. На первом этапе модель преобразует входной текст в мел-спектрограмму, на втором — соответствующий аудиосигнал формируется при помощи вокодера, в качестве которого могут использоваться различные решения, включая WaveNet. Такая схема обеспечивает корректное моделирование интонационных характеристик и позволяет гибко управлять параметрами результирующего сигнала.

Tacotron 2 применяется в системах озвучивания текста и голосовых интерфейсах благодаря сочетанию высокого качества синтеза и умеренных требований к вычислительным ресурсам. К ограничениям Tacotron 2 относится сравнительно невысокая скорость генерации, что необходимо учитывать при использовании в интерактивных системах, работающих в режиме реального времени.

3.3. FastSpeech

FastSpeech [3] разработан как более производительная и стабильная альтернатива модели Tacotron 2. Ключевая особенность архитектуры заключается в параллельной генерации спектрограммы, что обеспечивает значительное увеличение скорости синтеза по сравнению с автоагрегатными моделями. Такой подход позволяет получать аудиосигнал с сохранением основных интонационных и просодических характеристик при существенно меньших задержках.

Модель демонстрирует высокую пригодность для приложений, работающих в режиме реального времени, включая интерактивные чат-боты и мобильные решения. В то же время обучение FastSpeech требует крупного и качественно размеченного набора данных, а при обработке редких или нестандартных слов могут возникать артефакты произношения.

Благодаря оптимальному соотношению скорости генерации и качества синтеза FastSpeech широко используется в современных системах TTS, ориентированных на практические сценарии с высокими требованиями к производительности.

3.4. Интеграция распознавания речи и TTS (Whisper + TTS)

Одним из направлений развития голосовых систем является интеграция моделей автоматического распознавания речи (ASR), таких как Whisper [4] от OpenAI, с системами синтеза речи (TTS). В такой архитектуре входной речевой сигнал пользователя преобразуется в текст, который затем обрабатывается модулем обработки естественного языка (NLP) и передаётся в систему синтеза для формирования аудиовыхода.

Whisper обеспечивает устойчивое распознавание речи в условиях различного уровня шумов и изменяющихся характеристик аудиосигнала. Совмещение ASR, NLP и TTS позволяет создавать архитектуры голосовых интерфейсов, функционирующих в режиме диалоговой системы и поддерживающих обмен репликами в реальном времени.

Основной сложностью интеграции является необходимость одновременной работы нескольких моделей, что увеличивает вычислительные требования и требует оптимизации архитектуры или применения специализированного аппаратного обеспечения.

3.5. Сравнительный анализ моделей

Для более наглядного понимания различий между рассмотренными моделями целесообразно привести их сравнительный анализ по ключевым параметрам: естественность речи, скорость генерации, вычислительные ресурсы и области применения. Основные характеристики WaveNet, Tacotron 2, FastSpeech и интеграции Whisper + TTS приведены в табл. 1.

Таблица 1

Сравнение современных моделей синтеза речи

Модель	Естественность	Скорость генерации	Вычислительные ресурсы	Применение
WaveNet	Очень высокая	Низкая	Высокие	Высококачественные ассистенты, аудиокниги
Tacotron 2	Высокая	Средняя	Средние	Озвучка текста, голосовые ассистенты
FastSpeech	Высокая	Высокая	Средние	Реальное время, интерактивные чат-боты
Whisper+TTS	Средняя/Высокая	Высокая	Средние/Высокие	Диалоговые системы, колл-центры

Из сравнения видно, что выбор модели TTS зависит от конкретных задач. WaveNet обеспечивает наивысшее качество речи, но медленный и ресурсоёмкий. Tacotron 2 сочетает естественность с умеренной скоростью, подходя для голосовых ассистентов. FastSpeech оптимизирован для работы в реальном времени и интерактивных чат-ботов. Интеграция Whisper с TTS обеспечивает точное понимание речи и динамическое построение диалога, что делает её перспективной для современных голосовых систем.

4. Ограничения и перспективы применения TTS-технологий

Современные голосовые чат-боты широко используются для автоматизации обслуживания в колл-центрах, банках, сервисных службах и устройствах «умного дома», помогая обрабатывать стандартные запросы, снижать нагрузку на операторов и сокращать время ожидания клиентов. При этом существуют определённые ограничения при использовании TTS-систем. Передача эмоций и интонации пока остаётся ограниченной, особенно при синтезе сложных или нестандартных фраз. Кроме того, редкие слова, имена и специфические термины могут произноситься некорректно, а крупные нейросетевые модели требуют значительных вычислительных ресурсов, что затрудняет их внедрение на маломощных устройствах. Эффективное обучение таких моделей, как FastSpeech или Tacotron 2, также требует больших и качественно размеченных датасетов.

Несмотря на эти ограничения, перспективы применения TTS-технологий остаются очень широкими. Интеграция с системами распознавания речи и NLP позволяет создавать интерактивные диалоговые системы, развитие эмоциональной и адаптивной речи повышает естественность общения, а мульти-языковые модели и оптимизация под мобильные устройства расширяют возможности глобального применения голосовых ассистентов. Таким образом, современные технологии синтеза речи открывают новые горизонты для эффективного, удобного и максимально приближенного к человеческому взаимодействия с пользователями.

Заключение

Синтез речи является ключевым компонентом голосовых чат-ботов. Современные нейросетевые модели обеспечивают естественную и разборчивую речь, значительно расширяя возможности голосовых систем. Перспективные направления развития, включая эмоциональную и адаптивную речь, мультиязыковые модели и оптимизацию под мобильные устройства, создают дополнительные возможности для интерактивных голосовых ассистентов. Использование таких технологий позволяет создавать голосовых ассистентов, чья речь максимально приближена к естественному человеческому общению, что повышает качество взаимодействия и эффективность автоматизированных систем.

Литература

1. Рыбин С. В. Синтез речи: учебное пособие / С. В. Рыбин. – Санкт-Петербург : Университет ИТМО, 2014. – 92 с.
2. Пантиухин Д. В. Нейронные сети синтеза речи голосовых помощников и поющих автоматов // Речевые технологии / Speech Technologies. – Москва : Издательский дом «Народное образование», 2021. – № 3–4. – С. 3–16.
3. Ren Y., Ruan Y., Tan X., Qin T., Zhao S., Zhao Z., Liu T. Y. FastSpeech: Fast, Robust and Controllable Text-to-Speech. – URL: <https://huggingface.co/papers/1905.09263> (дата обращения: 12.11.2025).
4. OpenAI. Whisper: система автоматического распознавания речи. – URL: <https://openai.com/index/whisper/> (дата обращения: 17.11.2025).

МЕТОДЫ КОНТРОЛЯ ГЕНЕРАЦИИ ИЗОБРАЖЕНИЙ В ДИФФУЗИОННЫХ МОДЕЛЯХ

А. О. Левин, Ю. Е. Гагарин

Калужский филиал Московского государственного технического университета
имени Н. Э. Баумана

Аннотация. Диффузионные модели продемонстрировали беспрецедентные возможности в синтезе фотoreалистичных изображений, вызвав трансформации во многих креативных и научных областях. Однако фундаментальной проблемой остается переход от стохастической генерации к контролируемому синтезу. Неконтролируемый или слабо контролируемый синтез, часто основанный исключительно на текстовых описаниях, не способен удовлетворить требования прикладных задач, требующих точного соблюдения композиции, геометрии, ракурса и консистентности объектов. В данной статье представлен систематический обзор и анализ современных методов контроля генерации изображений в диффузионных моделях. Особое внимание уделяется их применимости для специфической задачи генерации двумерных изображений объектов в ортографической проекции.

Ключевые слова: диффузионные модели, генерация изображений, контролируемый синтез, ControlNet, ортографическая проекция, T2I-адаптеры, мультимодальный контроль, text-to-image, условная генерация.

Введение

Диффузионные модели продемонстрировали высокую эффективность в задачах генерации фотoreалистичных изображений и в настоящее время широко используются в области генеративной графики [1]. Их применение выходит за рамки художественных экспериментов и включает задачи дизайна, технической визуализации, медицины, робототехники и аналитики данных.

Вместе с тем ключевой исследовательской проблемой остается переход от малоуправляемой генерации изображений к контролируемой. Генерация, опирающаяся преимущественно на текстовые описания, во многих случаях не обеспечивает соблюдения строгих прикладных требований: точной композиции, согласованной геометрии, заданного ракурса и консистентности представления объекта в серии изображений [2].

Цель данной работы состоит в анализе применимости современных методов контроля генерации изображений в диффузионных моделях, в том числе по отношению к задаче генерации двумерных изображений в ортографической проекции.

1. Базовые принципы диффузионных моделей и условной генерации

Диффузионные модели рассматривают генерацию изображения как обратный процесс по отношению к его «зашумлению». На прямом этапе к исходному изображению шаг за шагом добавляют гауссовский шум, пока оно полностью не разрушается; на обратном этапе нейросетевая модель постепенно убирает шум и восстанавливает осмысленное изображение [3].

В современных реализациях этот процесс выполняется не в исходном пиксельном представлении, а в более компактном внутреннем представлении изображения, что снижает вычислительные затраты при сопоставимом качестве результата [4]. В условной генерации дополнительно вводится сигнал-условие. В задачах типа «текст–изображение» текстовый запрос кодируется языковым энкодером, а полученный вектор признаков используется в архитектуре U-Net через механизм cross-attention.

Механизм classifier-free guidance (CFG) позволяет регулировать степень влияния текста за счет комбинирования условного и безусловного предсказаний шума. Вместе с тем один только текстовый сигнал не обеспечивает точного контроля геометрии и ракурса. Текстовое описание неоднозначно, а многие пространственные отношения трудно задать словами. Поэтому текст обычно дополняют другими типами управляющих сигналов.

2. Типы управляющих сигналов в диффузионных моделях

Методы контроля генерации в диффузионных моделях удобно классифицировать по типу вводимого управляющего сигнала. Рассмотрим четыре основных класса: текстовый контроль, структурный контроль на основе карт, визуальный контроль через примеры изображений и высокоуровневый (символический) контроль (рис 1). Отдельно выделяется группа геометрических и ракурс-осознанных методов, основанных на многовидовых и трехмерных представлениях.

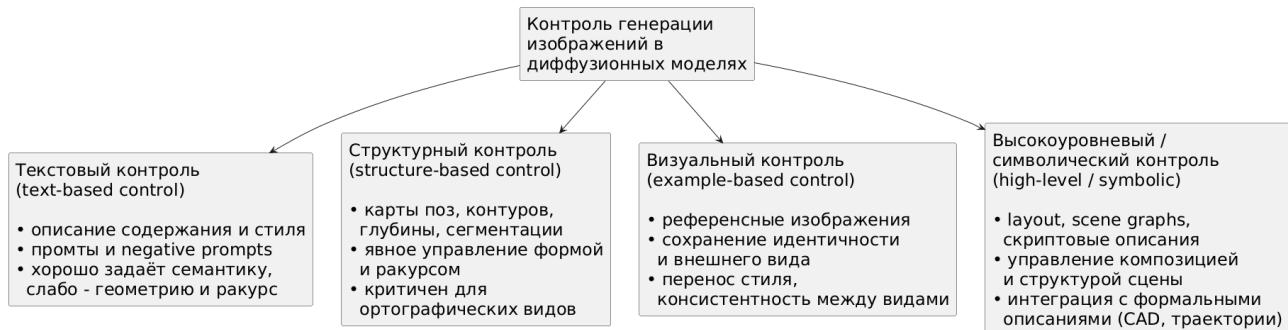


Рис. 1. Основные типы управляющих сигналов в диффузионных моделях

2.1. Текстовый контроль

Текстовый контроль опирается на использование текстовых промтов, которые остаются самым распространенным механизмом управления. Пользователь формулирует описание содержания и стиля изображения, а также, при необходимости, добавляет негативные промты для исключения нежелательных элементов. Преимуществами такого подхода являются естественность взаимодействия и низкий порог входа, а также высокая выразительность текста для задания семантики сцены и стилистики изображения. Текстовые описания также хорошо сочетаются с современными языковыми моделями.

В то же время интерпретация промтов неизбежно остается неоднозначной: близкие по смыслу формулировки могут приводить к различным результатам, а небольшие изменения формулировки или случайного seed заметно влияют на структуру изображения. Особенно чувствительными оказываются задачи, требующие строгого контроля геометрии и ракурса (в том числе ортографических видов), где текстовое описание не обеспечивает достаточной предсказуемости и воспроизведимости [5]. В этом контексте текст целесообразно рассматривать в первую очередь как средство задания содержания и стиля, а не как основной инструмент геометрического управления.

2.2. Структурный контроль

Структурный контроль основан на введении дополнительных карт, описывающих геометрию и композицию сцены. В качестве таких карт могут использоваться скелеты поз, контуры и карты границ, карты глубины и нормалей, а также семантические маски и результаты сегментации. Модели семейств ControlNet и T2I-Adapter интерпретируют эти структуры как дополнительные каналы, связанные с основным U-Net через специализированные адаптеры [6].

Подобная схема позволяет явно управлять композицией и формой: при наличии корректных структурных карт модель надежно следует заданному силуэту и ракурсу объекта, при этом текст играет вспомогательную роль, уточняя содержание и стиль. Вместе с тем структурный контроль требует либо специальной разметки, либо использования внешних алгоритмов для извлечения карт, что усложняет процесс работы. Добавление соответствующих модулей увеличивает архитектурную и вычислительную сложность системы, а ошибки в картах могут приводить к «жесткому» навязыванию некорректных структур. Для задач ортографической проекции именно такой тип управления оказывается принципиально важным: глубинные и карты нормали, а также контуры позволяют явно задавать проекцию объекта на выбранную плоскость и фиксировать ракурс.

2.3. Структурный контроль

Визуальный контроль реализуется за счет использования референсных изображений, задающих желаемый стиль, текстуру или идентичность объекта. Специализированные адаптеры позволяют разделять внимание между текстовыми и визуальными признаками, что обеспечивает совместный учет семантики текстового описания и внешнего вида примера.

Этот класс методов применяется, прежде всего, для сохранения идентичности объекта в серии изображений, переноса художественного стиля и обеспечения частичной консистентности между различными ракурсами. В условиях ортографической генерации референсное изображение может выступать в роли эталонного вида, по отношению к которому требуются согласованные ортографические проекции: модель должна воспроизводить одинаковые визуальные характеристики при изменении направления проекции.

2.4. Высокоуровневый контроль

Высокоуровневый, или символический, контроль задает сцену через формальные описания. В качестве таких описаний могут выступать layout-репрезентации с явным указанием положения и размера объектов, сцены в виде графов объектов и их отношений, а также сценарные или скриптовые спецификации. Методы класса layout-to-image и grounded diffusion позволяют связать текстовые описания с конкретными областями изображения, что повышает точность расположения объектов и облегчает контроль композиции [7].

Эти подходы особенно перспективны для технической и научной визуализации, где исходные данные часто уже представлены в формализованном виде — в формате CAD-моделей, траекторий робототехнических систем, медицинских информационных систем и т.п. Интеграция таких структурированных описаний с диффузионными моделями открывает возможности для более строгого и проверяемого контроля генерации.

3. Геометрический-осознанный контроль

Отдельную группу составляют геометрически-осознанные и ракурс-осознанные методы, использующие многовидовые или трехмерные представления для обеспечения согласованности изображений одного объекта под различными ракурсами. Модели класса multi-view diffusion генерируют несколько видов сцены одновременно, синхронизируя внутренние представления для различных направлений обзора и тем самым добиваясь согласованности геометрии и цветовых характеристик между ракурсами [8].

В качестве альтернативного решения рассматриваются методы, в которых предварительно формируется скрытое трёхмерное представление сцены (например, в виде нейросетевого поля

излучения NeRF), после чего из этого представления вычисляются двумерные проекции для заданных параметров виртуальной камеры.

Для задач ортографической проекции такие методы имеют принципиальное значение. Они позволяют рассматривать набор проекций не как совокупность независимых изображений, а как реализацию единого совместного распределения видов объекта. Это облегчает соблюдение пропорций и форм при переходе от одного ракурса к другому и дает возможность задавать согласованный набор ортографических представлений, ориентированных на разные плоскости проекции.

В то же время широкое применение геометрически-осознанных подходов ограничивается рядом факторов. Во-первых, многовидовая генерация существенно увеличивает вычислительную стоимость по сравнению с традиционными одновидовыми моделями. Во-вторых, обучение таких систем требует многовидовых датасетов или трехмерных моделей с известными параметрами обзора, которые доступны далеко не для всех предметных областей. В-третьих, интеграция трехмерно-ориентированных блоков с текстовыми, структурными и визуальными условиями в рамках единой архитектуры представляет собой нетривиальную исследовательскую задачу и пока не имеет устоявшихся решений.

4. Генерация ортографических представлений

Под генерацией ортографических представлений в текущем контексте понимается получение набора изображений объекта в стандартных ракурсах (вид спереди, сбоку, сверху и т. д.) без перспективных искажений. Подобные наборы широко используются в инженерной графике, робототехнике, технических каталогах, учебных материалах и других областях, где требуется строгое соблюдение масштабов и форм (рис. 2).

К ортографическим представлениям предъявляется ряд специфических требований. Во-первых, необходима геометрическая согласованность между ракурсами: пропорции и контуры объекта должны согласованно воспроизводиться при смене направления проекции. Во-вторых, важна стабильность визуальных атрибутов, включая цвет, текстуру и маркировку, особенно если изображения рассматриваются как элементы единого описания объекта. В-третьих, требуется строгое соблюдение параметров проекции: отсутствие перспективных

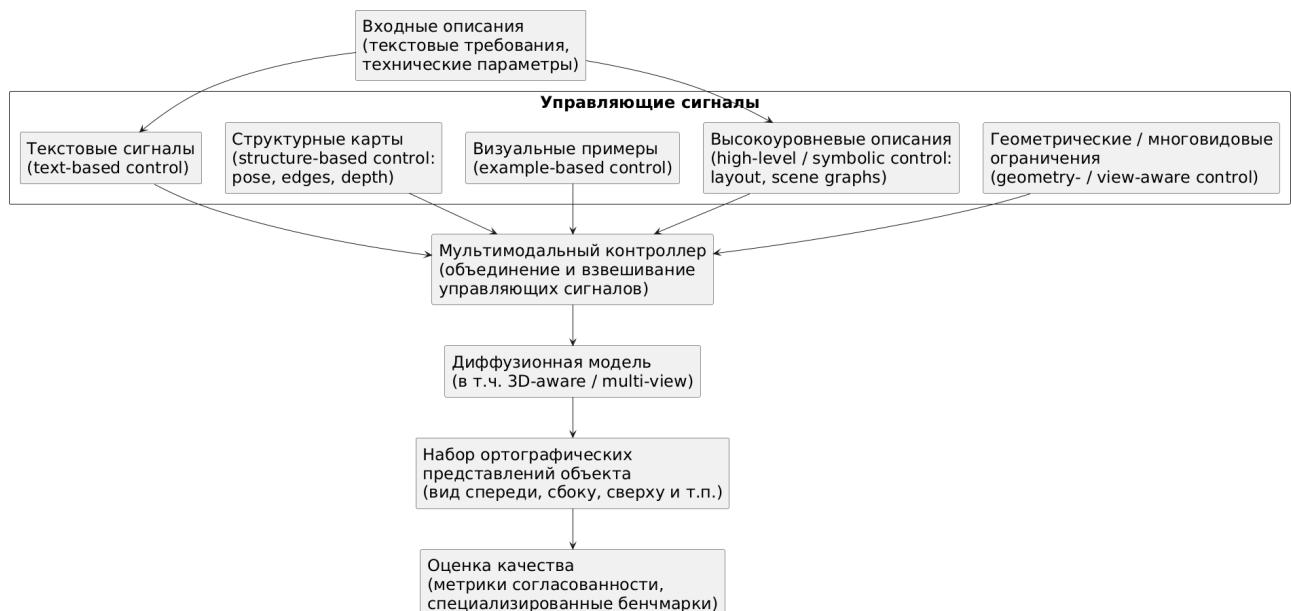


Рис. 2. Схема конвейера контролируемой генерации ортографических представлений на основе диффузионной модели

искажений и корректное отображение масштабов в соответствии с выбранной системой координат [9].

Рассмотренные выше типы управляющих сигналов вносят разные элементы в решение этой задачи. Текстовые описания задают семантическую сущность объекта и общий стиль изображения, но практически не обеспечивают точного контроля геометрии. Структурные карты, такие как карты глубины, нормалей и контуры, позволяют явно задать форму и ракурс, включая ортографическую проекцию на выбранную плоскость. Визуальные примеры служат для сохранения идентичности внешнего вида при переходе от одного вида к другому, что важно при построении наборов согласованных проекций. Геометрически-осознанные многовидовые модели, в свою очередь, обеспечивают согласованность набора проекций как целостного представления объекта, а не набора независимых изображений.

Несмотря на наличие таких «строительных блоков», специализированные решения, нацеленные именно на генерацию ортографических представлений, пока остаются ограниченными. На сегодняшний день отсутствуют общепринятые датасеты с ортографическими проекциями и детальной аннотацией ракурсов, равно как и метрики, количественно оценивающие межракурсную согласованность и точность соблюдения проекционных свойств. Это существенно затрудняет систематическое сравнение методов и обучение моделей, ориентированных на подобные задачи, и подчеркивает необходимость дальнейших исследований в данном направлении.

5. Ограничения существующих подходов и перспективы развития

Обобщение рассмотренных методов позволяет выделить несколько ключевых ограничений, характерных для текущего состояния области. Одной из центральных проблем является конфликт управляющих сигналов: одновременное использование текста, структурных карт, визуальных примеров и многовидовых условий нередко приводит к противоречивым требованиям к результирующему изображению. На практике балансировка влияния различных сигналов осуществляется эмпирически, посредством подбора коэффициентов и параметров обучения, и пока не существует строго обоснованных методик, гарантирующих согласованное взаимодействие всех источников информации.

Дополнительную сложность создает рост архитектурной и вычислительной нагрузки. Наращивание числа адаптеров, модулей структурного контроля и 3D-aware блоков приводит к увеличению размеров моделей и затрат на их использование. Это ограничивает применение таких систем в условиях ограниченных ресурсов и в сценариях, где требуется близкая к реальному времени генерация изображений.

Существенным ограничением является также отсутствие специализированных бенчмарков и метрик для задач ортографической и многовидовой двумерной визуализации. Имеющиеся показатели, такие как FID или CLIP-score, ориентированы преимущественно на оценку перспективного качества и семантического соответствия тексту и не отражают геометрическую согласованность и точность проекций. Отсутствие стандартных наборов данных с ортографическими представлениями и четко определенными геометрическими характеристиками затрудняет объективное сравнение методов и замедляет развитие направления.

Наконец, важной практической проблемой остаются устойчивость и воспроизводимость результатов. Случайный характер генерации в диффузионных моделях осложняет получение детерминированных изображений при повторных запусках даже при небольших изменениях условий. Для ответственных приложений, таких как медицинская визуализация или инженерные отчеты, это особенно критично и требует разработки дополнительных механизмов контроля и механизмов оценки надежности результата.

Перспективные направления исследований включают создание мультимодальных контроллеров, способных явно управлять приоритетами разных сигналов и формализованно разрешать их конфликты; разработку специализированных бенчмарков и метрик для оценки ортографической и многовидовой согласованности; а также изучение интерактивных систем типа «человек + ИИ», в которых пользователь поэтапно уточняет структурные и геометрические ограничения, а модель, в свою очередь, обеспечивает сохранение уже достигнутой согласованности при внесении изменений.

Заключение

Таким образом, в данной работе был представлен анализ методов контроля генерации изображений в диффузионных моделях с классификацией по типам управляющих сигналов и акцентом на их применимость к задачам генерации двумерных изображений объектов в ортографической проекции. Показано, что текстовые сигналы эффективно задают семантику и стиль, но слабо контролируют геометрию; структурные карты и визуальные примеры позволяют управлять формой и идентичностью объекта, тогда как геометрически-осознанные многовидовые модели обеспечивают согласованность наборов проекций.

Выделенные ограничения — конфликт условий, рост архитектурной сложности, отсутствие специализированных бенчмарков и метрик, а также проблемы устойчивости и воспроизведимости — определяют ключевые направления дальнейших исследований. Дальнейшее развитие области видится в создании мультимодальных контроллеров, интегрирующих различные типы управляющих сигналов, в формализации критериев геометрической согласованности и в разработке ориентированных на ортографические представления наборов данных, которые позволяют систематически оценивать и сравнивать новые методы контролируемой генерации.

Литература

1. Cao P. [et al.] Controllable Generation with Text-to-Image Diffusion Models: A Survey // arXiv preprint. – 2024.
2. Po R. [et al.] State of the Art on Diffusion Models for Visual Computing // Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV). – 2023.
3. Ho J., Jain A., Abbeel P. Denoising Diffusion Probabilistic Models // Advances in Neural Information Processing Systems (NeurIPS). – 2020.
4. Rombach R. [et al.] High-Resolution Image Synthesis with Latent Diffusion Models // Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). – 2022.
5. Zhang L., Rao A., Agrawala M. Adding Conditional Control to Text-to-Image Diffusion Models (ControlNet) // Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV). – 2023.
6. Mou C. [et al.] T2I-Adapter: Learning Adapters to Dig out More Controllable Ability for Text-to-Image Diffusion Models // arXiv preprint. – 2023.
7. Li Y. [et al.] GLIGEN: Open-Set Grounded Text-to-Image Generation // Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). – 2023.
8. Liu Y. [et al.] SyncDreamer: Generating Multiview-Consistent Images from a Single-View Image // Int. Conf. on Learning Representations (ICLR). – 2024.
9. Tang S. [et al.] MVDiffusion: Enabling Holistic Multi-View Image Generation with Correspondence-Aware Diffusion // Advances in Neural Information Processing Systems (NeurIPS). – 2023.

ПРИМЕНЕНИЕ НЕЙРО-НЕЧЕТКИХ СИСТЕМ В ЗАДАЧАХ ЗАПОЛНЕНИЯ ПРОПУСКОВ

А. В. Лепендин, Т. В. Азарнова

Воронежский государственный университет

Аннотация. Цель исследования — оценить эффективность использования в задачах заполнения пропусков нейро-нечетких гибридных систем таких как: адаптивная нейро-нечеткой системы вывода (ANFIS) с моделью нечеткого вывода Такаги-Сугено и нечеткая кластерная импутация (Fuzzy Cluster Imputer). Оба подхода используют гауссовые функции принадлежности и инициализацию через метод ближайших соседей, параметры ANFIS настраиваются алгоритмом градиентного спуска, в Fuzzy Cluster Imputer используется фиксированная кластерная структура. Экспериментальная апробация данных двух подходов показывает, что они позволяют существенно повысить качество заполнения пропусков по сравнению со стандартными наиболее часто используемыми методами. Результаты исследования подтверждают перспективность интеграции нейро-нечётких систем в модули анализа данных.

Ключевые слова: неполные данные, нечеткие системы, Такаги — Сугено, ANFIS, нечеткая кластерная импутация, метод ближайших соседей, переобучение.

Введение

Одной из серьезных проблем при решении задачи прогнозирования остаются неполные данные. Пропущенные значения возникают по различным причинам от технического характера (при неполадках и сбоях оборудования), до связанных с человеческим фактором (отказ человека от прохождения процедуры или ответа на вопрос). Наличие пропусков в данных снижает качество их предварительной обработки, а также эффективность использования данных для дальнейшего моделирования. [7]

Настоящая статья посвящена проблеме заполнения пропущенных значений с использованием гибридных нечетких методов. В рамках исследования решается задача построения адаптивной нейро-нечеткой системы вывода (ANFIS) с моделью Такаги — Сугено первого порядка и нечеткого кластерного импутера. Проводится оценка качества заполнения пропусков и сравнение точности предлагаемых решений со стандартными методами заполнения пропусков. Поднимается проблема переобучения при малом объеме данных. [5]

Объектом исследования являются методы заполнения пропущенных значений в табличных, в первую очередь, медицинских данных. Предметом исследования являются гибридные нечеткие системы, сочетающие кластерный анализ и нечеткий вывод. Также стоит отметить, что нечеткий вывод Такаги — Сугено был выбран вместо алгоритма Мамдани по причине преимущества первого в гладкости отображения, возможности применения градиентной оптимизации, а также из-за отсутствия необходимости этапа дефазификации, что делает его предпочтительным в задаче регрессии при заполнении пропусков. [6]

В процессе исследования анализируются условия, при которых рассматриваемые подходы демонстрируют свои преимущества и недостатки, внимание сконцентрировано на балансе между выразительностью, устойчивостью и интерпретируемостью подходов.

1. Описание эксперимента

1.1. Данные для эксперимента

При выборе подходящего для исследования набора данных наличие пропусков в данных не являлось обязательным условием, поскольку обучение и тестирование проводилось на той

его части, которую составляли полные записи по всем переменным. Одной из причин для выбора медицинских наборов данных была особенность таких данных, связанная с асимметричностью связей между признаками. Например, для больных диабетом людей характерен как высокий инсулин (инсулинорезистентность при сахарном диабете второго типа), так и крайне низкий инсулин (отсутствие выработки инсулина при сахарном диабете 1-го типа). Исходя из данных особенностей медицинских данных было принято решение строить правила для заполнения пропущенных значений с учетом известных исходов (людей с диабетом или без). Полные данные использовались, как наиболее подходящие для контролируемых экспериментов и сравнительного анализа методов заполнения пропусков. [4]

1.2. Методы заполнения пропусков и оценка качества заполнения пропусков

Для восстановления пропущенных значений использовались три стандартных метода:

- заполнение средним или медианным значением;
- заполнение методом k-ближайших соседей (KNNImputer);
- итеративное заполнение регрессионной моделью (MICE).

И два гибридных метода:

- адаптивная нейро-нечеткая система вывода (ANFIS) с моделью нечеткого вывода Такаги — Сугено

- нечеткая кластерная импутация (Fuzzy Cluster Imputer)

Для каждого из них рассчитывались метрики MAE и RMSE между восстановленными и исходными значениями.

Метрика MAE (среднее абсолютное отклонение рассчитывается) по формуле (1):

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n}. \quad (1)$$

Метрика RMSE (среднеквадратическое отклонение рассчитывается) по формуле (2):

$$RMSE = \sqrt{\frac{\sum_{i=1}^n |\hat{y}_i - y_i|^2}{n}}. \quad (2)$$

Выбор гауссовой нечеткой функций принадлежности обусловлен математическими, вычислительными и интерпретационными преимуществами в контексте гибридных нейро-нечетких систем. [9]

Гауссова функция — бесконечно дифференцируема по всем параметрам, что важно для обучения ANFIS, в то время как для других функций (треугольных, трапециевидных) в точках излома градиенты не определены или обучение нестабильно из-за скачкообразности, гауссова функция представима по формуле (3):

$$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\sigma^2}\right). \quad (3)$$

1.3. Обучение нейро-нечеткой системы, генерация правил

Из 768 строк в исходных данных только 392 оказались полностью заполненными, в обучающую выборку попало 313 записей, в тестовую выборку попало 79 записей. Центры и ширины гауссовых функций принадлежности инициализировались методом ближайших соседей, а затем настраивались в процессе обучения ANFIS. Для каждой зависимой переменной строилось по 3 правила для исхода с диабетом и без диабета, итого — по 6 правил.

Пример построения правил в нейро-нечеткой системе ANFIS для пациентов без диабета (Outcome = 0) для заполнения признака «Кровяное давление» — «BloodPressure» показан ниже (рис. 1).

✖ Нечёткие правила для BloodPressure (Outcome=0):

Правило 1:

```
Pregnancies      ≈ 2.9
Glucose          — высокий (132.7)
SkinThickness    ≈ 37.2
Insulin          — высокий (210.5)
BMI              — высокий (40.0)
DiabetesPedigreeFunction ≈ 0.6
Age              ≈ 29.8
Outcome          ≈ 0.0
→ BloodPressure = -0.40 - 0.34·Pregnancies + 0.26·Glucose + 0.42·SkinThickness + 0.18·Insulin - 0.04·BMI + 0.30·DiabetesPedigreeFunction + 0.16·Age
```

Правило 2:

```
Pregnancies      ≈ 1.6
Glucose          — низкий (88.6)
SkinThickness    ≈ 25.4
Insulin          — низкий (72.7)
BMI              — низкий (26.8)
DiabetesPedigreeFunction ≈ 0.6
Age              ≈ 17.4
Outcome          ≈ 0.0
→ BloodPressure = 0.55 - 0.37·Pregnancies + 0.15·Glucose + 0.09·SkinThickness - 0.08·Insulin + 0.32·BMI - 0.28·DiabetesPedigreeFunction + 1.27·Age
```

Правило 3:

```
Pregnancies      ≈ 8.9
Glucose          — средний (121.5)
SkinThickness    ≈ 22.4
Insulin          — высокий (178.7)
BMI              — высокий (36.3)
DiabetesPedigreeFunction ≈ 0.5
Age              ≈ 46.0
Outcome          ≈ 0.0
→ BloodPressure = -0.42 + 0.38·Pregnancies + 0.19·Glucose - 0.74·SkinThickness - 0.26·Insulin + 0.67·BMI - 0.31·DiabetesPedigreeFunction + 0.19·Age
```

Рис. 1. Построение правил с помощью нечеткого вывода Такаги — Сугено первого порядка

Для большей наглядности были построены несколько функций принадлежности различных признаков, используемых в правилах для пропусков в других признаках (рис. 2, 3).

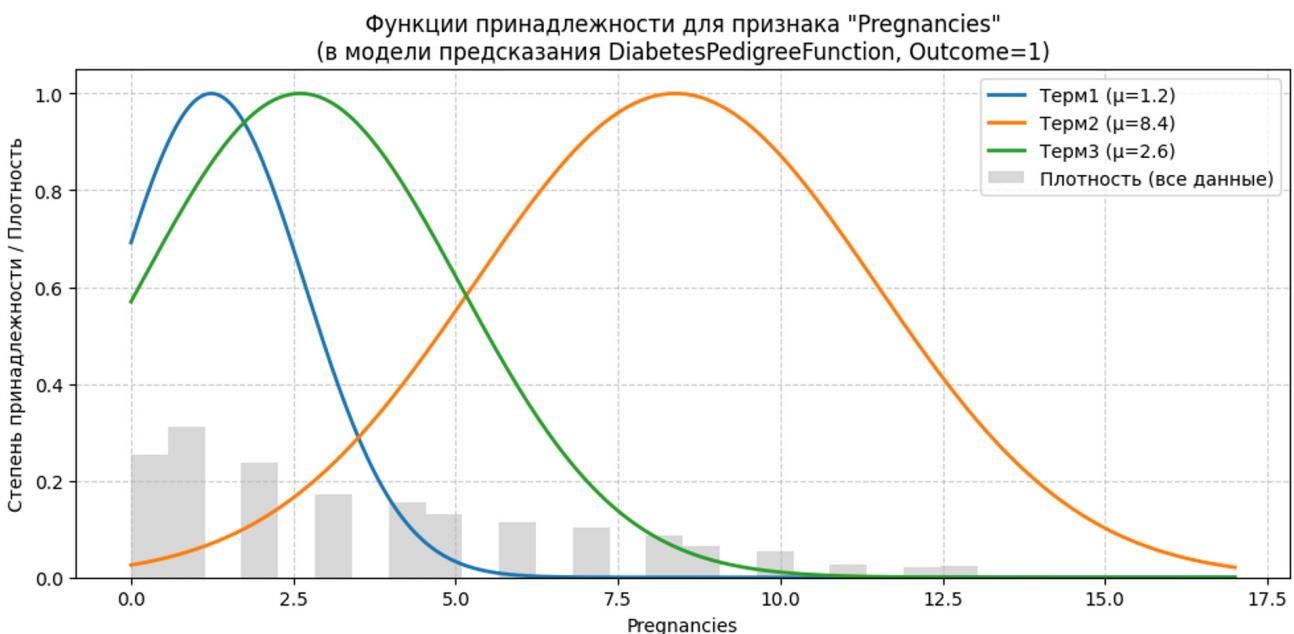


Рис. 2. Функция принадлежности для признака «Количество беременностей»
в модели предсказания «Функции родословной диабета»

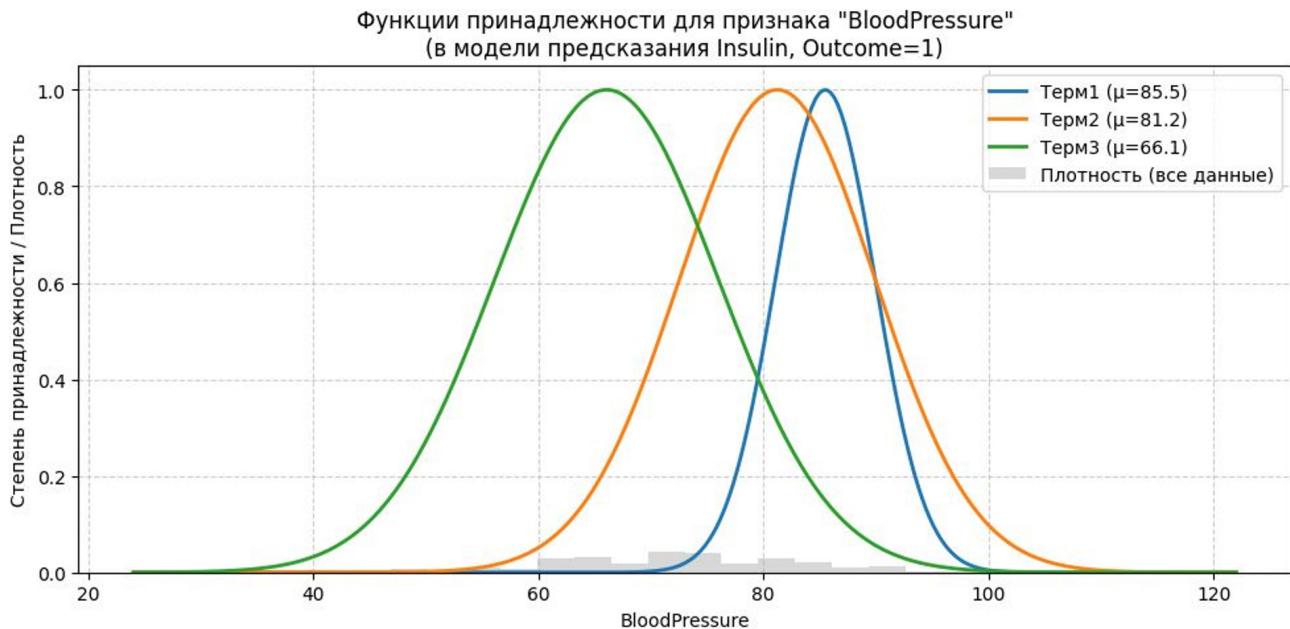


Рис. 3. Функция принадлежности для признака «Кровяное давление» в модели предсказания «Инсулина»

1.4. Нечеткое кластерное заполнение пропущенных данных

Для полностью заполненных данных была проведена кластеризация с разным количеством кластеров, целью задачи был расчет средних значений признаков в зависимости от принадлежности к кластеру для дальнейшего заполнения пропущенных данных с учетом степени принадлежности записи с пропуском одному из выявленных кластеров.

На рис. 4 приведена демонстрация кластеров в проекции на признаки «Глюкоза» и «Индекс массы тела».

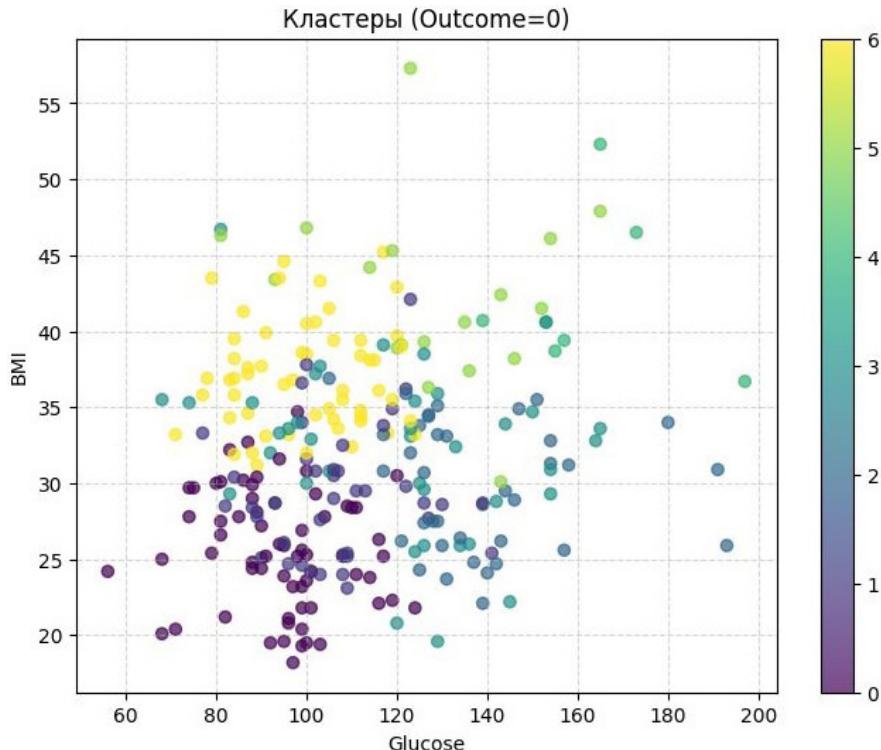


Рис. 4. Построение 7 кластеров в проекции на признаки «Глюкоза» и «Индекс массы тела»

2. Результаты

Для оценки качества заполнения пропущенных значений, в полных данных поочередно для каждого предиктора (независимой переменной) создаются пропуски, которые заполняются на основании полных данных из других переменных.

Предлагаемая нейро-нечеткая система ANFIS показывает отличные результаты, однако уже при построении 3 правил для каждого признака и исхода наблюдается повышение ошибки на тестовой выборке в сравнении с обучающей выборкой, что говорит о переобучении (рис. 5).

```
[ANFIS] Оценка для: Glucose
Train MAE: 15.14, RMSE: 19.80
Test MAE: 18.61, RMSE: 24.74
▲ Возможное переобучение (test/train MAE = 1.23)

[ANFIS] Оценка для: BloodPressure
Train MAE: 8.06, RMSE: 10.43
Test MAE: 10.31, RMSE: 14.23
▲ Возможное переобучение (test/train MAE = 1.28)

[ANFIS] Оценка для: SkinThickness
Train MAE: 5.80, RMSE: 7.28
Test MAE: 7.69, RMSE: 9.87
▲ Возможное переобучение (test/train MAE = 1.33)

[ANFIS] Оценка для: Insulin
Train MAE: 59.23, RMSE: 83.42
Test MAE: 66.82, RMSE: 97.81
— Умеренное обобщение (test/train MAE = 1.13)

[ANFIS] Оценка для: BMI
Train MAE: 3.49, RMSE: 4.40
Test MAE: 4.80, RMSE: 6.14
▲ Возможное переобучение (test/train MAE = 1.38)
```

Рис. 5. Расчет средней абсолютной ошибки и среднеквадратического отклонения на обучающей и тестовой выборках

При увеличении количества генерируемых правил, например, с 3 до 7, ошибка на обучающей выборке продолжает снижаться, в то время как на тестовой выборке все больше наблюдается явление переобучения. Такой результат связан с высокой параметризацией системы ANFIS и с малым объемом полных данных для обучения.

На рис. 6 приведена визуализация средних абсолютных ошибок для разных методов, ANFIS показывает лучшие результаты, заполнение средним по кластерам с использованием нечеткой принадлежности дает схожие результаты, а по признаку «Индекс массы тела» — «BMI», пре-восходит по качеству нейро-нечеткую систему.

В условиях малых данных простые, устойчивые методы могут превосходить сложные обучаемые модели, Fuzzy Cluster Imputer показывает более высокую точность при увеличении числа кластеров с 3 до 7, результаты расчета средней абсолютной ошибки для разного числа кластеров приведены в табл. 1.

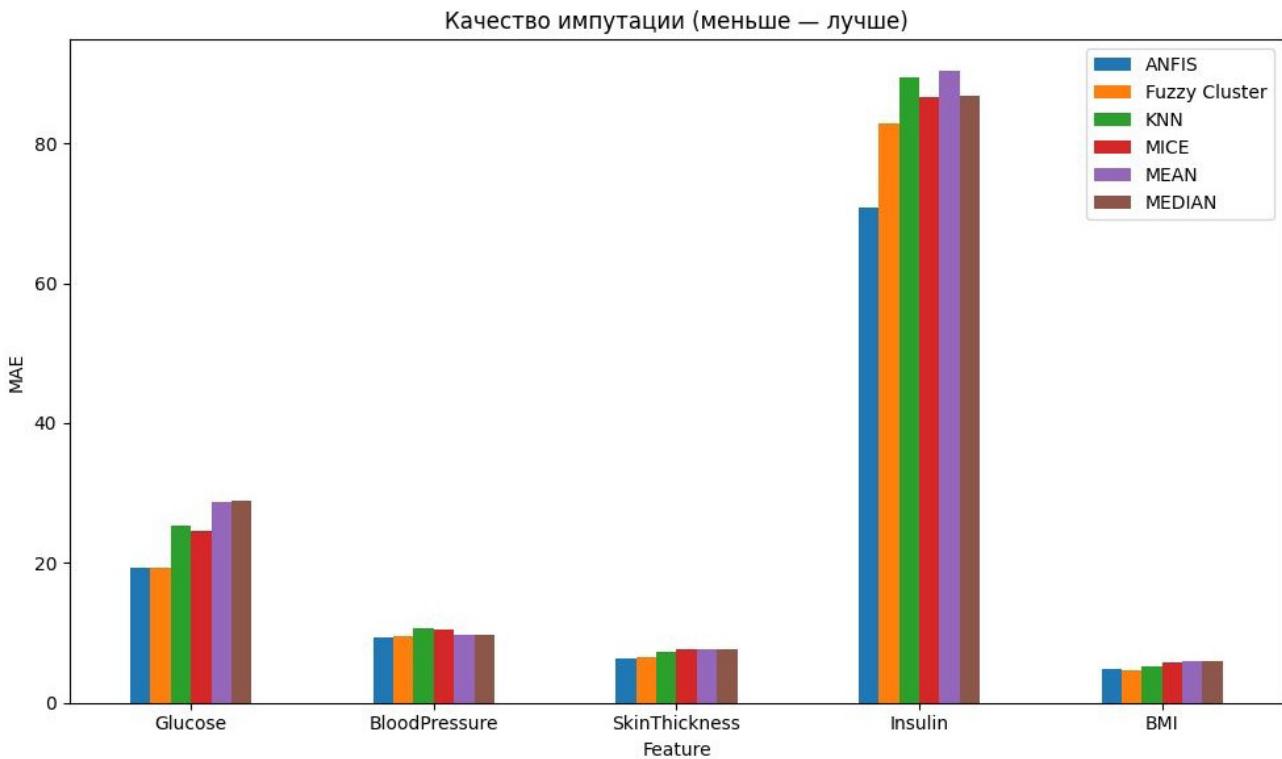


Рис. 6. Визуализация средней абсолютной ошибки, полученной при заполнении пропусков различными методами

Таблица 1

Средняя абсолютная ошибка по признакам

Количество кластеров	Уровень глюкозы	Кровяное давление	Толщина кожи	Уровень инсулина	Индекс массы тела
2	23.23	10.26	6.53	83.18	5.44
3	23.39	9.56	6.58	82.84	4.61
4	22.81	9.53	6.8	85.3	4.91
5	20.68	9.08	5.94	76.97	4.46
6	18.8	9.45	6.29	71.02	4.19
7	19.76	9.11	6.1	67.29	4.08

Заключение

В статье была рассмотрена проблема применения нечетких информационных технологий для заполнения пропусков в неполных данных. Проведен сравнительный анализ гибридных подходов и нескольких стандартных методов заполнения пропущенных значений.

Исходя из результатов анализа, можно сделать вывод о том, что оба предложенных гибридных подхода превосходят стандартные методы по точности заполнения пропусков.

Нечеткий кластерный импутер особенно хорошо проявляет себя для признаков с явно выраженной кластерной структурой, увеличение числа кластеров позволяет существенно улучшить метрики качества импутации.

Адаптивная нейро-нечеткая система вывода (ANFIS) с моделью нечеткого вывода Такаги-Сугено первого порядка является параметрической моделью с большим числом обучаемых параметров из-за чего при малом числе данных модель начинает запоминать информацию вместо обобщения и наблюдается явление переобучения. Важными преимуществами ANFIS

являются интерпретируемость правил и высокий потенциал обучения при большом объеме данных. ANFIS можно комбинировать с другими моделями с целью получения более эффективной ансамблевой модели. [5]

Выбор метода заполнения пропусков во многом зависит от специфики конкретной задачи или исследования. Полные данные могут дать более значимые результаты, если они не иска- жены систематическими пропусками. Однако если пропуски носят неслучайный характер, не- обходимо использовать специальные методы обработки пропусков, способные выявлять за- кономерности в данных, чтобы избежать искажений и потери статистической значимости. [4]

Литература

1. Теория и практика машинного обучения / В. В. Воронина, А. В. Михеев, Н. Г. Ярушкина, К. В. Святов. – Ульяновск : УлГТУ, 2017. – 290 с.
2. Хайндман Р. Прогнозирование : принципы и практика / пер. с англ. А. Логунов / Р. Хайн-дман, Дж. Атанасопулос. – Москва : ДМК Пресс, 2023. – 382 с.
3. Rubin D. Inference and missing data / D. Rubin // Biometrika – 1976 P. – 581-592.
4. Emmanuel T. A survey on missing data in machine learning / T. Emmanuel, T. Maupong, D. Mpoeleng // Journal of Big Data – 2021.
5. Vijayakumar V. Grey Fuzzy Neural Network-Based Hybrid Model for Missing Data Imputaion in Mixed Database / V. Vijayakumar, I. Paramasivam // International Jouranal of Inelligent Engineering & Systems – 2017.
6. Chai Y. Mamdani Model based Adaptive Neural Fuzzy Inference System and its Application / Y. Chai, L. Jia, Z. Zhang // International Journal of Computer, Electrical, Automation, Control and Information Engineering. – 2009.
7. Nijman S. Missing data is poorly handled and reported in prediction model studies using machine learning: a literature review / S. Nijman // Journal of Clinical Epidemiology – 2022. P. 218–229.
8. Shadbahr T. The impact of imputation quality on machine learning classifiers for datasets with missing values / T. Shadbahr, M. Roberts, J. Stanczuk // Communications Medicine. – 2023 P. 139–146.
9. El-Bakry M. Fuzzy based Techniques for Handling Missing Values / M. El-Bakry, A. El-Kilany, S. Mazen, F. Ali // International Journal of Advanced Computer Science and Applications – 2021.
10. Центр статистического анализа. – URL: <https://scc.ms.unimelb.edu.au/> /resources/preparing-your-data/missing-data/ (дата обращения: 01.04.2025).

ИСПОЛЬЗОВАНИЕ ТЕХНОЛОГИЙ АВТОМАТИЧЕСКОГО РАСПОЗНАВАНИЯ ЭМОЦИЙ ДЛЯ ДИАГНОСТИКИ И ПОДДЕРЖКИ ДЕТЕЙ С РАССТРОЙСТВАМИ АУТИСТИЧЕСКОГО СПЕКТРА

К. А. Лихачева, М. Ю. Лихачев

Воронежский государственный университет

Аннотация. Расстройства аутистического спектра (PAC) — многоаспектные нейроразвитийные расстройства, затрудняющие социальное взаимодействие и коммуникацию. В последние годы внедрение автоматических методов анализа речи и эмоций становится одним из перспективных направлений в разработке средств диагностики и поддержки данной категории детей. В данной статье рассматриваются современные методы автоматического распознавания эмоций (SER), их применение в контексте диагностики и терапии детей с PAC, а также перспективные направления развития технологий. Обоснована актуальность автоматизации оценки эмоционального состояния, что способствует более объективной и своевременной диагностике, а также расширяет возможности индивидуализированной терапии.

Ключевые слова: автоматическое распознавание эмоций, речь, расстройства аутистического спектра, машинное обучение, диагностика, реабилитация.

Введение

Расстройства аутистического спектра (PAC) представляют из себя сложные нейроразвитийные состояния, проявляющиеся нарушениями коммуникации, социального взаимодействия и наличием стереотипных интересов и поведения. Традиционные методы диагностики данных расстройств основаны на наблюдении, беседах и использовании анкетных опросников, что зачастую требует много времени и субъективных оценок со стороны специалистов [1].

В связи с этим, растет интерес к автоматизированным системам, способным анализировать речь и эмоциональные реакции детей с PAC. Точное и быстрое определение эмоциональных состояний позволит повысить качество диагностики, мониторинга и терапии, а также поможет адаптировать методы коррекции в реальном времени [4]. Значительную роль в этом играет автоматическое распознавание эмоций, которое позволяет выявлять эмоциональные отклики по акустическим признакам речи.

1. Методы автоматического распознавания эмоций

Для распознавания эмоций у людей с аутизмом используются различные методы: визуальные пособия (карточки с эмоциями), социальные истории, когнитивно-поведенческая терапия, а также специализированные программы и технологии, например, приложения с использованием дополненной реальности. Важно также использовать стратегии для обучения и саморегуляции, такие как дыхательные упражнения и практика осознанности. Одним из данных направлений является распознавание эмоций по речевым признакам.

Распознавание речевых эмоций (SER) — это область исследования, которая занимается выводом человеческих эмоций из речевых сигналов. Такие системы сосредоточены на идентификации голосового ввода как относящегося к различным категориям эмоций. Контролируемые методы машинного обучения, известные своей эффективностью, часто используются во многих исследованиях распознавания эмоций речи [3].

Для обучения моделей в SER используется датасеты с записанными речевыми образцами. В идеале, такие датасеты должны содержать акустические записи детей с РАС, аннотированные по эмоциональному признаку (радость, страх, печаль, гнев и т.п.).

Предварительная обработка аудио включает шумоподавление, нормализацию громкости, сегментацию речи на отдельные фрагменты. Данные процессы имеют большое значение для стабильной работы алгоритмов [5].

Наиболее широко применяемые признаки аудиоданных, используемые в подобных задачах — мел-частотные спектральные характеристики (MFCC), спектрограммы, ритмические параметры, и тембровые признаки. Они позволяют моделям «понимать» акустическую структуру речи.

Для обучения подобных моделей используются классические алгоритмы машинного обучения, такие как метод опорных векторов (SVM), случайные леса, и нейронные сети, такие как сверточные (CNN), рекуррентные (RNN), а также трансформеры.

Задачи машинного обучения в SER, как правило, относятся к задачам классификации. Их эффективность оценивается такими метриками, как точность, полнота, F1-оценка.

Использование сверточных и рекуррентных нейронных сетей в данных задачах позволяет добиться высокой точности распознавания эмоций на стандартных датасетах. Например, исследования подтверждают, что такие системы могут достигать точности выше 80% в классификации базовых эмоций у взрослых. Для детей с РАС задачи усложнены из-за вариативности выражения эмоций, поэтому требуется создание специальных датасетов и алгоритмов, учитывающих особенности данного контингента [5].

2. Перспективы применения методов SER

В области диагностики автоматическое распознавание эмоций помогает специалистам получать дополнительные объективные показатели по эмоциональной реакции ребенка во время обследования. Это особенно важно при диагностике детей с РАС, у которых выраженность эмоциональных проявлений может быть затруднена или искажена [3].

В реабилитации и терапии благодаря мониторингу в реальном времени системы могут отслеживать эмоциональное состояние ребенка в ходе занятий, предоставляя ценную обратную связь родителям и педагогам. Полученные данные о реакциях помогают корректировать терапевтическую стратегию в реальном времени. Доступные к самостоятельной установке и использованию приложения с функциями SER позволяют родителям в домашних условиях лучше понимать эмоциональное состояние ребенка и реагировать на него.

3. Сложности и проблемы использования методов SER

Для обеспечения качества используемых данных необходимо создание больших и репрезентативных датасетов с записями детей с РАС. В процессе обработки данных могут возникнуть некоторые сложности с акустическими условиями, такие как шум, фоновая музыка, а также разные микрофоны оказывают влияние и могут ухудшить качество распознавания.

Также имеют место быть затруднения в этических аспектах, такие как обеспечение приватности и защита персональных данных, а также соблюдение этических стандартов при сборе данных [2].

Необходимо также учитывать и особенности выражения эмоций у детей с РАС. У разных пациентов возможны отличия в выражении одних и тех же эмоций, что требует разработки специальных алгоритмов.

Несмотря на описанную выше проблематику, использование технологий автоматического распознавания эмоций в контексте диагностики и поддержки детей с аутизмом значимая

перспективная область, сочетающая достижения искусственного интеллекта и практические потребности медицины. Внедрение таких систем обещает повысить точность оценки эмоционального состояния, сократить субъективность и повысить эффективность коррекционно-реабилитационных мероприятий.

Тем не менее, для широкого внедрения необходимо продолжить разработку и адаптацию алгоритмов к особенностям данной категории детей, расширять базы данных и учитывать этические стандарты. В перспективе, интеграция SER с другими технологиями, например, компьютерным зрением, позволит создать мульти-модальные системы, значительно расширяющие возможности поддержки и диагностики.

Заключение

Современные достижения в области автоматического распознавания эмоций представляют собой инструмент, открывающий новые горизонты в диагностике, лечении и поддержке детей с аутизмом. Использование технологий SER позволяет объективно и систематически оценивать эмоциональные реакции детей, что особенно важно в условиях, когда традиционные методы часто основываются на субъективном наблюдении и могут быть ограничены в своей точности и последовательности.

Одним из ключевых преимуществ внедрения систем автоматического распознавания эмоций является возможность повысить точность диагностики эмоциональных особенностей детей с аутизмом. Автоматизированные алгоритмы способны выявлять нюансы эмоциональных проявлений, которые могут ускользать при ручной оценке, а также обеспечить постоянный мониторинг динамики эмоционального состояния в реальном времени. Это способствует более быстрому реагированию и подбору индивидуальных методов коррекции и реабилитации, что в конечном итоге способствует повышению эффективности терапии и улучшению качества жизни детей [3].

Перспективы развития технологий автоматического распознавания эмоций включают интеграцию с другими мультимодальными системами, например, компьютерным зрением, акустической обработкой речи и биометрическими датчиками. Такой мульти-модальный подход способен существенно расширить возможности диагностики и поддержки, позволяя комплексно оценивать состояние ребенка и предлагать более точные и персонализированные методы вмешательства.

В целом, внедрение систем SER в клиническую практику открывает новые возможности для более объективной, быстрой и эффективной оценки эмоциональных реакций детей с аутизмом. При правильной адаптационной работе и соблюдении этических стандартов они способны стать важной частью современного арсенала терапевтических и диагностических методов. В перспективе такие технологии будут играть важную роль в создании условий для более эффективной реабилитации, содействуя формированию условий для гармоничного развития и повышения качества жизни детей с аутизмом и их семей.

Литература

1. Максимова Е. В. Диагностика нарушения восприятия особых детей с опорой на теорию построения движений Н. А. Бернштейна / Е. В. Максимова // Психология и педагогика: методика и проблемы практического применения. – 2018. – № 59. – С. 59–63.

2. Нейсон Б. О ключевых проблемах аутизма. Сенсорные аспекты аутизма / Б. Нейсон // Аутизм и нарушения развития. – 2017. – № 35. – С. 35–41.

3. Переверзева Д. С. Особенности зрительного восприятия у детей с расстройствами аутистического спектра / Д. С. Переверзева // Современные проблемы науки и образования. – 2019. – № 21. – С. 21–26.

4. Шпицберг И. Л. Коррекция нарушений развития сенсорных систем у детей с расстройствами аутистического спектра / И. Л. Шпицберг // Коррекционная педагогика: теория и практика. – 2020. – № 42. – С. 42-48.

5. Schuller B. W. Speech Emotion Recognition: Two Decades in a Nutshell, Benchmarks, and Ongoing Trends / B. W. Schuller, A. Batliner // Communications of the ACM. – 2018. – Vol. 61(5). – P. 90–99.

6. Schuller B. W. Speech Emotion Recognition: Two Decades in a Nutshell, Benchmarks, and Ongoing Trends / B. W. Schuller, A. Batliner // Communications of the ACM. – 2018. – Vol. 61(5). – P. 90–99.

ПРОТОКОЛ MCP КАК АРХИТЕКТУРНАЯ ОСНОВА ДЛЯ РАЗРАБОТКИ AI-АГЕНТОВ

Н. Р. Мишин

Воронежский государственный университет

Аннотация. Статья посвящена архитектуре AI-агентов, построенных на основе языковых моделей с возможностью подключения к внешним цифровым инструментам. Описываются функциональные компоненты таких агентов: механизм поиска информации, работа с памятью и выполнение действий через вызовы инструментов. Представлен протокол Model Context Protocol (MCP), обеспечивающий структурированное взаимодействие модели с внешними API, базами данных и сервисами. Приведен пример клиент-серверной архитектуры на основе MCP, а также описаны принципы интеграции новых инструментов для расширения возможностей языковой модели.

Ключевые слова: AI-агент, языковая модель, Model Context Protocol, Retrieval-Augmented Generation, MCP-клиент, MCP-сервер, протокол взаимодействия, клиент-серверная архитектура, архитектура агентов, искусственный интеллект, обработка данных, инструменты.

Введение

Большие языковые модели открывают широкие возможности для автоматизации рутинных задач, начиная от генерации текста и заканчивая многошаговой обработкой запросов. Однако для решения задач, выходящих за рамки генерации текста, требуется интеграция таких моделей с внешними источниками данных. Поэтому в последние годы активно развивается подход, при котором большие языковые модели становятся частью AI-агентов.

При этом на практике возникает проблема: современные модели не имеют прямого доступа к внешнему окружению, и для выполнения действий за пределами внутреннего контекста требуется организация интерфейса взаимодействия. Варианты интеграции моделей с API, базами данных, файловыми системами часто реализуются в виде специализированных решений, которые не имеют общего стандарта.

В качестве решения данной проблемы предлагается Model Context Protocol (MCP) — протокол, предназначенный для стандартизации обмена данными между языковой моделью и внешним миром.

В данной статье описываются принципы построения современных AI-агентов, особенности использования инструментов и памяти в их работе, а также механизм подключения языковой модели к цифровой инфраструктуре с помощью протокола MCP.

1. AI-агенты

Современные AI-агенты [1] представляют собой системы, которые способны не только интерпретировать запросы, но и выполнять действия, запрашивать данные и сохранять информацию в памяти. Архитектурно такие агенты строятся на основе языковой модели, расширенной дополнительными механизмами.

Базовые языковые модели, на которых основаны такие агенты, обучаются на большом количестве текстовых данных и демонстрируют способность к обобщению и генерации осмысленных ответов. Однако без доступа к внешним источникам информации, программным интерфейсам и памяти, их применимость остаётся ограниченной.

Чтобы преодолеть эти ограничения, в архитектуру агента добавляются функциональные модули, которые позволяют модели действовать динамически: выполнять вызовы к инстру-

ментам, извлекать релевантные данные и учитывать накопленный контекст в процессе общения. Такое расширение превращает языковую модель в компонент не только рассуждающий, но и способный к действию.

Далее приведены три ключевые возможности AI-агентов.

Механизм поиска позволяет модели запрашивать информацию из внешней базы знаний: векторного хранилища или поискового индекса. Запрос формируется автоматически на основе исходного задания, и найденные фрагменты передаются обратно в модель в качестве дополнительного контекста. Такой подход известен как Retrieval-Augmented Generation (RAG) и позволяет агенту опираться не на память обученной модели, а на актуальные, обновляемые данные из базы знаний или документации.

Механизм памяти реализуется через сохранение промежуточной информации в кратковременное или долговременное хранилище. Это может быть история диалогов, состояние задачи, факты о пользователе или внешние знания, полученные в процессе работы. При последующих обращениях память восстанавливается и включается в контекст. Таким образом достигается последовательность поведения агента и способность к межсессионному обучению.

Модель, интегрированная в агента, получает описание доступных инструментов в виде структурированных спецификаций (имя, параметры, результат) [2]. В процессе генерации модель может принять решение о необходимости вызова инструмента и сгенерировать соответствующую инструкцию. Эта инструкция передаётся агентом на исполнение и результат возвращается обратно для последующей обработки в общий контекст.

Такой механизм позволяет модели выходить за пределы обычной генерации, выполняя реальные действия: запросы к API, SQL-запросы, манипуляции с файлами и так далее.

Однако, чтобы взаимодействие между моделью и внешними инструментами было управляемым и масштабируемым, необходим способ описания и вызова этих функций. Этую задачу решает протокол MCP.

2. Использование протокола MCP

Model Context Protocol (MCP) представляет собой механизм подключения внешних инструментов к языковой модели [3]. Он реализует клиент-серверную архитектуру, в которой модель взаимодействует с промежуточным компонентом — MCP-клиентом, который координирует обращения к одному или нескольким MCP-серверам. Каждый сервер предоставляет интерфейс к конкретному ресурсу: API, базе данных, файловой системе или вычислительному модулю. На рис. 1 представлена обобщённая схема архитектуры MCP:

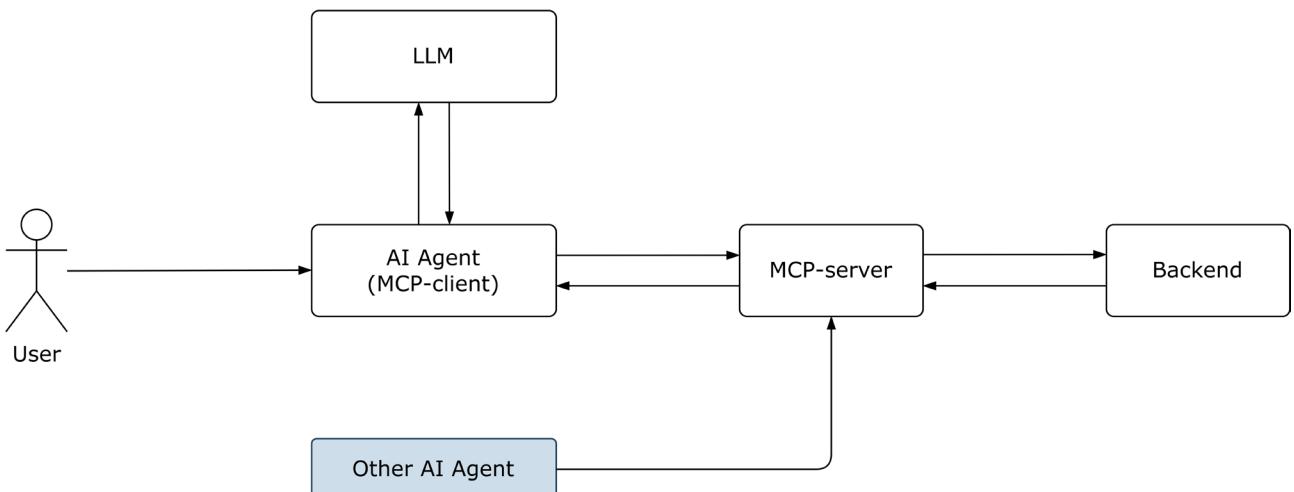


Рис. 1. Архитектура MCP

Взаимодействие между моделью и внешними компонентами по протоколу MCP осуществляется в следующей последовательности:

1. MCP-клиент получает от подключённых серверов описания доступных инструментов (их имена, параметры, форматы ответов)
2. Спецификации инструментов передаются в контекст модели
3. Модель, обнаружив необходимость внешнего действия, формирует структуру вызова
4. MCP-клиент валидирует вызов, пересыпает его на соответствующий MCP-сервер;
5. Сервер выполняет операцию и возвращает результат
6. Клиент передаёт результат обратно модели, которая продолжает рассуждение с учётом новых данных.

На рис. 2 показан пример такой последовательности в виде диаграммы запросов.

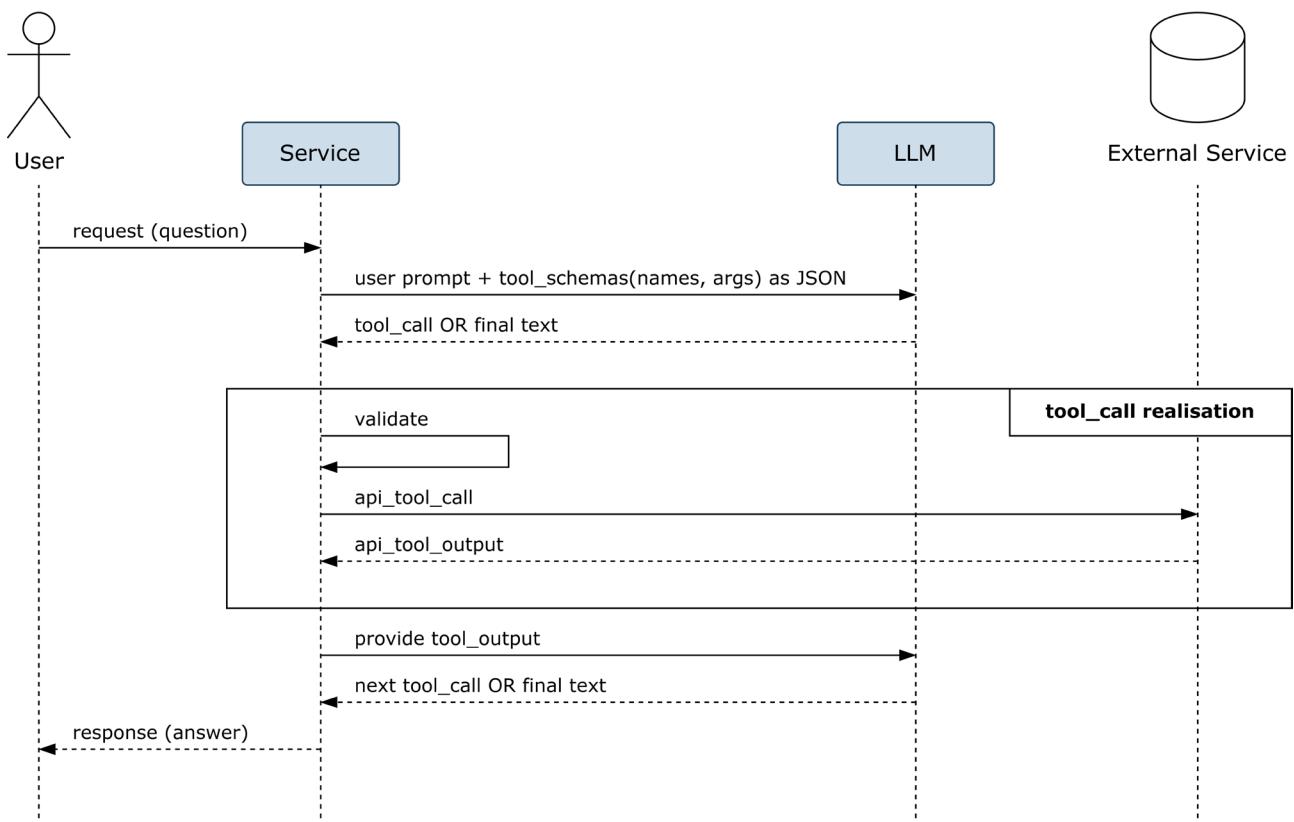


Рис. 2. Диаграмма запросов

Коммуникация между компонентами основана на сериализованных форматах, как правило JSON, что обеспечивает межъязыковую совместимость и упрощает внедрение в существующие инфраструктуры. Модель не имеет прямого доступа к внешним сервисам. Все вызовы проходят через MCP-клиент, который обеспечивает проверку данных, маршрутизацию и контроль выполнения.

Протокол поддерживает реализацию политик контроля доступа: можно ограничивать набор допустимых инструментов, проводить аутентификацию, журналировать вызовы и применять фильтрацию по параметрам. Такой подход особенно важен при использовании агента в сценариях с доступом к конфиденциальным или критически важным ресурсам. Все команды модели проверяются до исполнения, что снижает вероятность ошибок и неправомерных действий.

Одним из ключевых преимуществ MCP является модульность. Разработчики могут добавлять новые инструменты без внесения изменений в модель. Достаточно развернуть соответствующий MCP-сервер и передать его описание клиенту. Модель автоматически получит до-

ступ к новой функциональности. Это упрощает масштабирование и повторное использование компонентов в разных системах.

Таким образом, MCP выступает в роли интерфейса между языковой моделью и окружающей средой. Он обеспечивает прозрачное, безопасное и управляемое выполнение внешних действий, расширяя возможности модели и позволяя строить полноценные AI-агенты.

Заключение

Агенты на основе языковых моделей находят всё более широкое применение в инженерных и прикладных задачах, требующих не только понимания текста, но и взаимодействия с внешними источниками информации и сервисами. Эффективность таких систем определяется не только качеством самой модели, но и тем, насколько гибко и безопасно она может быть подключена к окружающей инфраструктуре.

Протокол MCP предоставляет структурированный способ организации такого взаимодействия. Он обеспечивает модульное, управляемое и расширяемое соединение между моделью и внешними инструментами. Такая архитектура позволяет создавать интеллектуальных агентов, способных не просто отвечать на вопросы, но и инициировать действия, принимать решения и адаптироваться к задаче в процессе работы.

Использование языковых моделей в связке с MCP формирует основу для разработки систем, где модель выступает не как продвинутый генератор текста, а как управляющий компонент, координирующий работу с данными и инструментами.

Литература

1. *Лэнхэм М. AI Agents in Action / М. Лэнхэм.* – Manning Publications, 2025 – 344 с.
2. *Крейг У. Spring AI in Action / У. Крейг.* – Manning Publications, 2025 – 320 р.
3. Документация Model Context Protocol. – URL: <https://modelcontextprotocol.io/docs> (дата обращения: 20.11.2025).

АЛГОРИТМЫ ОБУЧЕНИЯ С ПОДКРЕПЛЕНИЕМ ДЛЯ НЕПРЕРЫВНОГО ПРОСТРАНСТВА ДЕЙСТВИЙ

К. С. Некрасов, И. Л. Каширина

Воронежский государственный университет

Аннотация. Рассмотрена общая постановка задачи, которую решают алгоритмы обучения с подкреплением. Даны соответствующие определения и обозначена специфика алгоритмов, где пространство действий непрерывно. Сделан обзор ряда алгоритмов обучения с подкреплением, которые работают с непрерывным пространством действий. Проведен краткий анализ соответствующих исследований, приведены ключевые формулы и выводы. Сделан акцент на практической применимости соответствующих алгоритмов в прикладных задачах.

Ключевые слова: обучение с подкреплением, reinforcement learning, искусственный интеллект, машинное обучение, марковский процесс принятия решений.

Введение

Обучение с подкреплением (англ. reinforcement learning) представляет собой класс алгоритмов обучения нейронных сетей без учителя, где коррекция весов модели происходит как реакция на взаимодействие сети со средой. Другими словами, обучение с подкреплением позволяет модели «учиться на собственных ошибках», применяя действия к среде и оценивая (через заданную формулу наград) успешность примененного действия. Обучение с подкреплением находит применение во множестве разнородных задач, начиная от игры в компьютерные игры (где входными данными могут служить пиксели экрана), управления механизмами на производстве и заканчивая управлением беспилотными автомобилями и роботизированными системами.

В работе рассматриваются алгоритмы обучения с подкреплением, работающие в непрерывном пространстве действий.

1. Общие положения

Существуют три основных парадигмы машинного обучения: обучение с учителем, обучение без учителя и обучение с подкреплением. Последняя интересна тем, что модель не требует размеченных данных (в отличие от обучения с учителем) и работает с набором данных, не заданным статически (как в случае с обучением без учителя).

Модель обучения с подкреплением выполняет действия над средой посредством агента и оценивает успешность примененных действий через заданную формулу наград. Целью такой модели является максимизация получаемых наград. Иными словами, обучение с подкреплением можно определить как обучение через взаимодействие. Заметим, что сложности взаимодействия со средой инкапсулированы в агенте, что позволяет применять алгоритмы обучения с подкреплением в широчайшем спектре задач.

Математической основой алгоритмов обучения с подкреплением являются марковские процессы принятия решений (англ. Markov Decision Process, MDP) [1, с. 47–71]. В рамках формализма MDP в каждый момент дискретного времени $t = 0, 1, 2, \dots$ агент взаимодействует со средой, описываемой состоянием $S_t \in S$ и на основе него выбирает действие $A_t \in A$. На следующем шаге агент получает численную награду $R_{t+1} \in R \subset \mathbb{R}$ и оказывается в новом состоянии $S_{t+1} \in S$ (рис. 1).

Таким образом, процесс MDP описывается:

- множеством состояний S и распределением стартовых состояний $p(S_0)$;
- множеством действий A ;
- динамикой переходов $P(S_{t+1} | S_t, A_t)$, отображающей пару «состояние–действие» в момент времени t на распределение состояний в момент времени $t+1$;
- функцией мгновенных наград $R(S_t, A_t, S_{t+1})$;
- коэффициентом дисконтирования $\gamma \in [0,1]$.

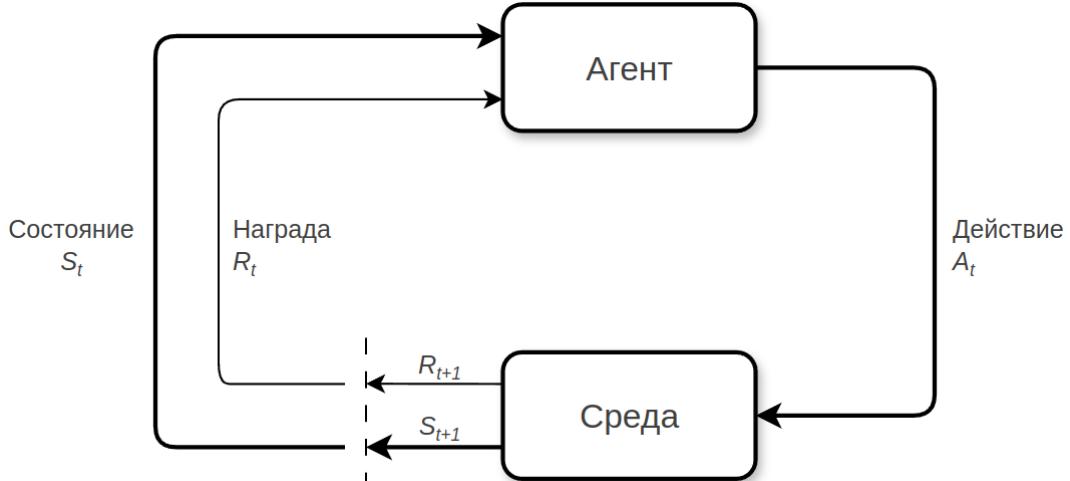


Рис. 1. Взаимодействие агента и среды в марковском процессе принятия решений

Эпизодом (траекторией) длины T называется последовательность состояний, действий и наград, которая заканчивается конечным состоянием. В конце каждого эпизода вычисляется общая дисконтированная награда вида $G = \sum_{k=0}^{T-1} \gamma^k R_{k+1}$.

Политика π является в общем случае вероятностным отображением множества состояний на множество действий вида $\pi: S \rightarrow p(A = a | S)$. Целью агента является нахождение такой политики π^* , которая приведет к максимизации ожидаемых дисконтированных сумм наград: $\pi^* = \arg \max_{\pi} E[G | \pi]$.

Функция (1) называется функцией ценности «состояние–действие». В работах также часто используется функция (2), которая называется функцией ценности состояния:

$$Q_{\pi}(s, a) = E_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_k | S_t = s, A_t = a \right], \quad (1)$$

$$V_{\pi}(s) = E_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_k | S_t = s \right]. \quad (2)$$

В контексте обучения с подкреплением важным является марковское свойство стохастического процесса, утверждающее, что условное распределение вероятностей будущих состояний процесса зависят только от текущего состояния и не зависят от истории предыдущих состояний.

Простейшие алгоритмы обучения с подкреплением подразумевают дискретные множества A и S (такие алгоритмы часто реализуются методами динамического программирования; см., например, Q-Learning [2]). Помимо них есть алгоритмы, работающие с непрерывным множеством состояний, но с дискретным множеством действий A (известным примером таких алгоритмов является DQN [3]). Данное исследование фокусируется на алгоритмах, работающих с непрерывным множеством действий.

2. Классификация моделей обучения с подкреплением, работающих с непрерывным пространством действий

2.1. Алгоритмы на основе градиента политики

Методы на основе градиента политики предполагают, что политика π задана вектором параметров $\theta \in \mathbb{R}^d$. Предлагается использовать некоторую оценку эффективности $J(\theta)$, которая используется для обновления параметров политики:

$$\theta_{t+1} = \theta_t + \alpha \overline{\nabla J(\theta_t)}, \quad (3)$$

где выражение $\overline{\nabla J(\theta_t)} \in \mathbb{R}^d$ представляет случайную оценку, чье математическое ожидание аппроксимирует соответствующий градиент оценки $J(\theta_t)$.

Алгоритм REINFORCE [4] использует семплирование методом Монте-Карло для получения эмпирической оценки ожидаемых наград. Правило обновления параметров политики в алгоритме REINFORCE выглядит следующим образом: $\theta_{t+1} = \theta_t + \alpha G_t \nabla_\theta \ln \pi(A_t | S_t, \theta_t)$. В силу использования алгоритма Монте-Карло на практике алгоритм REINFORCE может показывать медленную скорость обучения. Низкая скорость обучения обусловлена как необходимостью прогона полного эпизода для того, чтобы вычислить новый градиент, так и большой выборочной дисперсией.

В качестве одной из оптимизаций алгоритма предлагается использовать методику опорных точек (англ. baseline). Основная идея методики в том, что в правиле REINFORCE (1) вместо прямых дисконтированных наград G_t используются выражения $G_t - b$, где параметр b подбирается эвристически.

Последующие исследования проблем скорости обучения и практической сходимости алгоритмов на основе градиента политики породили метод натурального градиента политики (англ. Natural Policy Gradient). В работе [5] автор показывает, что движение в сторону увеличения градиента относительно параметров θ ведет в общем случае к нестабильности алгоритма, поскольку не учитывает кривизну распределения политики $\pi(\theta)$, так что малому изменению параметров «в сторону» увеличения градиента может соответствовать драматически неоптимальное изменение величины $\pi(\theta + \Delta\theta)$. Чтобы решить проблему неоптимальных шагов в [6] предлагается использовать информационную метрику Фишера. Таким образом, натуральный градиент имеет вид (4):

$$\hat{\nabla} J(\theta) = F(\theta)^{-1} \nabla J(\theta), \quad (4)$$

где матрица метрики Фишера локально может быть представлена через расстояние Кульбака — Лейблера [6]:

$$F(\theta) = \nabla_\theta^2 D_{KL}(\pi_\theta(x) \| \pi_{\theta+\Delta\theta}(x))|_{\Delta\theta=0}, \quad (5)$$

и в результате правило обновления параметров (1) принимает вид (4):

$$\theta_{t+1} = \theta_t + \sqrt{\frac{2\epsilon}{\nabla J(\theta_t)^T F(\theta_t)^{-1} \nabla J(\theta_t)}} \hat{\nabla} J(\theta_t). \quad (6)$$

2.2. Алгоритмы «актор-критик»

Алгоритмы «актор-критик» являются логическим продолжением и улучшением алгоритмов на основе градиента политики. Вместо оценки функции ценности $Q_\pi(s, a)$ посредством семплирования (как в случае REINFORCE), алгоритмы «актор-критик» ее аппроксимируют (как правило, нейронной сетью). Сеть, которая аппроксимирует функцию ценности $Q_\pi(s, a)$, называется критиком, аппроксиматор политики $\pi_\theta(s | a)$ называется актором.

В алгоритме DPG (Deterministic Policy Gradient) [7] стохастическую политику $\pi_\theta : S \rightarrow p_\theta(A = a | S)$ предлагается заменить детерминированной политикой вида $\mu_\theta : S \rightarrow A$. В работе [7] показывается, что в этом случае $\nabla_\theta J(\theta) \approx E_\mu \left[\nabla_\theta \mu(s, \theta) \cdot \nabla_a Q(s, a) \Big|_{a=\mu(s)} \right]$. Для того, чтобы обеспечить исследовательское поведение в детерминированном алгоритме, к выходу актора явным образом добавляется случайный шум (например, в виде стохастического процесса Орнштейна — Уленбека). Алгоритм представляет интерес, в основном, с теоретической точки зрения; с практической точки зрения куда более известен его наследник — Deep DPG (DDPG) [8].

DDPG сочетает теоретическую базу работы [7] (в частности, теорему о градиенте детерминированной политики) с практическими успехами сети DQN [3], показавшей выдающейся для своего времени результат устойчивой сходимости глубокой нейронной сети. В частности, использованы следующие особенности:

- Вводится вторая пара сетей «актор-критик» (называемые целевыми), которые обучаются с меньшим шагом обучения.
- Вводится буфер воспроизведения (англ. experience replay buffer) вида $D = \langle s, a, r, s' \rangle$, который пополняется новым кортежем после каждого перехода из состояния s в s' , получив награду r . Буфер организован по принципу кольцевой памяти, т. е. количество кортежей ограничено, и наиболее старые записи исчезают при переполнении.
- Обучение мини-пакетами (англ. mini-batching), когда данные для обучения «вне политики» семплируются по подмножеству буфера воспроизведения $D' \subset D$.

В DDPG критик обучается в режиме «вне политики» (англ. off-policy), используя адаптированный вариант Q-learning. В качестве функции потерь используется (7):

$$L(\theta^Q) = E_\beta \left[\left(Q(s_t, a_t | \theta^Q) - (r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1}) | \theta^Q)) \right)^2 \right], \quad (7)$$

где β — отдельная стохастическая политика.

Алгоритм TD3 (Twin-Delayed DDPG) является значительным улучшением алгоритма DDPG. Авторы в [9] показывают, что алгоритмы «актор-критик» подвержены проблеме систематического завышения оценки (англ. overestimation bias), которая ранее была исследована [10] в контексте табличных алгоритмов Q-learning. Авторы алгоритма TD3 предложили три меры улучшения:

- Использование двух критиков для оценки сверху максимально возможной ценности. Оценка сверху имеет вид (8):

$$y = r + \gamma \min_{i=1,2} Q_{\theta_i^*}(s', a'), \quad (8)$$

где θ_i — вектор параметров i -го критика.

• Регуляризация слаживания целевой политики. Авторы утверждают, что модель актора чувствительна к неточностям аппроксимирующей функции и переобучение пиковыми значениями ценности является проблемой. В качестве решения предлагается дополнительно обуславливать модель на узких областях вокруг точки целевого действия:

$$y = r + E_\varepsilon \left[Q_{\theta^*}(s', a' + \varepsilon) \right]. \quad (9)$$

• Задержка в обновлении целевых сетей. Как и оригинальный алгоритм DDPG, TD3 использует как «основные» сети «актор-критик», так и т. н. целевые сети (с учетом вышеуказанных правок, в каждой из них по два критика). Новым является замечание, что целевые сети не должны обновляться при каждом обновлении весов основной модели. В оригинальной работе предлагается обновлять их при каждом втором обновлении, но говоря вообще, эта частота может считаться гиперпараметром.

Заключение

В работе представлен краткий анализ некоторых наиболее известных алгоритмов обучения с подкреплением, которые работают с непрерывным пространством действий. В частности, рассмотрен классический алгоритм REINFORCE, рассмотрены решения на основе градиентов политики и представлены алгоритмы на основе архитектуры «актор-критик». Приведены основополагающие идеи и уравнения, лежащие в основании алгоритмов.

Литература

1. *Sutton R. S. Reinforcement learning: an introduction* / R. S. Sutton, A. G. Barto. – Cambridge : The MIT Press., 2020. – 526 p.
2. *Watkins C.J.C.H., Dayan P. Technical note: Q-Learning* // Machine Learning. – 1992. – Vol. 8. – P. 279–292.
3. *Mnih V. Playing Atari with Deep Reinforcement Learning* / V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller // arXiv. 2013. URL: <https://arxiv.org/abs/1312.5602> (дата обращения: 25.10.2025).
4. *Williams R. J. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning* // Machine Learning. – 1992. – Vol. 8. – P. 229–256.
5. *Kakade Sh. M. A Natural Policy Gradient* / Sh. M. Kakade // Advances in Neural Information Processing Systems. 2001. URL: https://papers.nips.cc/paper_files/paper/2001 (дата обращения 20.11.2025).
6. *van Heeswijk W. J. A. Natural Policy Gradients In Reinforcement Learning Explained* / W. J. A. van Heeswijk // arXiv. 2022. URL: <https://arxiv.org/abs/2209.01820> (дата обращения 20.11.2025).
7. *Silver D. Deterministic Policy Gradient Algorithms* / D. Silver, G. Lever, H. Nicolas, D. Thomas, W. Daan, R. Martin // Proceedings of Machine Learning Research. – 2014. – URL: <https://proceedings.mlr.press/v32/silver14.pdf> (дата обращения 25.10.2025).
8. *Lillicrap T. P. Continuous control with deep reinforcement learning* / T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra // arXiv. 2015. URL: <https://arxiv.org/abs/1509.02971> (дата обращения: 25.10.2025).
9. *Fujimoto S. Addressing Function Approximation Error in Actor-Critic Methods* / S. Fujimoto, H. van Hoof, D. Meger // arXiv. 2018. URL: <https://arxiv.org/abs/1802.09477> (дата обращения: 20.11.2025).
10. *Hasselt H. V. Double Q-learning* / H. V. Hasselt // Advances in Neural Information Processing Systems. 2010. URL: https://papers.nips.cc/paper_files/paper/2010 (дата обращения 25.11.2025).

ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ В АРХИТЕКТУРЕ: ВОЗМОЖНОСТИ И ПЕРСПЕКТИВЫ

П. А. Никольникова

Воронежский государственный университет

Аннотация. В статье рассматриваются возможности и перспективы применения технологий дополненной реальности (AR) в архитектурно-строительной отрасли. Описываются основные методы AR-визуализации, принципы реалистичного рендеринга и композитинга, а также методы отображения. Анализируются конкретные сценарии использования AR в строительстве, охватывающие визуализацию и контроль соответствия проекту, предоставление пошаговых инструкций, контроль качества, повышение безопасности и улучшение совместной работы. Особое внимание уделяется перспективам развития AR, включая ее интеграцию с искусственным интеллектом (ИИ) и Интернетом вещей (IoT), создание цифровых двойников строительных площадок и расширение применения в градостроительстве и реставрации архитектурного наследия.

Ключевые слова: дополненная реальность (AR), архитектура, визуализация, маркерное отслеживание, безмаркерное отслеживание, рендеринг, окклюзия, цифровой двойник, Искусственный интеллект (ИИ), Интернет вещей (IoT), градостроительство, реставрация, проектирование.

Введение

Архитектурно-строительная отрасль находится в постоянном поиске инновационных решений для повышения эффективности проектирования, строительства и управления эксплуатацией объектов. В этом контексте цифровые технологии играют ключевую роль. Можно выделить одну из наиболее перспективных технологий — технология дополненной реальность (AR).

В отличие от виртуальной реальности (VR), которая полностью погружает пользователя в синтетическую среду, AR накладывает генерированные цифровые объекты на реальное физическое окружение в режиме реального времени. Этот подход позволяет инженерам, архитекторам и строителям взаимодействовать с цифровым контентом, сохраняя при этом осведомленность о физическом пространстве.

Актуальность AR в архитектуре подкрепляется и общеотраслевыми трендами: повышением требований к детализации, необходимостью сокращения сроков строительства, повышением стандартов безопасности и стремлением к минимизации ошибок на этапе реализации. Внедрение дополненной реальности становится не просто технологическим экспериментом, а стратегическим инструментом, способным кардинально изменить привычные рабочие процессы.

1. Основные методы визуализации в AR

Эффективность дополненной реальности в архитектуре напрямую зависит от качества и реалистичности наложения виртуального контента на физическое окружение. Это достигается за счет комплекса методов, охватывающих отслеживание, рендеринг и композитинг, а также учет физических свойств света и геометрии.

1.1. Методы отслеживания и регистрации

Основой любой AR-системы является точное и стабильное определение положения и ориентации камеры или устройства пользователя в реальном мире. Без надежного отслеживания

виртуальные объекты будут «дрейфовать», нарушая иллюзию дополненной реальности. Различают два основных подхода:

- маркерное отслеживание: этот метод использует заранее известные графические маркеры (например, QR-коды, AR-метки), размещенные в реальном окружении. Камера устройства распознает эти маркеры, вычисляет их положение и ориентацию в пространстве, а затем использует эту информацию для точного позиционирования виртуальных объектов относительно маркера.

- преимущества: высокая точность, относительная простота реализации, устойчивость к изменяющимся условиям освещения при хорошем дизайне маркера;

- недостатки: требует физического размещения маркеров, ограничивает область взаимодействия, не подходит для широкомасштабных сценариев;

- применение в архитектуре: визуализация 3D-модели здания на печатном плане, демонстрация интерьера на маркере, расположенному на столе;

- безмаркерное отслеживание: этот более продвинутый метод не требует предварительно размещенных маркеров. Он использует алгоритмы Simultaneous Localization and Mapping (SLAM) для одновременной локализации устройства в пространстве и построения карты окружающей среды. Система анализирует визуальные особенности (ключевые точки/признаки) из видеопотока камеры, сопоставляет их в последовательных кадрах и на основе этих данных вычисляет движение камеры и создает трехмерную карту.

- преимущества: полная свобода перемещения, возможность визуализации в произвольных пространствах (от открытой строительной площадки до пустого помещения), масштабируемость;

- недостатки: высокая вычислительная сложность, чувствительность к однородным текстурам, резким изменениям освещения и быстрым движениям, потенциальный «дрейф» (накопление ошибки позиционирования со временем);

- применение в архитектуре: размещение полномасштабной виртуальной модели здания на пустом земельном участке, интерактивная «прогулка» по будущему интерьеру в существующем помещении, наложение инженерных сетей на незавершенные конструкции на стройке.

1.2. Методы рендеринга и композитинга

После того как положение и ориентация AR-устройства в пространстве определены, следующим шагом является реалистичное отображение цифровых архитектурных моделей и их бесшовное слияние с видеопотоком реального мира. Этот процесс включает в себя:

- оптимизацию 3D-моделей: архитектурные BIM-модели часто содержат избыточное количество полигонов, сложные материалы и высококачественные текстуры, что делает их непригодными для рендеринга в реальном времени на мобильных или AR-гарнитурах. Для AR-визуализации требуется существенная оптимизация:

- уменьшение полигонажа: использование алгоритмов для сокращения количества вершин и граней без значительной потери визуальной детализации;

- уровни детализации: Отображение упрощенных версий модели, когда объект находится далеко от пользователя, и более детализированных версий при приближении;

- оптимизация текстур и материалов: сжатие текстур, использование атласов текстур, преобразование Physically Based Rendering (PBR) материалов в более легковесные шейдеры для мобильных платформ;

- реалистичное освещение и тени: для создания иллюзии принадлежности виртуального объекта реальному миру необходимо, чтобы он корректно освещался и отбрасывал тени, соответствующие реальному окружению:

– оценка освещения окружения: современные AR-SDK могут анализировать видеопоток камеры для оценки параметров реального освещения (направление, интенсивность, цвет) и применять их к виртуальным объектам;

– динамические тени: виртуальные объекты должны отбрасывать тени на реальные поверхности, а также на другие виртуальные объекты. Это требует сложных алгоритмов рендеринга теней в реальном времени, таких как Shadow Mapping или Volumetric Shadows;

• окклюзия: один из самых сложных аспектов реалистичной AR-визуализации. Окклюзия означает правильное перекрытие объектов:

– виртуальные объекты, перекрывающие реальные: это относительно просто реализуется через буфер глубины, когда виртуальный объект рисуется поверх реального;

– реальные объекты, перекрывающие виртуальные: это значительно сложнее и требует информации о глубине реального мира;

– семантическая окклюзия: с использованием алгоритмов компьютерного зрения и машинного обучения для распознавания и сегментации реальных объектов (например, людей, мебели) и их последующего использования в качестве маски для виртуальных объектов.

1.3. Методы отображения

Способ, которым пользователь воспринимает дополненную реальность, также имеет решающее значение:

• Video See-Through (VST) Displays: камера устройства захватывает реальный мир, а затем видеопоток с наложенными виртуальными объектами отображается на экране.

– преимущества: широкая доступность, относительно низкая стоимость, высокое качество картинки с камеры;

– недостатки: непрямое взаимодействие с реальным миром («смотреть через экран»), ограниченное поле зрения, необходимость держать устройство в руках;

• Optical See-Through (OST) Displays: Пользователь смотрит на реальный мир через прозрачные линзы, а виртуальные изображения проецируются непосредственно в его поле зрения.

– преимущества: прямое восприятие реального мира, возможность работы без рук, более естественное ощущение «присутствия» виртуальных объектов;

– недостатки: высокая стоимость, ограниченное поле зрения (FOV), потенциальные проблемы с яркостью виртуальных объектов при ярком внешнем освещении, громоздкость.

2. Использование AR в строительстве

Технологии дополненной реальности (AR) находят все более широкое применение не только на этапах проектирования и визуализации, но и в реальном строительном процессе, а также на протяжении всего жизненного цикла здания, включая его эксплуатацию и техническое обслуживание. AR трансформирует традиционные подходы, предлагая новые инструменты для повышения эффективности, точности и безопасности работ.

2.1. Визуализация и контроль соответствия проекту

Одним из ключевых преимуществ AR на этапе строительства является возможность наложения цифровых информационных моделей зданий (BIM) или 3D-моделей непосредственно на реальную физическую среду. Рабочие и инженеры могут использовать планшеты или AR-очки для:

- визуализации проекта в масштабе 1:1: наложение виртуальных элементов (стен, коммуникаций, оборудования) на реальную площадку позволяет оперативно выявлять потенциальные коллизии и отклонения от проектной документации до их физического воплощения;
- проверки соответствия: быстрая оценка точности выполнения работ путем сравнения построенных элементов с их цифровыми двойниками;
- оптимизации размещения: AR может помочь в оптимальном расположении строительной техники, материалов и временных сооружений на площадке, улучшая логистику и безопасность.

2.2. Пошаговые инструкции и контроль качества

AR может служить мощным инструментом для предоставления рабочих инструкций и повышения качества выполнения задач:

- виртуальные инструкции: сложные монтажные операции могут быть разбиты на пошаговые виртуальные руководства, наложенные на рабочую среду. Это особенно полезно для обучения новых сотрудников или выполнения специализированных задач;
- контроль допусков: специалисты могут использовать AR для измерения и проверки геометрических параметров элементов в реальном времени, обеспечивая соответствие строительным нормам и допускам;
- мониторинг прогресса: наложение проектных планов на текущее состояние стройки позволяет визуально отслеживать прогресс выполнения работ и выявлять отставания.

2.3. Повышение безопасности и совместной работы

AR способствует созданию более безопасной рабочей среды и улучшает коммуникацию между участниками проекта:

- визуализация опасностей: AR-системы могут выделять потенциально опасные зоны, например, маршруты движения тяжелой техники, зоны работы кранов или места хранения опасных материалов;
- удаленная поддержка: опытные инженеры или консультанты могут предоставлять удаленную поддержку рабочим на месте, видя ту же AR-сцену, что и пользователь, и давая голосовые или визуальные указания;
- улучшенная коммуникация: общие AR-модели могут служить единым источником информации для всех участников проекта, минимизируя недопонимание и ошибки, вызванные устаревшими версиями чертежей или устными описаниями.

3. Перспективы развития AR-технологий в архитектуре

3.1. Интеграция с ИИ и IoT

Сочетание AR с искусственным интеллектом (ИИ) и интернетом вещей (IoT) открывает новые возможности для анализа данных в реальном времени и автоматизации процессов. Например, ИИ может обрабатывать информацию с датчиков на стройплощадке, погодные условия и чертежи, предлагая оптимальные решения для инженерных задач.

IoT позволяет подключать физические объекты к цифровым моделям, что особенно полезно для мониторинга состояния зданий и планирования ремонтных работ. Виртуальные элементы, наложенные на реальное окружение, помогают определить местоположение скрытых систем и элементов конструкции.

3.2. Развитие цифровых двойников стройплощадок

Концепция цифровых двойников в строительстве — это динамическая виртуальная репликация физической строительной площадки с обновлением данных в реальном времени. В отличие от статичных BIM моделей, цифровой двойник интегрирует множество источников: BIM-модель, данные IoT-датчиков, геопространственные данные, информацию о прогрессе работ, персонале, безопасности, план графики и бюджеты. Это позволяет отслеживать, анализировать и оптимизировать все аспекты строительного процесса с высокой точностью.

Создание и поддержание цифрового двойника требует комплекса технологий: BIM, IoT, лазерного сканирования, дронов, облачных вычислений, ИИ и AR/VR. IoT собирает данные с устройств, облачные вычисления обеспечивают масштабируемость и удалённый доступ, ИИ анализирует информацию и прогнозирует проблемы, а AR и VR визуализируют данные на месте и позволяют погружаться в виртуальные модели для планирования и обучения. Такие технологии повышают безопасность, сокращают сроки и затраты, улучшают качество управления строительством.

3.3. Расширение применения в градостроительстве и реставрации

В градостроительстве AR используется для визуализации проектов городского развития, оценки их воздействия на окружающую среду и улучшения взаимодействия с общественностью. Это позволяет более точно планировать новые объекты и инфраструктурные элементы, учитывая их влияние на пешеходные потоки, транспортную доступность и другие аспекты городской среды.

В управлении городской инфраструктурой AR помогает визуализировать подземные коммуникации, что критически важно для ремонтных работ и прокладки новых сетей. Интерактивные карты и навигация, созданные с использованием AR, улучшают ориентацию горожан и предоставляют информацию о достопримечательностях и общественном транспорте. Ситуационное моделирование на основе данных из IoT-датчиков позволяет службам быстро реагировать на чрезвычайные ситуации и эффективно планировать городские мероприятия.

В реставрации архитектурного наследия AR играет ключевую роль в документировании состояния объектов, визуализации скрытых элементов и планировании работ. С помощью AR реставраторы могут накладывать исторические чертежи и 3D-сканы на текущее состояние здания, что помогает выявить повреждения и деформации. Виртуальная реконструкция позволяет наглядно представить первоначальный облик разрушенных зданий, а пошаговые инструкции, отображаемые на рабочей поверхности, упрощают сложные реставрационные процессы.

Заключение

Технологии дополненной реальности становятся ключевым инструментом трансформации архитектурно-строительной отрасли. Интеграция AR с другими инновационными решениями, такими как искусственный интеллект и интернет вещей, открывает новые горизонты для проектирования, строительства и эксплуатации объектов. Это позволяет создавать более точные и детализированные модели, а также оптимизировать рабочие процессы и снижать затраты.

AR-технологии находят применение в различных аспектах архитектурной деятельности, начиная от визуализации проектов и заканчивая контролем строительства и техническим обслуживанием. Они значительно повышают точность проектирования, ускоряют процесс реализации проектов и улучшают коммуникацию между всеми участниками. В результате,

использование AR способствует повышению качества и эффективности работы в архитектурно-строительной сфере, делая её более конкурентоспособной и инновационной.

Литература

1. *Bimber O. Spatial Augmented Reality: Merging Real and Virtual Worlds* / O. Bimber, R. Ras-kar. – Natick : A K Peters, Ltd., 2005. – 392 c.
2. *Billinghurst M. A Survey of Augmented Reality* / M. Billinghurst, A. Clark, G. Lee. – Hanover : Now Publishers, Inc., 2015. – 218 c.
3. *Davila Delgado M. Augmented and Virtual Reality in Construction: Drivers and Limitations for Industry Adoption* / M. Davila Delgado, L. Oyedele, T. Beach, P. Demian // *Journal of Construction Engineering and Management* – 2020. – T. 146, № 7.

КЛАСТЕРИЗАЦИЯ НА ОСНОВЕ НЕЧЕТКИХ МЕТРИК

М. В. Оборотов, Т. М. Леденева

Воронежский государственный университет

Аннотация. В работе исследуется задача повышения эффективности кластеризации данных в условиях нечеткости и размытых границ между кластерами. Для этого предлагаются использование нечетких метрик, построенных на основе аддитивных генераторов и архимедовых треугольных норм. Рассмотрены теоретические основы аддитивных генераторов, треугольных норм и конорм, получены конкретные виды нечетких метрик. Проведен вычислительный эксперимент по нечеткой кластеризации синтетических данных с использованием алгоритма нечетких К-медоид и предложенных метрик. Результаты эксперимента демонстрируют качественное разбиение данных даже в условиях значительного перекрытия кластеров, что подтверждает перспективность предложенного подхода для анализа сложно структурированных данных.

Ключевые слова: нечеткая кластеризация, нечеткая метрика, аддитивный генератор, треугольная норма, треугольная конорма, архимедова норма, К-медоид, функция расстояния.

Введение

Одной из центральных задач анализа данных является разбиение множества объектов на группы схожих элементов. Традиционные методы кластеризации часто оказываются неэффективными, когда границы между кластерами размыты, а данные обладают высокой степенью неопределенности. В таких случаях более адекватными оказываются методы нечеткой кластеризации, позволяющие объекту принадлежать нескольким кластерам одновременно с различной степенью.

Особый интерес представляют методы, основанные на нечетких метриках, которые, в отличие от классических, отражают степень сходства между объектами. Использование нечетких метрик, порожденных некоторыми классами генераторов, позволяет адаптироваться к особенностям конкретной задачи.

Целью данной статьи является исследование возможности улучшения результатов кластеризации за счет применения нечетких метрик, построенных на основе аддитивных генераторов из класса рациональных функций. Для достижения этой цели в работе решаются следующие задачи: изложение общей теории аддитивных генераторов и треугольных норм; вывод конкретных формул для нечетких метрик и функций расстояния; проведение вычислительного эксперимента по кластеризации синтетических данных с использованием модифицированного алгоритма нечетких К-медоид.

1. Общая теория треугольных норм и их представление с помощью аддитивных генераторов

Основными операциями в нечеткой логике, обобщающими логические связки «И» и «ИЛИ», являются треугольные нормы (Т-нормы) и треугольные конормы (S-нормы). Рассмотрим соответствующие определения из [1].

Определение 1. Треугольной нормой (Т-нормой) называется операция $T : [0,1] \times [0,1] \rightarrow [0,1]$ удовлетворяющая условиям:

1. $T(x, y) = T(y, x);$
2. $T(T(x, y), z) = T(x, T(y, z));$

3. $T(0,0)=0, T(x,1)=T(1,x)=x;$
4. $(x \leq t) \wedge (y \leq z) \Rightarrow T(x,y) \leq T(t,z).$

Определение 2. Треугольной конормой (S-нормой) называется бинарная операция $S : [0,1] \times [0,1] \rightarrow [0,1]$, удовлетворяющая условиям коммутативности, ассоциативности, монотонности и граничному условию $S(1,1)=1, S(x,0)=S(0,x)=x$.

Важным классом Т-норм и S-норм являются архимедовы нормы, которые могут быть построены с помощью аддитивных генераторов, указанных в [1].

Утверждение 1. Операция $T : [0,1]^2 \rightarrow [0,1]$ является архимедовой Т-нормой тогда и только тогда, когда существует убывающий генератор $t(x)$, такой что

$$T(x,y)=t^{(-1)}(t(x)+t(y)). \quad (1)$$

Утверждение 2. Операция $S : [0,1]^2 \rightarrow [0,1]$ является архимедовой S-нормой тогда и только тогда, когда существует возрастающий генератор $s(x)$, такой что

$$S(x,y)=s^{(-1)}(s(x)+s(y)). \quad (2)$$

Определение 3. Убывающим генератором называется строго убывающая непрерывная функция $\varphi_{\downarrow} : [0,1] \rightarrow [0, \infty)$, такая что $\varphi_{\downarrow}(1)=0$.

Определение 4. Возрастающим генератором называется строго возрастающая непрерывная функция $\varphi_{\uparrow} : [0,1] \rightarrow [0, \infty)$, такая что $\varphi_{\uparrow}(0)=0$.

Также рассмотрим определение псевдообратного аддитивного генератора из [1].

Определение 5. Функция $\varphi_{\uparrow}^{(-1)} : [0, \infty) \rightarrow [0,1]$, такая, что

$$\varphi_{\uparrow}^{(-1)}(x)=\begin{cases} \varphi_{\uparrow}^{-1}(x), & \text{если } x \in [0, \varphi_{\uparrow}(1)], \\ 1 & \text{если } x \in (\varphi_{\uparrow}(1), \infty) \end{cases} \quad (3)$$

называется псевдообратной для $\varphi_{\uparrow}(x)$

Определение 6. Функция $\varphi_{\downarrow}^{(-1)} : [0, \infty) \rightarrow [0,1]$, такая, что

$$\varphi_{\downarrow}^{(-1)}(x)=\begin{cases} \varphi_{\downarrow}^{-1}(x), & \text{если } x \in [0, \varphi_{\downarrow}(0)], \\ 0 & \text{если } x \in (\varphi_{\downarrow}(0), \infty) \end{cases} \quad (4)$$

называется псевдообратной для $\varphi_{\downarrow}(x)$

Воспользуемся понятием нечеткой Т-метрики из [2].

Определение 7. Пусть U — произвольное множество, T — непрерывная треугольная норма, M — нечеткое множество на $U \times U \times (0, \infty)$ с функцией принадлежности, удовлетворяющей следующим свойствам для $\forall x, y, z \in U$ и $\forall u, v > 0$:

1. $\mu_M(x, y, u) \in (0,1]$;
2. $\mu_M(x, y, u)=1$ тогда и только тогда, когда $x=y$;
3. $\mu_M(x, y, u)=\mu_M(y, x, u)$;
4. $T(\mu_M(x, y, u), \mu_M(y, z, v)) \leq \mu_M(x, z, u+v)$;
5. $\mu_M(x, y, \cdot) : (0, \infty) \rightarrow [0,1]$ — непрерывная функция,

тогда μ_M называется нечеткой Т-метрикой.

Рассмотрим пару двойственных треугольных норм и конорм. Если R является отношением подобия, обладающим свойством ($\max - T$)-транзитивности, тогда, как указано в [1], можно построить \hat{R} — отношение различия, обладающим свойством ($\min - S$)-транзитивности, где $r_M(x, y)=1-\mu_M(x, y)$. В [6] предлагается представление нечеткой метрики.

Утверждение 3. Пусть U — произвольное множество, $d : U \times U \rightarrow [0, \infty)$ — метрика или псевдометрика на U , $\varphi : (0, \infty) \rightarrow (0, \infty)$ — возрастающая функция, T — непрерывная архимедова треугольная норма с аддитивным генератором t , тогда существует нечеткая метрика в форме нечеткого множества M с функцией принадлежности

$$\mu_M(x, y, u) = t^{(-1)} \left(\frac{d(x, y)}{\varphi(u)} \right).$$

2. Пример построения нечеткой метрики

Рассмотрим дробно-линейную функцию

$$\varphi_p(x) = \frac{x+p}{px+1} + C. \quad (5)$$

Воспользуемся условиями $\varphi_{\downarrow}(1)=0$, $\varphi_{\uparrow}(0)=0$ для нахождения констант:

$$\begin{aligned} \varphi_p(1) &= \frac{1+p}{p+1} + C = 1 + C = 0 \Rightarrow C_{\downarrow} = -1, \\ \varphi_p(0) &= \frac{p}{1} + C = p + C = 0 \Rightarrow C_{\uparrow} = -p. \end{aligned} \quad (6)$$

Для исследования свойства монотонности воспользуемся свойствами из [3]. Найдем производную функции φ :

$$\varphi'_p(x) = \frac{(1)(px+1) - (x+p)(p)}{(px+1)^2} = \frac{px+1 - px - p^2}{(px+1)^2} = \frac{1-p^2}{(px+1)^2}, \quad (7)$$

откуда получим, что $\varphi_p(x)$ строго возрастает при $p \in (-1, 1)$ и строго убывает при $p \in (-\infty, -1) \cup (1, +\infty)$. Учитывая непрерывность функции и область определения для убывающего генератора получим интервал для параметра $p \in (1, +\infty)$, так как $\varphi_{\downarrow, p}(0) = \frac{(1-p)(0-1)}{1+p0} = p-1$,

а неравенство $0 \leq p-1 \leq +\infty$ влечет неравенство $1 \leq p \leq +\infty$. В итоге получим на основе определений 3 и 4 убывающие генераторы

$$\begin{aligned} \varphi_{\downarrow, p}(x) &= \frac{(1-p)(x-1)}{1+px}, \quad p \in (1, +\infty), \\ \varphi_{\downarrow, +\infty}(x) &= \frac{1-x}{x}, \end{aligned} \quad (8)$$

и возрастающий генератор

$$\varphi_{\downarrow, p}(0) = \frac{(1-p)(0-1)}{1+p0} = p-1. \quad (9)$$

Найдем соответствующие псевдообратные функции

$$\varphi_{\downarrow, p}^{(-1)}(x) = \begin{cases} \frac{1-p+x}{1-p-px}, & \text{если } x \in [0, p-1], \\ 0, & \text{если } x \in (p-1, \infty); \end{cases} \quad (10)$$

$$\begin{aligned} \varphi_{\downarrow, +\infty}^{(-1)}(x) &= \frac{1}{1+x}; \\ \varphi_{\uparrow, p}^{(-1)}(x) &= \begin{cases} \frac{x}{1-p^2-px}, & \text{если } x \in [0, 1-p], \\ 1, & \text{если } x \in (1-p, \infty). \end{cases} \end{aligned} \quad (11)$$

Рассмотрим графики генераторов (8)–(11) в $[0, 1] \times [0, 1]$.

Если аддитивные генераторы известны, то на основе утверждений 1 и 2 можно построить соответствующие им нечеткие операции — архимедовы S-конормы и T-нормы. В данном случае они имеют следующий вид:

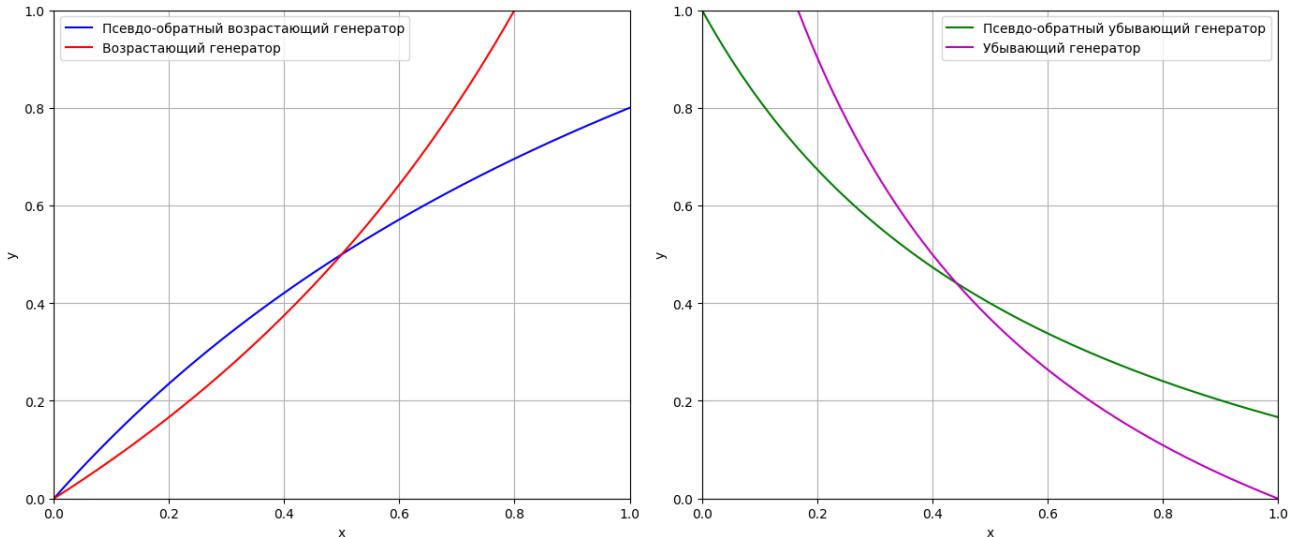


Рис. 1. Графики аддитивных генераторов и их псевдообратных функций (левый график при $p=-0.5$ для возрастающего, правый график при $p=2.75$ для убывающего)

$$\begin{aligned}
 S_p(x, y) &= \frac{x + y + 2pxy}{1 - p^2xy}, \quad p \in (-1, 1); \\
 T_{+\infty}(x, y) &= \frac{xy}{x + y - xy}; \\
 T_p(x, y) &= \frac{x + y - 1 + (p^2 + 2p)xy}{1 + 2p - p^2xy + p^2(x + y)}, \quad p \in (1, +\infty).
 \end{aligned} \tag{12}$$

Воспользовавшись примерами аддитивных генераторов, соответствующих архимедовым Т-нормам, найдем примеры нечетких метрик, общая формула которых представлена в утверждении 3.

$$\mu_{M,p}(x, y, u) = \begin{cases} 1 - p + \frac{d(x, y)}{\varphi(u)}, & \text{если } \frac{d(x, y)}{\varphi(u)} \in [0, p-1], \\ 1 - p - p \frac{d(x, y)}{\varphi(u)}, & \text{если } \frac{d(x, y)}{\varphi(u)} \in (p-1, \infty). \end{cases} \tag{13}$$

$$\mu_{M,+\infty}(x, y, u) = \frac{1}{1 + \frac{d(x, y)}{\varphi(u)}}.$$

На основе данных метрик получим функции расстояния

$$\mu_{M,p}(x, y, u) = \begin{cases} \frac{(1+p)d(x, y)}{p(d(x, y) + \varphi(u)) - \varphi(u)}, & \text{если } \frac{d(x, y)}{\varphi(u)} \in [0, p-1], \\ 1, & \text{если } \frac{d(x, y)}{\varphi(u)} \in (p-1, \infty), \end{cases} \tag{14}$$

$$\mu_{M,+\infty}(x, y, u) = \frac{d(x, y)}{\varphi(u) + d(x, y)}.$$

3. Результаты вычислительного эксперимента

Пусть задано множество объектов $X = \{x_1, \dots, x_n\}$, где каждый объект x_i описывается вектором признаков $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$. Для оценки близости объектов используется функция расстояния $d(x_i, x_j)$. Найдем разбиение множества объектов X на группы (классы, кластеры) близких объектов в смысле расстояния $d(x_i, x_j)$, при этом искомое разбиение должно удовлетворять следующим свойствам:

- 1) $C_i \neq \emptyset$ (каждый кластер не пуст);
- 2) $C_i \cap C_j = \emptyset$ при $i \neq j$ (кластеры не пересекаются в случае жёсткой кластеризации);
- 3) $\bigcup_{i=1}^k C_i = X$ (все объекты принадлежат какому-то кластеру).

Для тестирования метрик воспользуемся алгоритмом нечетких К-медоид из [5], так как данный алгоритм позволяет использовать произвольную метрику. Рассмотрим метрику с расстоянием Евклида $d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$ и $\varphi(u) = u$. Протестируем алгоритм на синтетических данных, чтобы установить влияние метрики на возможность выделить пересекающиеся в разной степени кластеры.

В ходе визуализации полученных метрик были выделены несколько отличительных графиков. Так, поверхность, изображенная на рис. 2, можно использовать для оценки индикации сходства объектов, а поверхность, рассматриваемая на рис. 3, соответствует задаче, когда оценка различий между наблюдениями основывается на позиции оптимизма.

Проведенный вычислительный эксперимент позволил оценить эффективность предложенного алгоритма нечеткой кластеризации на двух типах синтетических данных. В ходе визуального анализа результата обработки выборки с тремя кластерами, характеризующимися значительным перекрытием дисперсий, представленный на рис. 4, подтверждает, что, несмотря на высокий уровень неопределенности в зонах перекрытия, алгоритм обеспечил четкое разделение кластеров, что свидетельствует о его устойчивости к шуму. Кроме того, итоговое разделение, отображенное на рис. 5, выборки, состоящей из пяти хорошо разделимых класте-

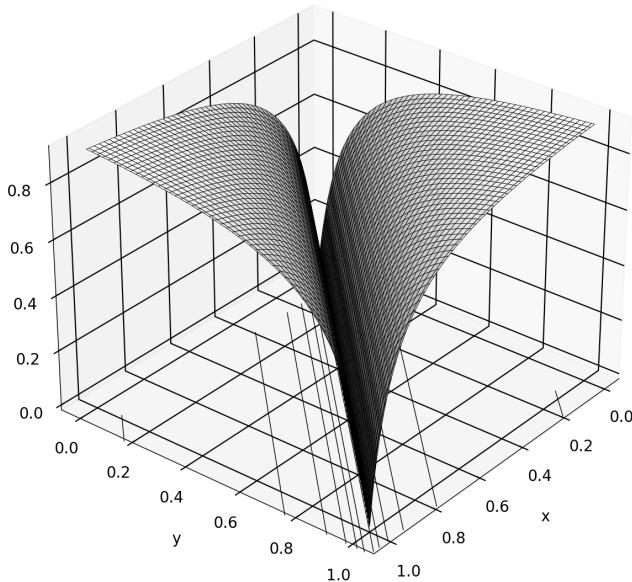


Рис. 2. Графики функции расстояния $\mu_{M,p}(x, y, u)$ при фиксированном значении параметра p и $\varphi(u) = u$ фиксированная с использованием евклидова расстояния

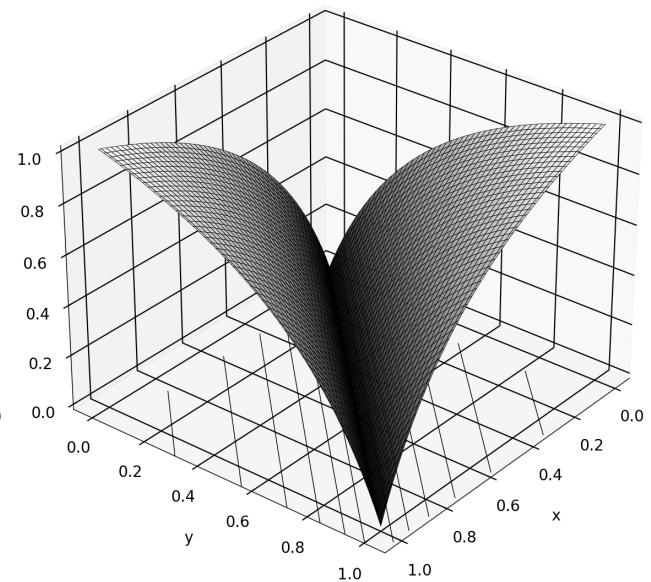


Рис. 3. Графики функции расстояния $\mu_{M,+infinity}(x, y, u)$ при $\varphi(u) = u$ фиксированная с использованием евклидова расстояния. используя евклидова расстояния

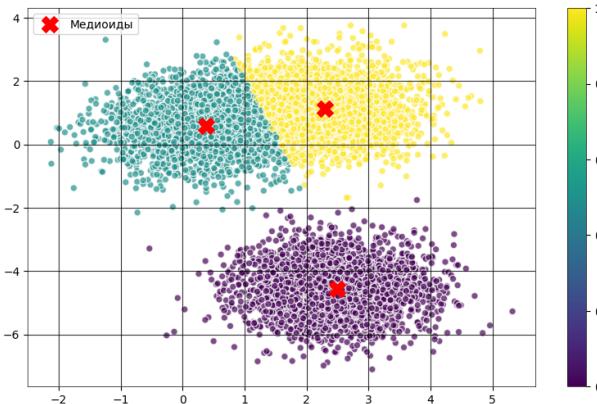


Рис. 4. Результат кластеризации данных, состоящих из выборки по 7000 элементов, 3 кластеров с перекрывающейся дисперсией

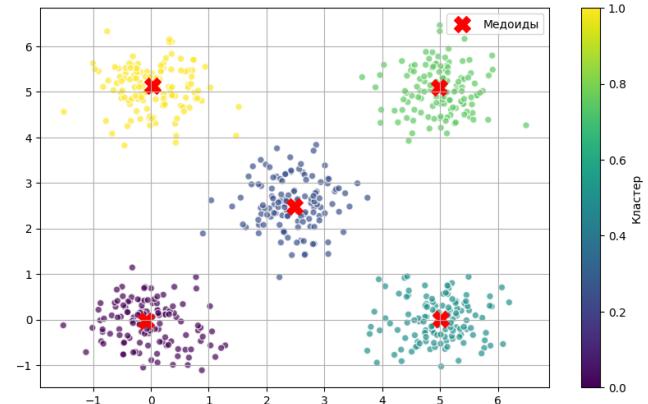


Рис. 5. Результат кластеризации данных, состоящих из выборки по 750 элементов, 5 кластеров с неперекрывающейся дисперсией

ров, демонстрирует, что алгоритм точно идентифицирует все кластеры с явными границами. Таким образом, ключевым преимуществом алгоритма является его способность к построению адекватных границ для данных с четкой кластерной структурой, так и в более сложных случаях со значительным перекрытием кластеров.

Заключение

Проведённое исследование подтверждает, что использование аддитивных генераторов треугольных норм и конорм открывает новые возможности для повышения эффективности кластеризации в условиях нечёткости. Предложенные методы позволяют достичь качественного разбиения данных даже при размытых границах между кластерами, что демонстрирует их значительный практический потенциал. Результаты работы создают основу для дальнейших исследований в области нечёткого анализа данных. Перспективными направлениями являются применение этих методов для работы с реальными данными, а также их адаптация для более сложных задач.

Литература

1. Леденева Т. М. Основы нечеткого моделирования : учебно-методическое пособие / Т. М. Леденева ; Воронежский государственный университет. – Воронеж : Издательский дом ВГУ, 2022. – 129 с.
2. Леденева Т. М. Нечеткие метрики на основе генераторов архимедовых треугольных норм из класса рациональных функций / Т. М. Леденева, Т. А. Моисеева // Искусственный интеллект и принятие решений. – 2024. – № 1. – С. 30–43.
3. Протасов Ю. М. Математический анализ : учебное пособие / Ю. М. Протасов. – 3-е изд., стер. – Москва : Флинта, 2024. – 165 с. – URL: <https://www.ibooks.ru/bookshelf/25511/reading> (дата обращения: 15.07.2024).
4. Воронцов К. В. Математические методы обучения по прецедентам : лекции МФТИ / К. В. Воронцов. – Москва, 2013. – 185 с.
5. Pinheiro D. N. Convex fuzzy k-medoids clustering / D. N. Pinheiro, D. Aloise, S. J. Blanchard // Fuzzy Sets and Systems. – 2020. – Vol. 389. – P. 66–92.
6. Grigorenko O. T. Two new methods to construct fuzzy metrics from metrics / O. T. Grigorenko, J.-J. Mifiana, O. Valero // Fuzzy Sets and Systems. – 2023. – Vol. 467. – P. 108483.

ШИФРОВАНИЕ ПО ТИПУ «ТЕКСТ В ТЕКСТЕ» С ИСПОЛЬЗОВАНИЕМ СПИРАЛЬНО ГЕНЕРИРУЕМЫХ ПЕРЕСТАНОВОК

И. Н. Попов

Северный (Арктический) федеральный университет имени М. В. Ломоносова

Аннотация. Под шифрованием по типу «текст в тексте» понимается скрытие сообщения внутри некоторого текста. При шифровании по исходному сообщению строится текст, содержащий это сообщение, при дешифровании — по тексту определяется сообщение. Один из подходов реализации такого шифра основывается на использовании части перестановки: буквы текста, номера которых совпадают с числами из выбранной части перестановки, совпадают с буквами сообщения. В работе рассматриваются алгоритмы для спиральной генерации перестановок и шифрования по типу «текст в тексте» с указанием необходимой секретной информации — ключей.

Ключевые слова: алгоритм, криптография, шифрование, дешифрование, ключ, шифрование по типу «текст в тексте», текст, перестановки, генерация перестановок, программирование.

Введение

Перестановка — один из объектов математики [1, 2], который используется в криптографии как науки о способах шифрования, и представляет собой биективное отображение начального отрезка натурального ряда на себя. Перестановки используются в разных вариациях. Классическим примером использования перестановок в криптографии является шифр, который называется перестановочным шифром: буквы исходного текста переставляются согласно выбранной перестановке, и тем самым, например, слово «информация» с помощью перестановки 9, 8, 3, 5, 4, 6, 7, 1, 2, 10 шифруется словом «цифромания». С помощью перестановки можно в определённый текст «вписать» нужный секретный текст — сообщение. Такой способ скрытия сообщения называется шифрованием по типу «текст в тексте». При любом способе шифрования используются ключи. Под ключами понимается информация (чаще всего, секретная), которая используется при шифровании и дешифровании; с одной стороны, не знание ключей приводит к неправильному результату шифрования и, с другой стороны, затрудняет процесс дешифрования. Исходными данными для рассматриваемого в работе способа шифрования по типу «текст в тексте» являются само сообщение, длина искомого текста и перестановка с некоторой информацией о ней. Результатом шифрования является текст, содержащий в себе сообщение. Раз речь идёт о текстах, то в первую очередь нужно определить алфавит, с порядком расположением в нём букв и их количеством; под текстом понимается конечный набор букв алфавита, его длина — это количество букв, из которых он состоит. При шифровании по типу «текст в тексте» буквы сообщения располагают в тексте не по порядку, а, наоборот, стремятся расположить их в хаотичном порядке. В большинстве случаев при шифровании искомый текст будет генерироваться случайным образом как набор случайно выбираемых букв алфавита (в этом случае говорят о бессмысленном тексте); при дешифровании текст может быть, наоборот, выбран смысловой.

Алгоритмы шифрования и дешифрования по типу «текст в тексте», рассматриваемые в работе, строятся с использованием перестановки. Учитывая то, что генерируемые тексты могут иметь большую длину, вопрос о передаче перестановки тем, кому предназначено сообщение, становится весьма непростым. Поэтому продвигается идея о том, чтобы не передавать «тем, кому нужно» всю перестановку в целом, а уметь «ими» её генерировать по некоторому алгоритму, исходя из переданных (или имеющихся) данных.

В работе предлагаются два алгоритма: один из них касается генерации перестановки, второй — использования перестановки для шифрования и дешифрования сообщения по типу «текст в тексте».

1. Теоретические аспекты

Алгоритм спиральной генерации перестановки

Алгоритм генерации перестановки следующий. Исходные данные: начальный отрезок натурального ряда и окружность с отмеченными точками, количество которых не превышает количеству чисел в выбранном отрезке. Около точек окружности по спирали по часовой стрелке расставляются числа из отрезка, беря их один за другим и начиная с 1, и тем самым у некоторых точек появляются списки чисел, упорядоченные по возрастанию. Выбирается число, называемое стартовым, которое считается первым числом перестановки. Определяется правило перехода от одного числа к другому на окружности: стартовое число удаляется с окружности и, если стартовое число чётное, то по часовой стрелке на окружности отсчитываются точки, количество которых равно стартовому числу, иначе отсчёт ведётся против часовой стрелки. Полученное число объявляется новым стартовым числом и вторым числом генерируемой перестановки. Аналогичным образом продолжается построение искомой перестановки. Отметим, что стартовым числом из списка назначается самое меньшее в нём. По ходу выбора нового стартового числа некоторые списки становятся пустыми, и они удаляются с окружности вместе с точками.

Данный алгоритм называется спиральной генерацией перестановки. Прилагательное «спиральное» подчёркивает, что отрезок чисел, сопоставляемых точкам, «накручивается» по спирали на окружность в несколько рядов; если количество точек на окружности совпадает с числом выбираемых чисел, то спирального накручивания не происходит и каждой точке сопоставляется только одно число, включённое в список.

Пример. На окружности отмечены 7 точек и начальный отрезок содержит первые 12 натуральных чисел. Тогда списки чисел, расставленные у точек на окружности, имеют вид:

[1; 8], [2; 9], [3; 10], [4; 11], [5; 12], [6], [7],

считаем, что списки расположены на окружности по часовой стрелке и после списка [7] идёт список [1; 8]; такой вид назовём строчной записью списков. Список, в котором содержится стартовое число, будем называть стартовым. Если стартовое число является нечётным, то отсчитывает справа налево нужное количество списков, начиная со списка, идущего сразу же за стартовым; если же стартовое число равно чётному числу, то отсчёт списков ведётся слева направо. Например, если стартовое число равно 9, то из стартового списка [2; 9] удаляется это само число и отсчитывается 9 списков справа налево, начиная со списка [1; 8]:

[1; 8]	[2]	[3; 10]	[4; 11]	[5; 12]	[6]	[7]
← 1; 8	← 7	← 6	← 5	← 4	← 3	← 2; 9

Направление выбора списков и их отсчёт

В итоге попадаем в список [7]. Новое стартовое число равно 7. Так как в списке находится одно это число, то с удалением этого числа удаляется и весь список. Учитывая нечётность числа 7, отсчёт ведётся снова справа налево, начиная со списка [6]:

[1; 8]	[2]	[3; 10]	[4; 11]	[5; 12]	[6]
← 6	← 5	← 4	← 3	← 2	← 1; 7

Направление выбора списков и их отсчёт

Обратим внимание, что один и тот же список может быть первым и последним в отсчёте (в данном случае отсчёт начали со списка [6] и на нём же закончили). Новым стартовым числом объявляется число 6. При этом получаем первые три числа перестановки: 9, 7, 6.

При первом стартовом числе 9 процесс генерации перестановки выглядит следующим образом:

[1; 8], [2; 9], [3; 10], [4; 11], [5; 12], [6], [7]	9
[1; 8], [2], [3; 10], [4; 11], [5; 12], [6], [7]	9, 7
[1; 8], [2], [3; 10], [4; 11], [5; 12], [6]	9, 7, 6
[1; 8], [2], [3; 10], [4; 11], [5; 12]	9, 7, 6, 1
[8], [2], [3; 10], [4; 11], [5; 12]	9, 7, 6, 1, 5
[8], [2], [3; 10], [4; 11], [12]	9, 7, 6, 1, 5, 12
[8], [2], [3; 10], [4; 11]	9, 7, 6, 1, 5, 12, 4
[8], [2], [3; 10], [11]	9, 7, 6, 1, 5, 12, 4, 11
[8], [2], [3; 10]	9, 7, 6, 1, 5, 12, 4, 11, 2
[8], [3; 10]	9, 7, 6, 1, 5, 12, 4, 11, 2, 8
[3; 10]	9, 7, 6, 1, 5, 12, 4, 11, 2, 8, 3
[10]	9, 7, 6, 1, 5, 12, 4, 11, 2, 8, 3, 10

Итак, спирально сгенерирована перестановка: 9, 7, 6, 1, 5, 12, 4, 11, 2, 8, 3, 10. ■

Отметим, что правило перехода от списка к списку может быть изменено. Например, можно совершать движения на окружности от точки к точке (от списка к списку) только в одном направлении, скажем, по часовой стрелке (или для строчной записи списков слева направо); число отсчитываемых точек (списков) на окружности может вычисляться по определённой формуле (в качестве формулы можно выбрать функцию Эйлера, значение которой от натурального числа равно количеству натуральных чисел, не превосходящих данного и взаимно простых с ним, тогда в этом случае получили бы перестановку 9, 3, 1, 7, 8, 5, 6, 10, 4, 2, 11, 12).

Алгоритмы шифрования и дешифрования сообщения по типу «текст в тексте»

Алгоритм использования перестановки при способе шифрования по типу «текст в тексте» следующий. По сообщению нужно построить текст, в котором оно содержится. В скобках будем делать пометки о ключах. Вначале определяется алфавит как набор букв, из которых строятся тексты и сообщения (сам алфавит с расположением букв и их количеством является ключом). Заранее определяется длина искомого текста (ключ). Генерируется или выбирается перестановка (ключ), длина которой равна длине искомого текста. Выбирается часть перестановки, длина которой равна длине сообщения (здесь ключами являются начало выбираемой части и длина сообщения). Искомый текст строится в три приёма: сначала составляется заготовка будущего текста, состоящая из одного и того же символа, не входящего в алфавит; затем символы в заготовке с номерами из выбранной части перестановки заменяются буквами сообщения, беря их одну за другой; после этого оставшиеся символы заготовки (случайным образом) заполняются буквами алфавита. В результате получается текст, содержащий в себе сообщение.

Для дешифрования сообщения по данному тексту нужно знать: алфавит; перестановку; длину сообщения; начальная позиция выбираемой части перестановки.

2. Практическая реализация

Продемонстрируем использование алгоритмов шифрования и дешифрования по типу «текст в тексте» с применением спирально сгенерированной перестановки.

Пример. Алфавит (_ — пробел): абвгдеёжэйклмнопрстуфхцчшыъэюя_

Сообщение: буква; длина сообщения — 5; длина искомого текста — 12; перестановка длины 12: [9, 7, 6, 1, 5, 12, 4, 11, 2, 8, 3, 10]. В перестановке выбирается часть длины 5: [6, 1, 5, 12, 4]. Составляется заготовка будущего текста (буквам сообщения «буква», беря их слева направо, в заготовке присвоены номера, согласно числам из выбранной части перестановки: буква «б» записана на 6-м месте в заготовке, буква «у» — на 1-м месте и так далее):

6	1	5	12	4
6	у	к	в	а

→

1	2	3	4	5	6	7	8	9	10	11	12
у	*	*	а	к	б	*	*	*	*	*	в

Случайным образом подбираются буквы на вакантные места в заготовке, отмеченные в ней символом «*». В итоге получается искомый текст: «урцакблъюзцв».

Если даже знать, что в сообщении 5 букв и оно имеет смысл (а не в противовес — простой набор букв), то из букв предложенного текста можно составить 226 слов с 5-ю буквами, часто встречающиеся в речи или редкие и специфические; среди них: образ, выбор, крыло, буква, забор, рукав, бокал, кузов, рыбак, злоба, рубка, рыбка, обрыв, рыбок, арбуз, лавры, булка, обвал, колба, кварт, кобра, бурка, вокал, зубок, вобла, обуза, блузка, зарок, лубок, балык, браво, крабы, узоры. Всего из букв текста «урцакблъюзцв» можно составить 832 смысловых слова, содержащих от 2 («ау») до 8 букв («зубровка»). ■

Пример. Алфавит (_ — пробел): абвгдеёжэйклмнопрстуфхцчшыъэюя_

Текст длины 10: неосязаемо; длина искомого сообщения: 3.

Сpirальная генерация перестановки [5, 6, 2, 4, 10, 1, 3, 9, 8, 7] длины 10 со стартовым числом 5 с 6-ю точками на окружности (или, что то же самое, с 6-ю списками):

[1; 7], [2; 8], [3; 9], [4; 10], [5], [6]
[1; 7], [2; 8], [3; 9], [4; 10], [6]
[1; 7], [2; 8], [3; 9], [4; 10]
[1; 7], [8], [3; 9], [4; 10]
[1; 7], [8], [3; 9], [10]
[1; 7], [8], [3; 9]
[7], [8], [3; 9]
[7], [8], [9]
[7], [8]
[7]

Выбираемая часть с позиции 4: [4, 10, 1] (так как длина искомого сообщения известна, то указываем только позицию, с которой начинается выбираемая часть перестановки).

По известным данных (текст, длина сообщения, информация о генерации перестановки, позиция выбора чисел из перестановки) можно дешифровать искомое сообщение:

1	2	3	4	5	6	7	8	9	10	→	4	10	1
н	е	о	с	я	з	а	е	м	о		с	о	н

Отбираем буквы с номерами 4, 10, 1. Искомое сообщение: сон.

Количество слов со смыслом длины 3, составляемые из букв слова «неосязаемо», равно 104. Среди них: она, сам, оно, сон, нос, яма, зам, ася, оса, сом, аня, яна. ■

Вообще говоря, сообщение может быть бессмысленным или быть сокращением слов (например, сен — сентябрь, ноя — ноябрь) или аббревиатурой (например, гто, азс и оон). У бессмысленных сообщений есть и психологическая значимость: по большей мере из текста пытаются выцепить сообщения именно со смыслом, а бессмысленные откидываются как не возможные. Важно также отметить относительность в криптографии понятия «бессмысленность»: то, что для одних бессмысленно, для других может иметь большую смысловую нагрузку.

Заключение

Предложенные алгоритмы спиральной генерации перестановки и шифрования и дешифрования по типу «текст в тексте» могут быть реализованы в виде компьютерных программ, например, с использованием Python [3]. Перестановки в разных вариациях используются при шифровании [4, 5]. Отличительной особенностью предлагаемого использования перестановок при шифровании является то, что берётся не вся перестановка, а только часть, что и позволяет включать сообщения в текст или, наоборот, определить сообщение по тексту при дешифровании.

Литература

1. Калужнин Л. А. Преобразования и перестановки / Л. А. Калужнин, В. И. Сущанский. – Москва : Наука. Главная редакция физико-математической литературы, 1985. – 160 с.
2. Шень А. Перестановки / А. Шень. – Москва : МЦНМО, 2022. – 40 с.
3. Лутц М. Изучаем Python / М. Лутц. – Санкт-Петербург : Символ-Плюс, 2011. – 1280 с.
4. Введение в криптографию / Под общей ред. В. В. Ященко. – Санкт-Петербург : Питер, 2001. – 288 с.
5. Кан Д. Взломщики кодов / Д. Кан. – Москва : Центрполиграф, 2000. – 471 с.

ПРОЕКТИРОВАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ ДАННЫМИ ДЛЯ ПРЕДИКТИВНОЙ КЛАССИФИКАЦИИ РАДИОСИГНАЛОВ НА ПРИМЕРЕ СИГНАЛОВ БПЛА

И. А. Ремезов

МИРЭА – Российский технологический университет

Аннотация. В статье рассматривается актуальная задача классификации беспилотных летательных аппаратов (БПЛА) по их радиоизлучению с использованием методов предиктивной аналитики. Предложен подход, сочетающий высокую точность классификации с возможностью последующей реализации на программируемых логических интегральных схемах (ПЛИС). Разработана архитектура системы управления данными, ориентированная на обработку сигналов в режиме, близком к реальному времени. Проведено сравнение различных моделей, включая сверточные и рекуррентные нейронные сети, а также ансамблевый метод. Показано, что наилучшие результаты достигаются при использовании ансамблирования. Особое внимание уделено анализу зависимости качества классификации от уровня отношения сигнал-шум.

Ключевые слова: беспилотные летательные аппараты (БПЛА), классификация радиосигналов, предиктивная аналитика, система управления данными, машинное обучение, ансамбли моделей, обработка сигналов в реальном времени, большие данные, Apache Spark, Apache Kafka, отношение сигнал-шум (SNR), радиоэлектронное противодействие, программируемые логические интегральные схемы (ПЛИС), I/Q сигналы, спектрограммы.

Введение

Широкое распространение беспилотных летательных аппаратов (БПЛА) в военной и гражданской сферах обуславливает актуальность задач их обнаружения, идентификации инейтрализации [1]. Одним из перспективных направлений является перехват управления БПЛА, который требует точного определения модели аппарата и используемых протоколов связи по его радиоизлучению [2]. Традиционные методы противодействия, такие как радиоэлектронное подавление или физическое воздействие, зачастую приводят к уничтожению цели, что не позволяет решить задачу ее захвата для последующего изучения.

Создание интеллектуальных систем, способных в реальном времени собирать данные, анализировать радиоэфир и классифицировать цели, требует применения современных методов машинного обучения и проектирования систем управления данными. Особую сложность представляет развертывание таких систем на аппаратных платформах с ограниченными вычислительными ресурсами и энергопотреблением, что характерно для полевых условий.

В данной работе предлагается подход к построению системы классификации БПЛА, сочетающий высокую точность с учетом требований к последующей реализации на программируемых логических интегральных схемах (ПЛИС).

1. Обзор

Обработка сигналов БПЛА сопряжена с работой в условиях больших объемов данных, высокой частоты дискретизации и жестких временных ограничений, обусловленных необходимостью принятия решений в реальном времени. Это предъявляет особые требования к вычислительным архитектурам систем радиоэлектронного противодействия. ПЛИС являются предпочтительной платформой для развертывания подобных алгоритмов благодаря их способности к параллельным вычислениям, низкому энергопотреблению и детерминированности.

Данное обстоятельство влечет за собой необходимость разработки и использования оптимизированных, легковесных архитектур вычислительных моделей, ориентированных на эффективную аппаратную реализацию. Это включает в себя минимизацию количества обучаемых параметров модели, использование операций, эффективно реализуемых на ПЛИС (таких как свертки), и отказ от ресурсоемких вычислений. Таким образом, задача классификации БПЛА рассматривается не только с точки зрения достижения максимальной точности, но и с учетом ограничений целевой аппаратной платформы.

Важнейшим этапами при разработке подобных систем является моделирование и оценка проектируемых систем. Для решения данных задач перспективным представляется создание системы управления данными с использованием инструментария big data.

1. Проектирование системы управления данными

Рассмотрим существующие решения. На практике в процессе цифровой обработки сигналов можно выделить несколько ключевых этапов:

- Прием данных с АЦП. АЦП преобразует непрерывный аналоговый сигнал в дискретные цифровые отсчеты. В процессе сигнал дискретизируется, квантуется и кодируется;
- Преобразование данных. После получения цифровых данных выполняется их обработка с помощью алгоритмов ЦОС. К основным сценариям относят фильтрацию, модуляцию, демодуляцию, спектральный анализ (быстрое преобразование Фурье) и обнаружение и распознавание сигналов;
- Передача данных. После обработки цифровые данные могут передаваться в другие системы или сохраняться;

Рассмотрим пример модели такой системы в нотации IDEF0 (рис. 1) и ее декомпозицию (рис. 2).

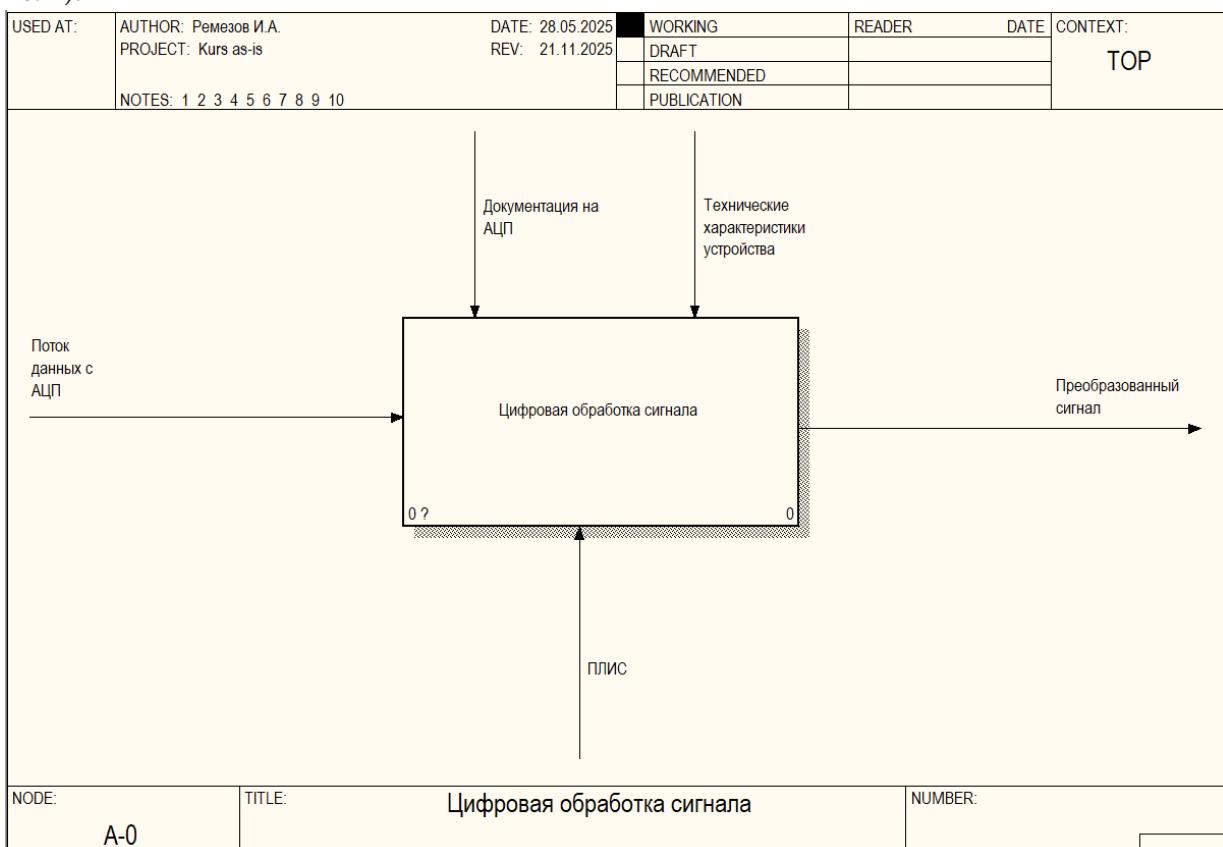


Рис. 1. Контекстная диаграмма типовой системы обработки радиосигналов

Отметим, что в качестве исполнителя в данном случае выступает ПЛИС.

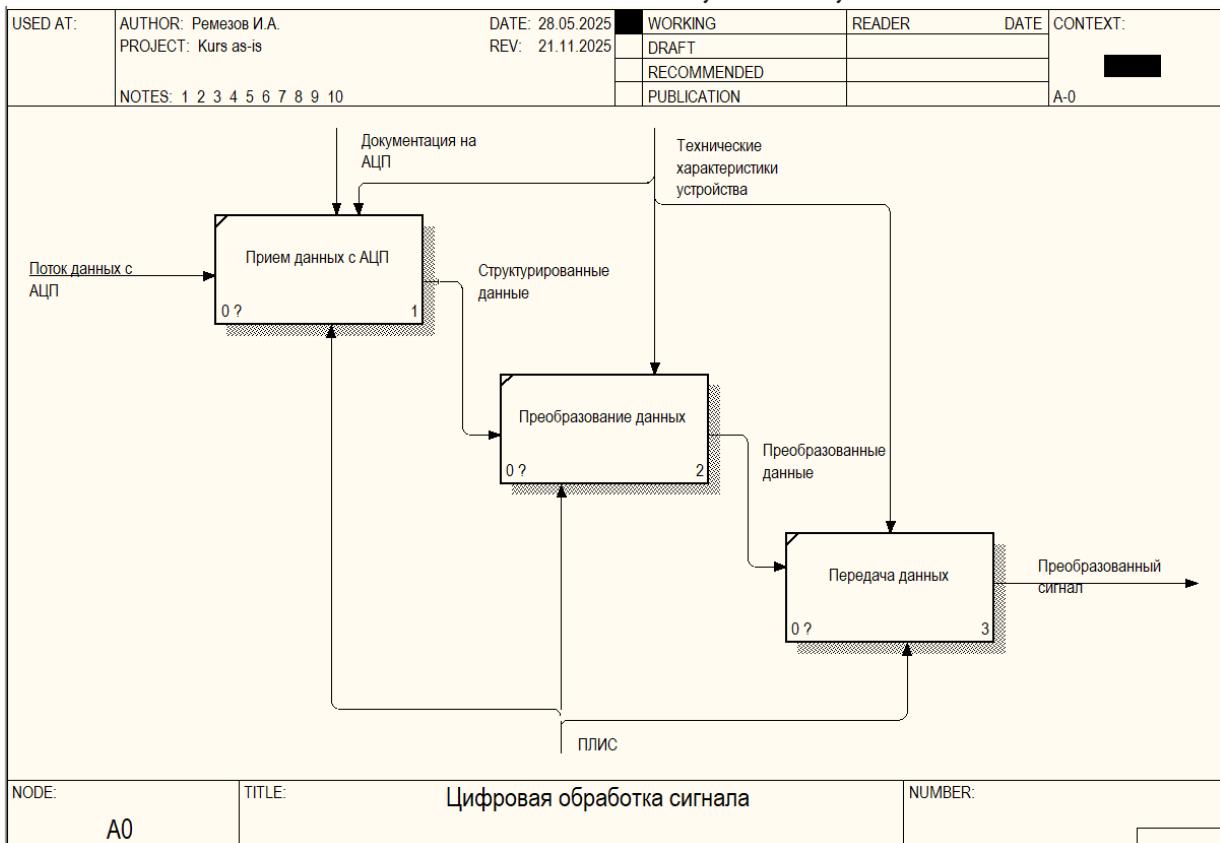


Рис. 2. Диаграмма декомпозиции типовой системы обработки радиосигналов

Стоит упомянуть, что в настоящее время активно развиваются программно-аппаратные комплексы решающие похожие задачи [3].

Рассмотрим архитектуру системы, реализующей минимально жизнеспособный прототип (MVP) конвейера по захвату, передаче, хранению и управлению радиосигналом в режиме, приближенном к реальному времени. Архитектура построена с акцентом на асинхронность, отказоустойчивость и масштабируемость, а также с использованием исключительно open-source решений.

Предлагаемый конвейер решает аналогичные задачи, однако в отличие от существующих решений в конвейере другие механизмы, а именно:

- Flume. Обеспечивает процесс приема данных с АЦП, собирает файлы, генерируемые АЦП для дальнейшей передачи в Kafka;
- Kafka. Принимает данные из топика flume и передает их в spark на обработку;
- Spark. Обеспечивает преобразование и передачу данных;

С учётом требований к высокой пропускной способности, отказоустойчивости и возможности масштабирования, в рамках данной работы была спроектирована архитектура конвейера обработки радиосигналов, ориентированная на обработку в режиме, близком к реальному времени. Модель архитектуры представлена на (рис. 3.).

Данная архитектура обладает рядом преимуществ:

- Горизонтальное масштабирование: все компоненты могут масштабироваться независимо;
- Отказоустойчивость: при сбое одного узла Kafka — данные не теряются;
- Асинхронность: захват, передача и сохранение радиосигналов работают независимо друг от друга;
- Open-source: все инструменты доступны бесплатно, без лицензирования;
- Гибкость: можно подключить аналитику или AI-модули для событийной фильтрации;

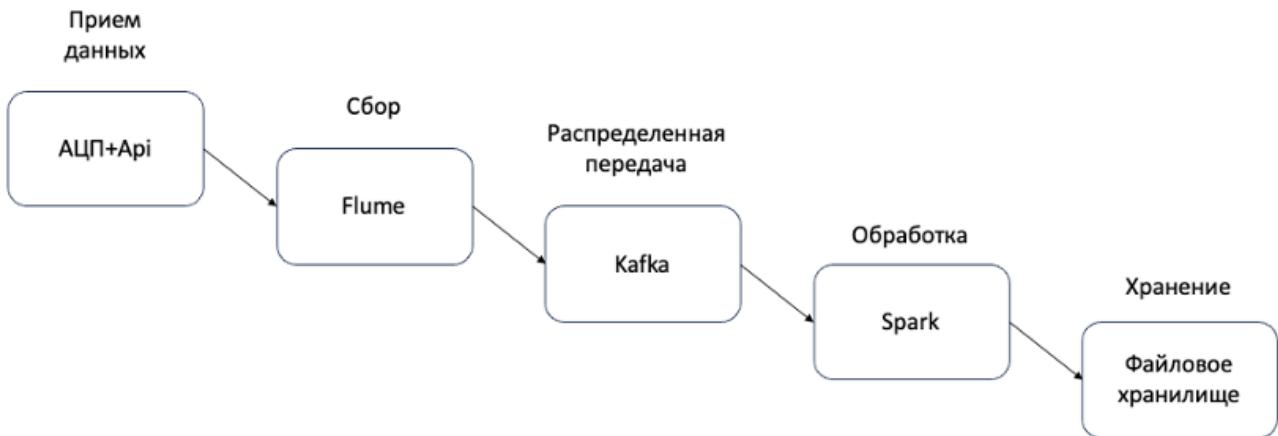


Рис. 3. Архитектура конвейера

Одной из основных частей подобной системы является предиктивная система, позволяющая в реальном времени генерировать предсказания для повышения качества работы. Рассмотрим процесс создания данной системы.

2. Предобработка

2.1. Обзор данных

В работе использовался общедоступный набор данных, предназначенный для разработки методов обнаружения и классификации сигналов БПЛА [4]. Набор содержит немодальные сигналы шести моделей БПЛА (DJI, FutabaT14, FutabaT7, Graupner, Taranis, Turnigy) и отдельный класс шума. Каждый сигнал представлен вектором синфазной (I) и квадратурной (Q) компонент длиной 16384 отсчета, что соответствует приблизительно 1.2 мс при частоте дискретизации 14 МГц.

После нормализации сигналы БПЛА были смешаны с лабораторными шумами (Bluetooth, Wi-Fi, усилитель) или гауссовым шумом с равной вероятностью (50/50). Класс шума был сформирован путем смешивания лабораторного и гауссова шума во всех возможных комбинациях. Для всех классов, включая шум, количество примеров для каждого уровня отношения сигнал-шум (SNR) в диапазоне от -20 дБ до 30 дБ с шагом 2 дБ распределено равномерно, по 3792–3800 образцов на каждый уровень SNR.

Анализ распределения данных выявил существенную несбалансированность классов. Наибольшее количество примеров приходится на класс 4, а наименьшее — на класс 0. (рис. 4.) При этом для каждого класса распределение по уровням SNR является однородным, что делает набор репрезентативным для оценки устойчивости моделей к шуму.

2.1. Предобработка данных

Исходный набор данных был разделен на обучающую и тестовую выборки в соотношении 80/20. Разбиение осуществлялось стратифицированно, с сохранением распределения примеров по классам и уровням SNR в обеих выборках, что обеспечило репрезентативность оценки моделей.

Для устранения дисбаланса классов, выявленного на этапе анализа данных, к обучающей выборке применена аугментация методом циклического временного сдвига (cyclic time shifting). Для каждого исходного сигнала генерировалось несколько его циклически сдвинутых копий. Это позволило искусственно увеличить количество примеров в миноритарных классах без искажения фундаментальных статистических свойств сигналов.

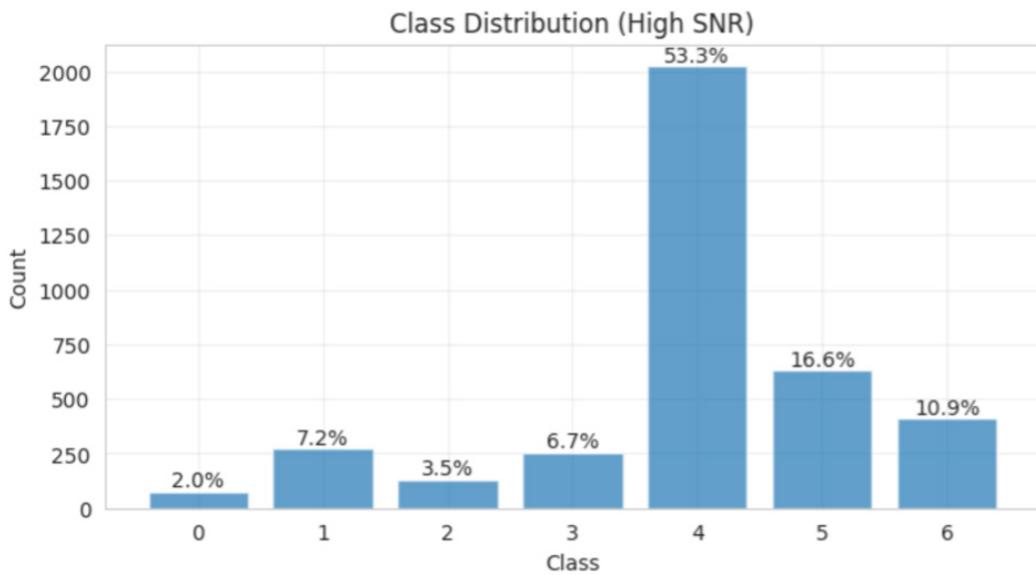


Рис. 4. Дисбаланс классов

Все данные прошли Z-score нормализацию. Для I/Q-сигналов нормализация проводилась по каждому каналу независимо. Спектрограммы также нормализовались.

3. Обучение моделей

3.1. CNN

Для классификации по спектрограммам использовалась сверточная нейронная сеть (CNN). Спектрограммы подавались на вход сети в виде одноканальных изображений. В ходе эксперимента модель показала значение F1-меры, равное 0.66, что указывает на умеренную эффективность спектрограмм как единственного источника информации для данной задачи. График обучения представлен на (рис. 5.)

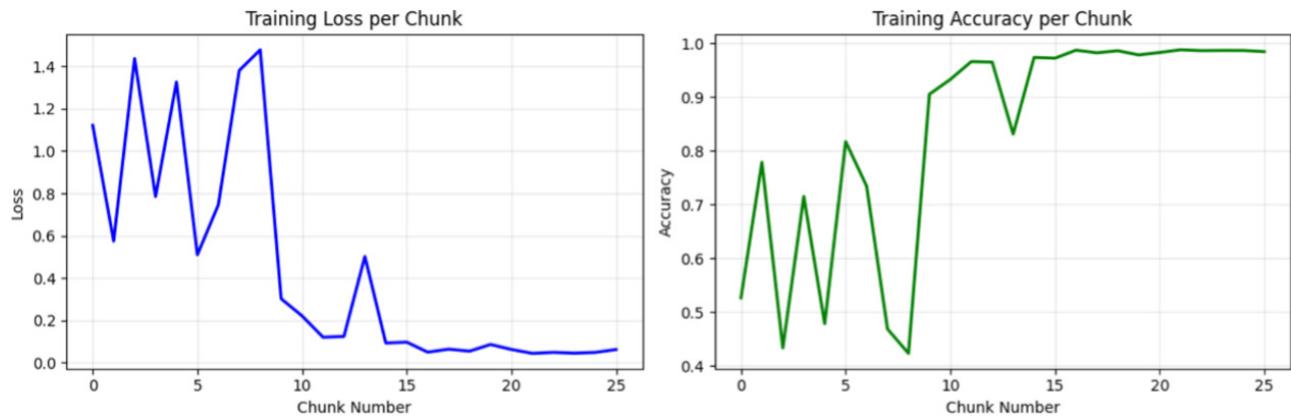


Рис. 5. Функция потерь и точность при обучении CNN

3.2. RNN

Для обработки временных последовательностей I/Q-отсчетов использовалась гибридная архитектура, сочетающая сверточную и рекуррентную нейронные сети. На первом этапе одномерные сверточные слои применялись к исходному сигналу для выделения локальных временных признаков низкого уровня. Полученная последовательность карт признаков подавалась на вход рекуррентного слоя с GRU-ячейками, которые способны улавливать долгосрочные временные зависимости в сигнале. Выход последней GRU-ячейки использовался для

классификации. Данная модель продемонстрировала наивысшее качество среди отдельных алгоритмов, достигнув F1-меры 0.96, что подтверждает высокую информативность исходной I/Q-последовательности. График обучения представлен на (рис. 6.)

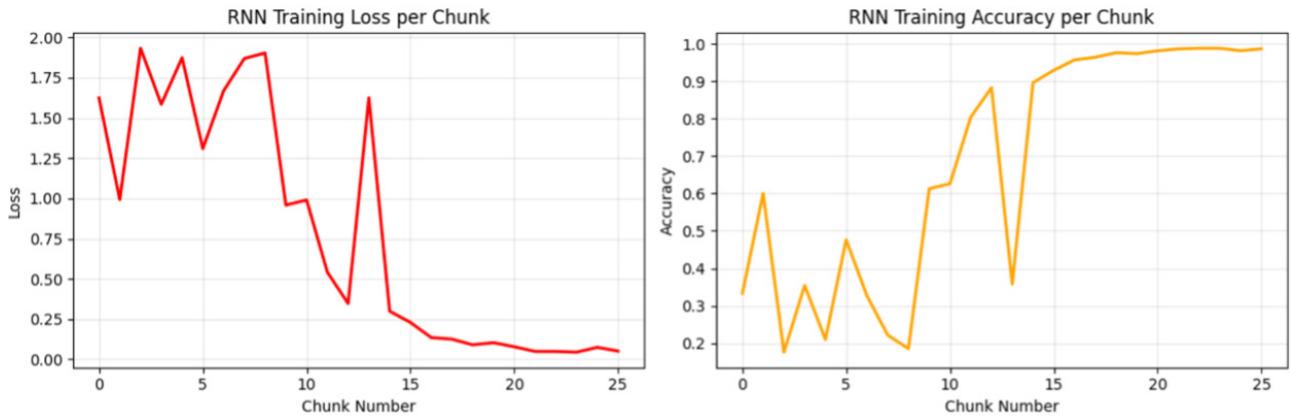


Рис. 6. Функция потерь и точность при обучении RNN

3.3. LGBM

Параллельно с глубинным обучением был применен алгоритм градиентного бустинга LightGBM, обученный на 23 предварительно извлеченных признаках. Комплекс признаков включал:

- Кумулянты высоких порядков (C20, C21, C40, C41, C42, C60) [5];
- Статистики огибающей (среднее значение, стандартное отклонение, асимметрия, эксцесс);
- Фазовые признаки (стандартное отклонение фазы, стандартное отклонение разности фаз, нелинейность фазы);
- Спектральные характеристики (центроид, разброс, отношения энергий в различных частотных диапазонах, равномерность, Crest-фактор);
- Временные признаки и оценку SNR;

Модель LightGBM показала F1-меру 0.45. Столь низкий результат по сравнению с нейросетевыми подходами указывает на то, что извлеченные признаки в совокупности несут менее информативную нагрузку для разделения классов, чем исходные сигналы или их спектрограммы.

3.4. Ансамбль

Для получения финального прогноза использовано ансамблирование предсказаний трех описанных выше моделей. Выходные вероятности каждой из моделей конкатенировались в единый вектор признаков. Этот вектор подавался на вход финального мета-классификатора — многослойного перцептрана с одним скрытым слоем. Такой подход позволяет модели научиться взвешивать предсказания базовых алгоритмов, учитывая их сильные и слабые стороны на разных типах сигналов. Итоговая F1-мера ансамбля составила 0.98, что на 2 % превышает результат базовой рекуррентной модели и демонстрирует эффективность комбинирования разнородных источников информации.

3.5. Анализ зависимости качества от уровня SNR

Проведен анализ изменения F1-меры итоговой ансамблевой модели в зависимости от уровня отношения сигнал-шум (SNR) на тестовой выборке. Результаты, представленные на рис. 7, показывают, что при низких уровнях SNR (от -20 дБ до -6 дБ) качество классифика-

ции находится на уровне, близком к случайному угадыванию ($F1 \approx 0.10$). Существенный рост метрики начинается с уровня -4 дБ. Начиная с $SNR = 8$ дБ, наблюдается резкое улучшение качества ($F1 > 0.80$), а при $SNR \geq 14$ дБ модель выходит на плато с $F1$ -мерой выше 0.98. Данная зависимость подтверждает ожидаемое поведение: надежная классификация возможна лишь при достаточном соотношении сигнал-шум. (рис. 7.)



Рис. 7. График зависимости $F1$ от SNR

Заключение

В ходе работы предложен и исследован подход к классификации БПЛА по радиоизлучению, сочетающий современные методы машинного обучения и требования к аппаратной реализации на ПЛИС. На основе данного подхода создана предиктивная система, обеспечивающая классификацию типа БПЛА в режиме, близком к реальному времени. Разработана отказоустойчивая и масштабируемая архитектура системы управления данными, которая легла в основу этой системы. Экспериментально подтверждено, что наивысшее качество классификации достигается при использовании ансамблевой модели, демонстрирующей $F1$ -меру 0.98. Анализ зависимости качества от уровня SNR показал, что надежная классификация возможна при $SNR \geq 8$ дБ. Полученные результаты, включая работоспособный прототип предиктивной системы, могут быть использованы для создания интеллектуальных систем радиоэлектронного противодействия БПЛА.

Литература

1. Николаев Н. В., Ильин В. В., Некрасов М. И. Актуальные вопросы противодействия современным автономным беспилотным летательным аппаратам и FPV-дронам // Вопросы безопасности. – 2024. – № 1. – URL: <https://cyberleninka.ru/article/n/aktualnye-voprosy-protivodeystviya-sovremennym-avtonomnym-bespilotnym-letatelnym-apparatam-i-fpv-dronam> (дата обращения: 19.11.2025).

2. Аджахунов Э. А., Николаев О. В. Комплекс перехвата управления БПЛА // Вестник Концерна ВКО Алмаз-Антей. – 2021. – № 2 (37). – URL: <https://cyberleninka.ru/article/n/kompleks-perehvata-upravleniya-bpla> (дата обращения: 19.11.2025).

3. Акопян Г. Л., Корнев В. В., Ремезов И. А. Стенд экспериментальной оценки эффективности помех при подавлении радио-приемного устройства обзора многофункционального радиолокатора // Радиотехника. – 2024. – Т. 88, № 10. – С. 137–147. DOI: <https://doi.org/10.18127/j00338486-202410-15>

4. Noisy Drone RF Signal Classification : [датасет] // Kaggle. – URL: <https://www.kaggle.com/datasets/sgluege/noisy-drone-rf-signal-classification> (дата обращения: 11.11.2025).

5. Курбаналиев В. К. Кумулянтные признаки для определения типа манипуляции сигналов // РЭНСИТ. – 2020. – № 3. URL: <https://cyberleninka.ru/article/n/kumulyantnye-priznaki-dlya-opredeleniya-tipa-manipulyatsii-signalov> (дата обращения: 20.11.2025).

ФУНКЦИОНАЛЬНО-КОГНИТИВНОЕ МОДЕЛИРОВАНИЕ КАК ИНСТРУМЕНТ СИСТЕМНОГО АНАЛИЗА БИЗНЕС-ПРОЦЕССОВ АГРОХОЛДИНГОВ

А. А. Репина

Воронежский государственный университет

Аннотация. В статье рассматривается вопрос интеграции IDEF0-моделирования, когнитивных карт и машинного обучения для оценки бизнес-процессов в агрохолдингах. На примере ЭкоНивы определены ключевые элементы бизнеса и принципы построения когнитивных карт. Описаны возможности применения алгоритмов обучения, таких как алгоритм Хебба, генетические алгоритмы и дифференциальная эволюция, для настройки параметров карт. Предложены теоретические и методологические принципы многоступенчатого подхода, от декомпозиции функциональных моделей до анализа сценариев. Статья является основой для будущих исследований в сфере формализации бизнес-процессов и разработки инструментов анализа организационных систем.

Ключевые слова: функциональное моделирование, IDEF0, когнитивные карты, нечеткие когнитивные карты, машинное обучение, алгоритм Хебба, обратное распространение ошибки, генетические алгоритмы, дифференциальная эволюция, управление бизнес-процессами, системный анализ, агрохолдинг, интеллектуальные системы.

Введение

В наше время большие агропромышленные холдинги — это сложные организации со множеством уровней, поэтому важно регулярно анализировать и описывать их бизнес-процессы. Методология функционального моделирования IDEF0, вместе с когнитивным моделированием и машинным обучением, даёт теоретические и практические инструменты, чтобы понять, как устроены и как меняются такие системы. Этот подход помогает зафиксировать знания о связях между частями организации и находить общие черты в управлении трудными процессами.

Группа компаний «ЭкоНива» — хороший пример того, как можно применить этот подход. Холдинг включает в себя предприятия молочного животноводства, растениеводства, переработки сельхозпродукции и семеноводства, расположенные в разных регионах. Из-за этого необходимо упорядочить и описать управление.

В этой статье мы рассмотрим теоретические и практические подходы к созданию функциональных моделей, когнитивных карт и их обучению с помощью машинного обучения на примере агрохолдинга. Также мы определим, что ещё можно подробно изучить в этой области.

1. Функциональное моделирование: фундамент системного анализа

1.1. Методология IDEF0 как инструмент структуризации

IDEF0 (Integrated Definition for Function Modeling) — это метод и нотация для создания моделей, описывающих бизнес-процессы. В IDEF0 особое внимание уделяется связям между элементами и логике операций, что полезно при анализе сложных организаций, например, агрохолдингов.

В основе IDEF0 лежит идея представления системы как набора взаимодействующих функциональных блоков, отражающих процессы и действия. Каждый блок имеет четыре типа связей: вход, управление, выход и механизм. В отличие от DFD и WFD, в IDEF0 обязательно указываются управляющие воздействия, что позволяет учитывать влияние правил и норм на процессы.

1.2. Анализ функциональной модели первого уровня «ЭкоНива»

Функциональная модель первого уровня агрохолдинга «ЭкоНива» показывает стандартное разделение системы управления на главные бизнес-направления. (рис. 1).

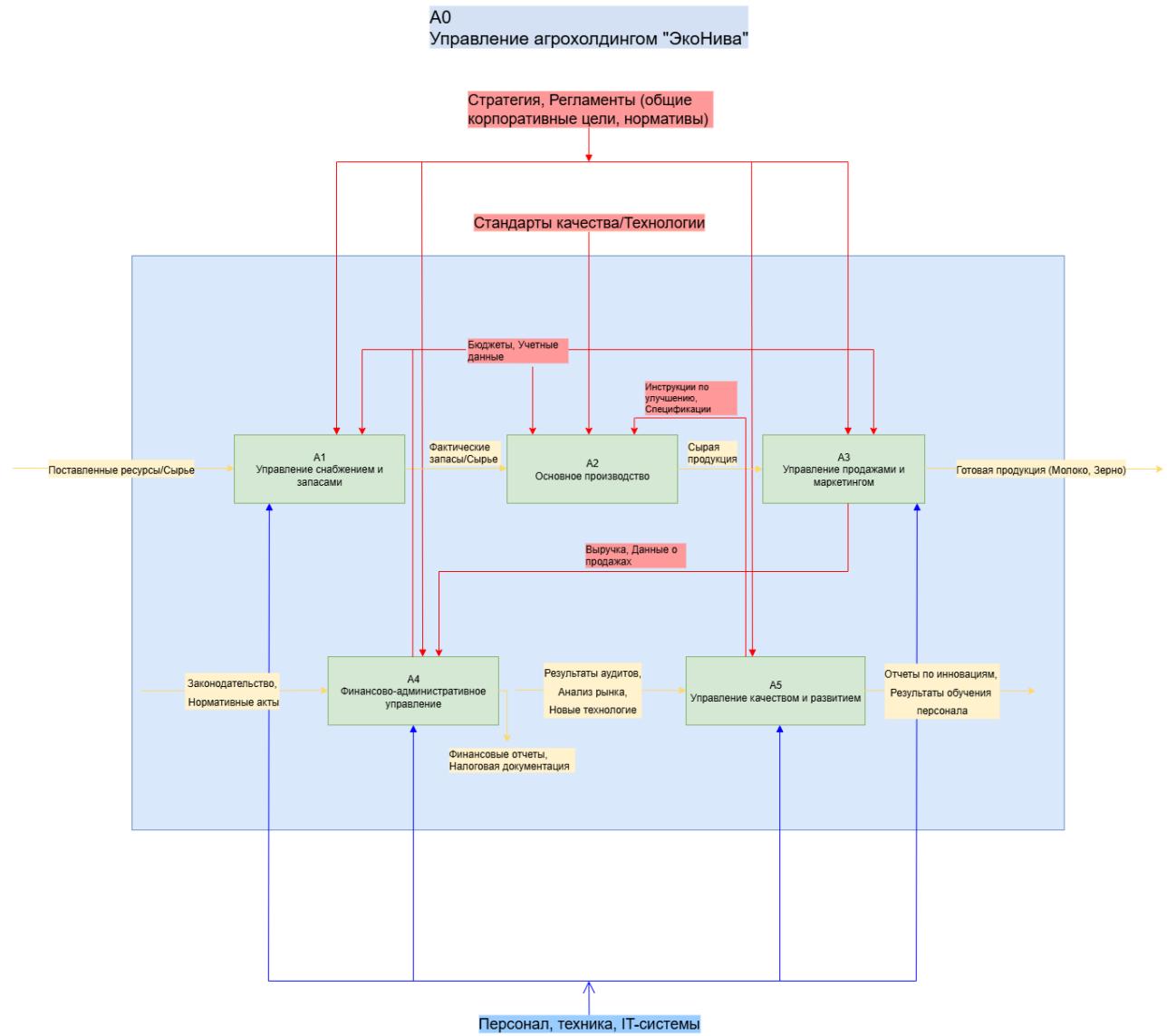


Рис. 1. Декомпозиция A0

Представлена контекстная диаграмма верхнего уровня (A0 «Управление агрохолдингом «ЭкоНива») и пять главных функциональных блоков:

- **A1 — Управление снабжением и запасами:** отвечает за поставки сырья, материалов и оборудования и создание нужных запасов. Данные о поставках ресурсов поступают на вход, управление происходит на основе законов и нормативов, на выходе формируются показатели запасов и закупок.

- **A2 — Основное производство:** главный блок, включающий молочное животноводство, растениеводство и переработку. Управление происходит на основе стандартов качества и технологических правил, на выходе — производственные показатели и объемы сырой продукции.

- **A3 — Управление продажами и маркетингом:** обеспечивает работу с рынком, сбыт готовой продукции и анализ конкурентов.

- **A4 — Финансово-административное управление:** центральный блок, отвечающий за финансы, бухгалтерию, налоги и стратегическое управление.

- **A5 — Управление качеством и развитием:** блок, который следит за качеством, развитием и обучением сотрудников.

Уровень «Персонал, техника, ИТ-системы» обеспечивает все блоки нужными ресурсами, а верхний управляющий контур задает стратегию и требования к процессам.

1.3. Принцип декомпозиции и структурная целостность модели

В IDEF0 применяется принцип иерархического разделения сложных функций на составляющие части. Каждый функциональный блок может быть детализирован на диаграмме, включающей от трёх до шести блоков. Это позволяет соблюсти баланс между уровнем детализации и удобством восприятия. При разделении все входящие и исходящие интерфейсные дуги родительского блока сохраняются на диаграмме, что обеспечивает целостность модели и даёт возможность выделить подпроцессы для анализа.

2. Когнитивные карты: моделирование причинно-следственных связей

2.1. Переход от функциональной модели к когнитивной карте

После разработки функциональной модели следующим шагом следует создание когнитивной карты. Это ориентированный граф, где узлы — это концепции (факторы, показатели, объекты), а дуги — причинно-следственные связи между ними. Когнитивная карта помогает увидеть, как разные факторы влияют друг на друга, в том числе и через цепочки причин и следствий.

Переход от IDEF0-модели к когнитивной карте происходит через выбор основных концепций из функциональных блоков. Для агрохолдинга «ЭкоНива» важными группами концепций являются: финансовые показатели (прибыль, расходы, инвестиции), параметры производства (объемы, качество, производительность), ресурсы (персонал, техника, земля, корма), рыночные показатели (спрос, цены, конкуренция) и управленческие параметры (эффективность управления, структура организации).

2.2. Нечеткие когнитивные карты (FCM)

Для лучшего отображения сложных систем применяют нечеткие когнитивные карты (Fuzzy Cognitive Maps — FCM). Они расширяют возможности обычных когнитивных карт благодаря использованию нечеткой логики. В FCM каждая связь между концептами имеет вес ω_{ij} , где $-1 \leq \omega_{ij} \leq 1$.

Положительный вес говорит об усилении влияния, отрицательный — об ослаблении, а ноль — об отсутствии прямой связи. Структура FCM описывается матрицей весов $[\omega_{ij}]$, которая полностью задает динамику системы.

Примеры причинно-следственных связей в системе управления холдингом:

- «Инвестиции в технологии» (+0.7) → «Производительность труда»
- «Производительность труда» (+0.8) → «Объем производства»
- «Объем производства» (+0.6) → «Прибыль»
- «Затраты на ресурсы» (-0.5) → «Прибыль»
- «Квалификация персонала» (+0.6) → «Качество продукции»

3. Алгоритмы обучения когнитивных карт

3.1. Актуальность применения машинного обучения

Применение экспертных методов при построении когнитивных карт сталкивается с рядом ограничений, таких как субъективность оценок, сложность параметризации и недостаточная адаптивность. Использование алгоритмов машинного обучения позволяет автоматически настраивать структуру и параметры карт на основе эмпирических данных. Это, в свою очередь, увеличивает объективность модели и ее способность адаптироваться к меняющимся условиям.

3.2. Алгоритм Хебба

Алгоритм Хебба, предложенный в 1949 году, основан на принципе: «Нейроны, которые активируются вместе, связываются вместе». Применительно к когнитивным картам:

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta C_i(t)C_j(t),$$

где η — коэффициент обучения. Этот алгоритм усиливает связи между одновременно активными концептами, и уменьшает их при противоположном поведении. Простота осуществления является положительным аспектом, в то время как отсутствие чётко заданной целевой функции выступает в качестве недостатка.

3.3. Генетические алгоритмы

Генетические алгоритмы (ГА) применяются для решения задач глобальной оптимизации, используя механизмы селекции, скрещивания и мутации. ГА позволяют одновременно оптимизировать как веса связей (параметрическая оптимизация), так и структуру карты (структурная оптимизация). Основные операторы включают представление решения (кодирование матрицы весов), функцию приспособленности (например, точность прогноза), селекцию лучших особей, скрещивание для создания потомков и мутацию для поддержания разнообразия.

3.4. Алгоритм дифференциальной эволюции (*Differential Evolution*)

Алгоритм DE эффективен для непрерывной оптимизации, используя операторы мутации, скрещивания и селекции:

$$v_i = x_{r1} + F(x_{r2} - x_{r3}),$$

где F — коэффициент дифференциации. В силу своей хорошей сходимости и устойчивости к локальным экстремумам, алгоритм дифференциальной эволюции (DE) представляет большой интерес для оптимизации весов когнитивных карт.

3.5. Алгоритм Гроссберга (*Instar/Outstar*)

Нейроны инстар и оутстар Гроссберга решают задачи распознавания образов и воспроизведения паттернов. Инстар адаптирует входящие веса:

$$\omega_i(t+1) = \omega_i(t) + \alpha o(x_i - \omega_i(t)).$$

Оутстар согласовывает исходящие веса с целевыми значениями. Эти механизмы применимы для локальной оптимизации отдельных узлов когнитивной карты.

4. Методологические основы и направления развития

4.1. Многоэтапный методологический подход

Систематизированный подход к построению и улучшению когнитивных карт состоит из нескольких этапов:

1. **Детальная декомпозиция функциональных моделей** до третьего-четвёртого уровня для выделения подпроцессов и их параметров.
2. **Построение локальных когнитивных карт** для каждого функционального блока с выявлением причинно-следственных связей и экспертной параметризацией.
3. **Объединение в единую корпоративную модель** путём нахождения связей между блоками и формирования полной когнитивной карты организации.
4. **Параметризация и калибровка** на основе исторических данных деятельности холдинга, используя корреляционный анализ, тесты причинности и экспертные оценки.
5. **Обучение и оптимизация** с помощью алгоритмов машинного обучения, которые выбираются в зависимости от имеющихся данных.
6. **Валидация и сценарный анализ** для проверки адекватности модели и прогнозирования последствий различных управленческих решений.

4.2. Возможные направления дальнейшего исследования

Теоретические и методологические результаты, полученные в статье, открывают следующие направления для дальнейших исследований:

1. **Декомпозиция и формализация бизнес-процессов.** Рекомендуется построить полные IDEF0-модели второго и третьего уровней для всех пяти основных блоков функциональной модели «ЭкоНивы». Особое внимание стоит уделить моделированию блока А2 «Основное производство», выделив подпроцессы молочного животноводства, растениеводства, переработки и селекции. Это позволит определить внутренние взаимосвязи и составить каталог управляемых переменных для каждого подразделения холдинга.
2. **Создание когнитивных карт.** Исследование может быть продолжено созданием когнитивных карт для отдельных производственных блоков, учитывая временные, технологические и управленческие аспекты. Например, когнитивная карта молочного животноводства могла бы включать аспекты, касающиеся животноводческого цикла, качества кормов, ветеринарного управления и производственных показателей. Разработка таких моделей позволит лучше понять механизмы, определяющие эффективность отдельных направлений деятельности.
3. **Интеграция корпоративной когнитивной модели.** Следующим шагом должно стать объединение локальных когнитивных карт в единую корпоративную модель с акцентом на межблочные связи. Это включает выявление общих факторов, влияющих на несколько блоков (например, квалификация персонала, уровень технической оснащённости), и установление иерархии причинно-следственных связей между различными уровнями организации.
4. **Параметризация на основе данных.** Важным этапом является разработка способов параметризации когнитивных карт на основе исторических данных холдинга. Это может включать: анализ временных рядов производственных показателей с использованием методов проверки причинности по Грэнджеру, построение регрессионных моделей для оценки силы связей, разработку процедур проверки полученных параметров на основе экспертных оценок.
5. **Сравнительный анализ алгоритмов обучения.** Перспективным направлением является сравнение различных алгоритмов обучения когнитивных карт (Хебба, генетических алгоритмов, дифференциальной эволюции) с точки зрения точности, скорости сходимости, устойчивости к локальным минимумам и необходимого объёма данных. Такой анализ позволит

выработать рекомендации по выбору подходящего алгоритма в зависимости от имеющихся данных и сложности когнитивной карты.

6. Развитие инструментов анализа сценариев. Дальнейшие исследования могут быть направлены на разработку методов и инструментов для анализа сценариев на основе обученных когнитивных карт. Это включает: разработку методологии определения воздействий на управляемые аспекты, анализ распространения изменений через сеть причинно-следственных связей, определение ключевых аспектов, оказывающих наибольшее влияние на целевые показатели, сравнение сценариев на основе критериев.

Заключение

Представленная в статье теоретическая и методологическая основа показывает, что функциональное моделирование IDEF0 в сочетании с когнитивным моделированием и машинным обучением служит хорошим способом систематизировать и анализировать бизнес-процессы в сложных организациях, например, в крупных агрохолдингах.

Изучение применимости этих методов на примере агрохолдинга «Эко-Нива» говорит о том, что такой подход полезен для формализации знаний о структуре и динамике организации. Рассмотренные алгоритмы обучения (Хебба, генетические алгоритмы, дифференциальная эволюция и алгоритм Гроссберга) дают разные возможности для настройки параметров когнитивных карт в зависимости от данных и целей анализа.

Статья — база для дальнейшего изучения формализации бизнес-процессов, создания и обучения когнитивных моделей организаций, а также разработки методов и инструментов для их практического применения. Дальнейшее развитие включает подробную декомпозицию функциональных моделей, создание специализированных и интегрированных когнитивных карт, настройку на основе реальных данных, сравнение алгоритмов обучения, развитие инструментов сценарного анализа.

Литература

1. Коско Б. Нечеткие когнитивные карты / Б. Коско // International Journal of Man-Machine Studies. – 1986. – Vol. 24. – P. 65–75.
2. Новиков Д. А. Теория управления организационными системами : учебник / Д. А. Новиков. – 2-е изд. – Москва : МПСИ, 2005. – 584 с.
3. Ясенев В. Н. Функциональное моделирование бизнес-процессов : учеб. пособие / В. Н. Ясенев. – Москва : Финансы и статистика, 2007. – 192 с.
4. Хебб Д. О. Организация поведения / Д. О. Хебб ; пер. с англ. Н. Ю. Алексеенко. – Москва : Эксмо, 2003. – 640 с.
5. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилинский, Л. Рутковский ; пер. с польск. И. Д. Рудинского. – Москва : Горячая линия-Телеком, 2004. – 452 с.
6. Осипов Г. С. Методы искусственного интеллекта : учеб. пособие / Г. С. Осипов. – 2-е изд., испр. – Москва : Физматлит, 2015. – 296 с.

ПРИМЕНЕНИЕ ГРАФОВЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСКРОЯ

В. О. Решетов¹, А. А. Арзамасцев^{1,2}, Н. А. Зенкова³

¹Воронежский государственный университет

²Тамбовский филиал Межотраслевого научно-технического комплекса «Микрохирургия глаза»
имени академика С. Н. Федорова

³Тамбовский государственный университет имени Г. Р. Державина

Аннотация. В статье рассматривается применение графовых нейронных сетей (Graph Neural Networks, GNN) для решения задачи раскюя рулонных материалов. Показано, что задача раскюя может быть представлена в виде графовой структуры, где вершины соответствуют элементам раскраиваемого материала, а рёбра — отношениям совместимости или конфликтов между ними. Представлены основные архитектуры GNN, такие как Graph Convolutional Network и Graph Attention Network, а также предложен комбинированный подход к применению GNN в задаче раскюя. Приведены результаты синтетического эксперимента, демонстрирующие преимущества GNN по сравнению с традиционными эвристическими методами.

Ключевые слова: графовые нейронные сети (GNN), Архитектуры GNN, Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), задача раскюя.

Введение

Задача раскюя рулонных материалов относится к классу комбинаторных задач оптимизации и широко используется в промышленности — от упаковочного производства до текстильной и полиграфической отраслей.

Традиционные методы решения, включая линейное программирование и эвристические алгоритмы, часто демонстрируют ограниченную эффективность при увеличении размерности задачи и усложнении конфигураций раскюя.

В последние годы активно развиваются методы глубокого обучения, способные решать задачи с высокой комбинаторной сложностью. Особое внимание привлекают графовые нейронные сети, которые позволяют описывать и обрабатывать данные в форме графов. Применение GNN к задаче раскюя открывает новые возможности для оптимизации процесса и поиска решений, учитывающих сложные взаимосвязи между элементами, так как в отличие от традиционных нейронных сетей, которые работают с данными, представленными в виде табличных или последовательных структур, кроме того, графы обладают сложной нерегулярной структурой.

Графовые нейронные сети предоставляют инструменты для распространения и агрегации информации между вершинами графа, что позволяет моделям обучаться на основе локальных и глобальных взаимосвязей. Это открывает широкие возможности для решения задач, где важны не только свойства отдельных элементов, но и их взаимосвязи.

Цель работы — проанализировать возможность минимизации отходов материала при разрезании рулона на фрагменты заданных размеров с помощью графовых сверточных сетей (GCN) и графовых сетей с механизмом внимания (GAT).

1. Особенности графовых нейронных сетей в задаче раскюя

Графовые нейронные сети — это разновидность глубоких нейронных сетей, разработанных для работы с данными, которые естественным образом описываются в виде графов. В отличие от традиционных архитектур, таких как полносвязные (Fully Connected Networks) или

сверточные сети (Convolutional Neural Networks), графовые нейронные сети способны обрабатывать данные произвольной структуры, где связи между элементами имеют важное значение для результата [1].

В контексте задачи раскроя графовое представление данных позволяет моделировать взаимосвязи между элементами, которые должны быть размещены на поверхности материала, следующим образом:

- Узлы графа могут соответствовать отдельным деталям (фрагментам раскроя), обладающим определёнными параметрами — размерами, формой, ориентацией, приоритетом производства и т. д.

- Рёбра графа описывают отношения между деталями: возможные конфликты размещения, допустимые комбинации, последовательность резов, технологические ограничения и т. п.

Благодаря такому представлению, GNN позволяют не просто анализировать отдельные элементы, а учитывать контекст их взаимного расположения, что особенно важно для оптимизации использования материала и минимизации отходов. Такое представление позволяет применять обучаемые модели, где GNN выполняет роль предсказателя оптимальной компоновки элементов. Модель может обучаться на синтетических данных, сгенерированных из реальных производственных параметров таких как ширина рулона, список заказов, минимальные зазоры и т. д.

Графовые нейронные сети выполняют распространение информации между связанными узлами, что позволяет модели «понимать» глобальную структуру задачи: например, какие элементы лучше расположить рядом, какие — разнести, или как упорядочить операции резки.

Таким образом, GNN применяют мощные методы глубокого обучения к структурам данных, которые описывают объекты (детали раскроя) и их взаимосвязи (технологические и геометрические зависимости).

Для задачи раскроя могут рассматриваться все три типа прогнозов, характерных для графовых моделей:

- на уровне узла — определение оптимального положения или ориентации конкретной детали;
- на уровне ребра — прогнозирование допустимости или предпочтительности совместного размещения двух деталей;
- на уровне графа — оценка качества всего раскроя в целом (например, коэффициента использования материала).

Таким образом, графовые нейронные сети обеспечивают единый подход для решения широкого спектра подзадач раскроя, позволяя интегрировать геометрические, технологические и экономические факторы в рамках одной модели.

2. Архитектуры графовых нейронных сетей

2.1. Графовые сверточные сети

Графовые сверточные сети — это одна из ключевых архитектур графовых нейронных сетей, которая адаптирует идею сверток из традиционных сверточных нейронных сетей для работы с графиками. GCN были предложены для того, чтобы эффективно обрабатывать данные, представленные в виде графов, и использовать информацию о взаимосвязях между вершинами для улучшения предсказательной способности модели [2].

В основе GCN лежит идея свертки на графике — операция, позволяющая объединять информацию от соседних вершин и обновлять представление каждой вершины на основе этих данных. В случае графов это аналогично тому, как в CNN каждая точка изображения (пиксель) обновляется на основе соседних пикселей. Однако в графах мы имеем нерегулярную структу-

ру, где у каждой вершины разное количество соседей, и ребра могут быть ненаправленными или взвешенными [3].

Свертка по графу происходит через следующие шаги:

- агрегация, где каждый узел получает представления своих соседей и объединяет их в единое представление;
- нормализация, которая делит вклады соседей на степень вершины (количество её соседей) и помогает сбалансировать вклад каждого соседа, избежать проблемы избыточного усреднения.;
- линейная трансформация.

После того как агрегированная информация собрана, она преобразуется с помощью линейной трансформации. Это похоже на линейный слой в традиционной нейронной сети, где к данным применяется обучаемая матрица весов для того, чтобы «обучить» модель выявлять важные признаки. Формально это можно записать как:

$$H^{(k+1)} = \sigma(\tilde{A}H^kW^k), \quad (1)$$

где H^k — матрица признаков на k -м слое, где каждая строка соответствует представлению (вектору признаков) каждой вершины,

W^k — обучаемая матрица весов для слоя k ,

\tilde{A} — нормализованная матрица смежности графа с добавлением единичной матрицы для учёта самой вершины (self-loop),

σ — нелинейная функция активации, например, ReLU.

После применения линейной трансформации и нелинейной активации (например, ReLU) каждая вершина получает обновленное представление, которое учитывает информацию не только о самой вершине, но и о её соседях. Этот процесс может повторяться на нескольких слоях сети, чтобы захватить информацию от всё более дальних соседей.

GCN использует нормализованную матрицу смежности графа для обеспечения устойчивой агрегации информации. Эта нормализация обеспечивает равномерное распределение вклада каждой вершины в обновлённое представление и предотвращает «раздувание» представлений у вершин с большим количеством связей.

Одной из ключевых характеристик GCN является количество слоёв — глубина сети. В задаче раскroя глубина сети определяет, насколько широко каждая деталь «видит» своё окружение — какие другие элементы и технологические ограничения учитываются при формировании её признакового представления.

На практике обычно используют от 2 до 3 слоёв, так как более глубокие GCN сталкиваются с проблемой размывания признаков (over-smoothing) [4]. Это явление заключается в том, что по мере распространения информации между узлами представления всех деталей (вершин) становятся излишне похожими друг на друга. Таким образом, модель теряет способность различать уникальные характеристики отдельных элементов — например, размеры, форму или приоритет размещения — и, как следствие, хуже различает потенциально выгодные и невыгодные конфигурации.

Таким образом, при проектировании GCN для задачи раскroя важно находить баланс между глубиной сети и выразительностью модели: достаточную глубину, чтобы учитывать контекст взаимодействия деталей, но не настолько большую, чтобы нивелировать различия между ними [5].

2.2. Graph Attention Networks

Graph Attention Networks — это одна из продвинутых архитектур графовых нейронных сетей, которая использует механизм внимания для более эффективной работы с графовыми данными. В отличие от Graph Convolutional Networks, где информация от всех соседей усред-

няется с одинаковыми весами, GAT позволяет сети обучать веса, которые отражают важность каждого соседа для текущей вершины. Это делает GAT более гибкими и мощными для задач, где значение связей между вершинами может варьироваться [6].

Вместо того чтобы одинаково учитывать всех соседей (как это делается в GCN), GAT динамически обучает веса внимания для каждого ребра графа. Модель определяет, какие вершины (соседи) наиболее важны для данной вершины и какова их относительная значимость. Этот процесс осуществляется следующим образом:

- каждая вершина v вычисляет коэффициенты внимания a_{vu} для своих соседей $u \in N(v)$;
- эти коэффициенты показывают, насколько важен каждый сосед u для вершины v при обновлении её состояния. Значения коэффициентов зависят от представлений самой вершины v и её соседа u .

Коэффициенты внимания a_{vu} вычисляются через механизм точечного внимания (dot-product attention), который принимает на вход векторные представления двух вершин — целевой вершины v и её соседа u . Формально, коэффициенты внимания можно описать следующим образом:

$$e_{vu} = \text{Leaky ReLU}(a^T [Wh_v \parallel Wh_u]), \quad (2)$$

где h_v, h_u — это входные представления (векторы признаков) вершин v и u ,

W — обучаемая матрица весов, применяемая для линейной трансформации признаков,

a — вектор, который определяет важность взаимодействия между вершинами,

\parallel — операция конкатенации (объединения) векторов h_v и h_u ,

Leaky ReLU — нелинейная функция активации для моделирования зависимости между вершинами.

Значение e_{vu} представляет собой ненормализованный коэффициент важности, который затем нормализуется с помощью softmax-функции по всем соседям вершины v . Softmax делает сумму всех коэффициентов внимания для соседей вершины v равной 1. Таким образом, каждый сосед вносит вклад в обновление представления вершины в соответствии с его важностью.

После вычисления коэффициентов внимания a_{vu} , происходит процесс агрегации. Обновлённое представление вершины v формируется как взвешенная сумма представлений её соседей u , где веса — это коэффициенты внимания:

$$h_v^i = \sigma \left(\sum_{u \in N(v)} a_{vu} Wh_u \right), \quad (3)$$

где h_v^i — обновлённое представление вершины v ,

a_{vu} — коэффициент внимания для соседа u ,

W — обучаемая матрица весов для линейной трансформации,

σ — нелинейная функция активации, например, ReLU.

Этот процесс позволяет каждому узлу обновлять свои признаки, уделяя больше внимания важным соседям и игнорируя менее значимых.

В контексте задачи это означает, что модель может динамически определять, какие связи между деталями наиболее значимы. Например:

- если две детали часто конфликтуют при размещении (пересекаются или нарушают технологические зазоры), GAT может присвоить их связи высокий вес, чтобы сильнее учитывать это ограничение;
- наоборот, для деталей, которые редко взаимодействуют, вес связи будет минимальным, чтобы не перегружать модель лишней информацией.

Таким образом, GAT позволяет модели «внимательнее» относиться к наиболее критичным взаимосвязям между элементами раскрыя, улучшая точность прогнозов размещения и эффективность использования материала.

Это особенно полезно в случаях, когда структура зависимостей между деталями сложна и неоднородна — например, при комбинировании элементов разных форматов, толщин или технологических приоритетов.

3. Синтетический эксперимент

В качестве демонстрационного эксперимента был сформирован синтетический набор данных, содержащий 500 элементов различной длины, соответствующих деталям для раскрова рулонного материала. Для всех элементов была задана фиксированная ширина рулона и минимальные технологические зазоры между деталями.

Были протестированы две архитектуры графовых нейронных сетей: GCN и GAT, с целью минимизации остатка материала. Узлы графа соответствовали отдельным деталям, а рёбра отражали совместимость и ограничения по размещению. Результаты эксперимента приведены в табл. 1.

Таблица 1

Сравнение методов по среднему проценту отходов и времени обучения

Метод	Процент отходов	Время обучения	Примечание
Жадный алгоритм	20 %	–	Классическая эвристика, базовый уровень использования материала; быстрый расчёт без обучения
GCN	18 %	1 час	Стабильные результаты на синтетическом наборе; небольшое улучшение по сравнению с жадным алгоритмом
GAT	21 %	1,5 часа	Механизм внимания позволяет выделять ключевые связи между деталями; эффект ограничен малым объёмом данных, возможны колебания результатов

На малом объёме данных (500 элементов) GCN показала стабильные и предсказуемые результаты благодаря относительно простой архитектуре. GAT, несмотря на возможность выделять более значимые связи между деталями за счёт механизма внимания, оказалась более чувствительной к объёму данных. В условиях ограниченного количества примеров она не продемонстрировала даже незначительное улучшение по сравнению с GCN, что объясняется сложностью обучения весов внимания [7].

Таким образом, демонстрационный эксперимент показывает, что даже на малых данных GCN и GAT могут быть сопоставимы с эвристикой, но значимое архитектурное преимущество GAT может проявиться только при увеличении объёма данных или усложнении взаимосвязей между деталями [8]. Время обучения GAT выше за счёт более сложной архитектуры и необходимости оптимизации весов внимания.

Заключение

Графовые нейронные сети показали себя перспективным инструментом для решения задач раскрова. Представление деталей и их взаимосвязей в виде графа позволяет учитывать как индивидуальные свойства элементов, так и сложные взаимодействия между ними.

В синтетическом эксперименте GCN продемонстрировала стабильные результаты на малом объёме данных, тогда как GAT, благодаря механизму внимания, способна выделять значимые связи, но требует большего количества данных для проявления преимущества.

Результаты подтверждают, что GNN могут сопоставимо работать с традиционными эвристическими методами, а при увеличении объёма и сложности данных способны повысить эф-

фективность использования материала. Дальнейшие исследования могут быть сосредоточены на гибридных моделях и адаптивном обучении для промышленных задач раскroя.

Литература

1. *Tanis J. H., Giannella C., Mariano A. V.* Introduction to graph neural networks: a starting point for machine learning engineers //arXiv preprint arXiv:2412.19419. – 2024.
2. *Cui G., Wei Z., Su H. H.* Rethinking the Expressiveness of GNNs: A Computational Model Perspective // arXiv preprint arXiv:2410.01308. – 2024.
3. *Holagh N. A., Kobti Z.* Survey of Graph Neural Network Methods for Dynamic Link Prediction // Procedia Computer Science. – 2025. – T. 257. – C. 436–443.
4. *Du H.* DenseGNN: universal and scalable deeper graph neural networks for high-performance property prediction in crystals and molecules //npj Computational Materials. – 2024. – T. 10, № 1. – C. 292.
5. *Cantürk F.* Scalable Primal Heuristics Using Graph Neural Networks for Combinatorial Optimization //Journal of Artificial Intelligence Research. – 2024. – T. 80. – C. 327-376.
6. *Guo S.* A Survey on GAT-like Graph Neural Networks //2020 International Conference on Communications, Information System and Computer Engineering (CISCE). – IEEE, 2020. – C. 303–308.
7. *Wang J.* A survey on graph neural networks //EAI Endorsed Transactions on E-Learning. – 2022. – T. 8. – №. 3.
8. *Zhou J.* Graph neural networks: A review of methods and applications // AI open. – 2020. – T. 1. – C. 57–81.

АЛГОРИТМЫ ГЕНЕРАЦИИ МНОЖЕСТВ ДАННЫХ НА ОСНОВЕ АКТИВНОГО ЭКСПЕРИМЕНТА ДЛЯ МАТЕМАТИЧЕСКОГО РЕМОДЕЛИРОВАНИЯ

П. В. Сараев

Липецкий государственный технический университет

Аннотация. Данная работа посвящена практической реализации концепции математического ремоделирования, т. е. процесса трансформации уже имеющейся модели. Построение новой модели производится на основе обучающего множества данных, которое генерируется на основе известной математической модели, т. е. с помощью активного эксперимента. Предложены алгоритмы для генерации множества данных: метод Монте-Карло; комбинаторный; на основе построения регулярной сетки; адаптивный метод, учитывающий степень сложности исходной функции. Вычислительные эксперименты на основе разработанного программного обеспечения на языке Python показали влияние генерируемого множества данных на результат ремоделирования, перспективными являются комбинаторный и адаптивный алгоритмы.

Ключевые слова: математическое ремоделирование, генерация множества данных, активный эксперимент.

Введение

Математическое ремоделирование это процесс преобразования моделей одного класса (одной структуры) в другой класс моделей [1]. В отличие от задачи идентификации математических моделей изначально должна быть известна модель. Построение новой модели производится путём параметрической идентификации на основе генерации обучающего множества данных с использованием известной модели и дальнейшей минимизацией функции ошибки. Ключевым вопросом в математическом ремоделировании выступают вопрос того, какого объема и на основе какого алгоритма целесообразнее генерировать промежуточное множество данных. Использование известной модели означает то, что производится активный эксперимент. Управление ходом активного эксперимента позволяет влиять на качество ремоделирования. В данной работе целью является формализация алгоритмов генерации множеств данных и анализ их влияния на качество построения новой модели на основе вычислительного эксперимента.

1. Основы математического ремоделирования

Обозначим через f известную математическую модель. Пусть для упрощения рассмотрения $f: X \subset R^n \rightarrow R$ — статическая модель. Обозначим через $M = \{M_i\}$ множество различных классов моделей, тогда для модели f класс, к которому она принадлежит, будем обозначать через M_f . Обозначим через g собой модель другого, ремоделирующего, класса M_g . Это класс моделей, в который стоит задача преобразования модели f . Математического ремоделирования это поиск оператора (процедуры, алгоритма) $T: M_f \rightarrow M_g$.

Цель ремоделирования — снижение сложности решения исходной задачи. Для этого может быть введено понятие сложности модели $C(f): C: f \rightarrow R_+$, где $R_+ = [0; +\infty]$. В ремоделирование необходимо построение новой модели g , такой что $C(g) < C(f)$. Оценка сложности может учесть время получения выходного результата для известного входа, тогда это задача становится задачей суррогатного моделирования [2]. Если сложность отражает норму вектора параметров модели, то тогда ремоделирование может считаться развитием метода регуляризации А. Н. Тихонова [3].

При построении новой модели необходимо быть готовым к снижению качества моделирования. Определим ошибку модели как $\varepsilon : f \rightarrow R_+$, которая в первом приближении может оцениваться как ошибка в заданной метрике на основе генерируемого множества данных $\{X, Y\}$.

Таким образом, можем формализовать задачу математического ремоделирования в следующей форме:

$$C(g)_{g \in M_g} + \alpha \varepsilon(g) \rightarrow \min,$$

где $\alpha > 0$ — коэффициент, задающий значимость снижения сложности и точности ремоделирования. В таком виде целью является достижение компромисса между повышением снижением сложности модели и снижением ее качества. Это делает интерпретацию задачи математического ремоделирования сходной с регуляризацией. Если оценка сложности может быть произведена после построения новой модели, то может решаться классическая задача параметрической идентификации путем минимизации $\varepsilon(g)$.

Обобщенный алгоритм математического ремоделирования представляется следующим образом:

1. Определение оценки сложности $C(g)$, исходя из постановки бизнес-задачи.
2. Определение границ изменения переменных $[a_i, b_i]$ для каждой входной переменной x_i .
3. Определение класса моделей M_g , зависящих от набора неизвестных, требующих идентификации, параметров $\{c_i \in [c_{i_{\min}}, c_{i_{\max}}]\}$.
4. Определение объема m для генерации множества данных.
5. Выбор стратегии генерации данных.
6. Формирование обучающего множества $\{X, Y\}$ на основе модели f путем проведения активного эксперимента.
7. Выбор метрики $\varepsilon(g)$ для оценки точности новой модели.
8. Параметрическая идентификация модели g путем минимизации $\varepsilon(g)$ на основе множества $\{X, Y\}$.
9. Оценка сложности модели $C(g)$ по сравнению с $C(f)$.
10. Вывод из решения исходной задачи относительно целесообразности применения процедуры ремоделирования.

Хотя большое значение играет алгоритмизация и реализация минимизации ошибки обучения, это задача является общей для всех методов машинного обучения. Специфика ремоделирования это формирование обучающего множества. Перейдем к описанию и анализу соответствующих алгоритмов.

2. Алгоритмы активного эксперимента для генерации множеств данных

Перед формулированием алгоритмов формирования множеств данных целесообразно определить объем множества m . Это значение определяется, в первую очередь, с учетом имеющихся вычислительных ресурсов. С одной стороны, в машинному обучении постулируется подход, который заключается в том, что чем больше данных, тем лучше. Однако, на практике увеличение множества данных приводит к сложению вида целевой функции, и, соответственно, к большим затратам на ее оптимизацию, невозможность применения ряда вычислительных процедур при решении оптимизационной задачи. С учетом того, что как правило, при построении моделей требуется подбор гиперпараметров, это существенно увеличивает время решения задачи. Объем множества m можно рассматривать гиперпараметром процесса ремоделирования. Предположим заданным значение m .

Предложим следующие подходы к генерации множества данных:

- 1) метод Монте-Карло;
- 2) комбинаторный — на основе перебора вершин гиперпараллелепипеда;

3) на основе регулярной сетки;

4) адаптивный метод, учитывающий степень сложности исходной функции f в разных частях рассматриваемой области изменения функции.

Данный список и их содержание изменены по сравнению с опубликованными ранее авторскими результатами. Кроме того, каждый из алгоритмов отражает основную сущность, но в своей работе может использовать элементы других алгоритмов формирования множества точек для обучающего множества. В каждом из последующих алгоритмов после формирования множества точек в пространстве изменения входных переменных вычисляется значение функции f , которое возвращается в качестве множества указаний учителя вместе с самим множеством точек.

2.1. Метод Монте-Карло

Базовый подход для генерации множества точек, который заключается в циклическом выборе некоторой случайной точки из $[a_i, b_i]^n$ на основе равномерного закона распределения. Цель заключается в максимально полном охвате точками всей рассматриваемой области изменения функции f . Можно использовать законы распределения, отличные от равномерного, но в этом случае плотность наполнения точками пространства будет различной. Выбор закона распределения для генерации должен быть некоторым образом обоснован. В ремоделировании проще использовать, всё же, равномерный закон распределения. Для практического применения требуется определить шаг дискретизации в пространстве входных переменных.

2.2. Комбинаторный алгоритм

Данный алгоритм основан на переборе различных комбинаций точек, начиная с граничных. По сути, этот алгоритм похож на подход решения задач планирования эксперимента. В классическом варианте, когда по каждой координате выбираются границы изменения переменных a_i , b_i и строится декартово произведение по всем n координатам $\{a_i, b_i\}^n$. Однако, такой способ малоприменим на практике, т. к. при $n = 100$ переменных таких комбинаций будет очень много — $2^{100} \approx 10^{30}$. При небольшом числе переменных, например, при $n = 2$, будет сгенерировано всего $2^2 = 4$ точки, что очень мало. В таком случае следует использовать промежуточные точки по каждой координате.

Для ремоделирования предлагается такой алгоритм:

1. Вычисление количества точек, приходящихся на одно измерение: $m_i = \lceil \sqrt[n]{m} \rceil$, где квадратные скобки означают выбор целой части, отбрасывание дробной.

2. Генерация по каждой координате множества равноудаленных друг от друга m_i точек, что может быть реализовано на Python с использованием функции `linspace` из пакета `numpy`.

3. Создание всех комбинаций точек из предыдущего пункта.

4. Определение оставшегося числа несгенерированных точек: $m := m - n \cdot m_i$.

5. Генерация множества случайных точек на основе равномерного закона распределения в области $[a_i, b_i]^n$ методом Монте-Карло.

2.3. Алгоритм на основе регулярной сетки

Алгоритм формирования множества данных на основе регулярной сетки похож во многом на метод Монте-Карло. Его задача так же состоит в максимально полном покрытии области определения функции f . При этом не требуется шаг дискретизации формируется автоматически, исходя из числа точек, которые требуется сгенерировать. Алгоритм представляется таким образом:

1. Генерация по каждой координате множества равнодistantных друг от друга m точек, что создает регулярную сетку по каждой координате с шагом $x_{ij} = \frac{(b_i - a_i)}{(m-1)}$.

2. Случайное перемешивание (перестановка) по каждой координате i множества точек x_{ij} , что на языке Python реализуется с использованием процедуры `random.shuffle` из пакета `numpy`.

Тем самым, в качестве строчек обучающего множества будут выступать точки из числа регулярно созданной сетки.

2.4. Адаптивный алгоритм

Основная идея данного алгоритма состоит в генерации большего количества точек в тех областях пространства изменения входных переменных, в которых больше область значений функции f с учетом размера этой области. Интуитивно это означает, что функция на этих участках сильнее возрастает или убывает, т.е. отражает идею более сложного поведения модели в этих участках. В основу разбиения исходной области на подобласти может использоваться процедура бисекции брусов (многомерных параллелепипедов), которая используется в алгоритмах интервального анализа [4]. Бисекция это разбиение области по середине i -й координаты на две новых подобласти. Получаемые подобласти будут одинакового объема (размера). Оценка образа изменения функции может осуществляться на основе функций включения — достаточно простых процедур для внешнего гарантированного оценивания диапазона изменения функции [4]. Однако, для упрощения может использоваться оценка на основе значений функции в нескольких генерированных с помощью метода Монте-Карло точек в каждой из подобластей. Можно предложить такой адаптивный алгоритм для создания множества данных в рекурсивной форме с учетом того, что задается еще и минимально генерируемое число точек в области m_r :

1. Если выполняется условие $m_r < m$, то генерируется в данной области m_r точек, процедура завершает работу, это множество возвращается в качестве результата.

2. Выбирается координата i^* , по которой изменение i -й входной переменной $b_i - a_i$ максимально. Если таких координат несколько может выбираться первая из них или случайная.

3. Производится разбиение области на две части путём бисекции по координате i^* .

4. В каждой из этих подобластей генерируется набор из $\frac{m_r}{2}$ точек (если значение m_r нечетно, то в одной из областей генерируется на одну точку меньше, это не существенно), в которых рассчитываются значения функции f .

5. Для каждой из этих двух подобластей определяются максимальные и минимальные значения, вычисляются диапазоны изменения функции в них $range_1$ и $range_2$.

6. Рассчитывается количество точек, которое необходимо генерировать в каждой из этих подобластей пропорционально диапазонам изменения функции с учетом того, что ряд точек уже генерирован: $k_1 = \left(m - \frac{m_r}{2} \right) \cdot \frac{range_1}{(range_1 + range_2)}$; $k_2 = m - k_1 - \frac{m_r}{2}$.

7. Рекурсивный вызов той же самой функции для генерирования двух множеств в каждой из подобластей с соответствующими значениями точек k_1 и k_2 .

8. Завершение работы процедуры и возврат в качестве ответа объединения четырех множеств в одно (два множества получились в ходе оценивания диапазонов функций в каждой из подобластей на шаге 4, а два — в ходе вызова функций на шаге 7).

3. Вычислительные эксперименты и анализ алгоритмов генерации множеств

Для проведения экспериментов было разработано программное обеспечение на языке Python, реализующее приведенные выше алгоритмы генерации множеств. В качестве известной модели f была выбрана функция из [5]:

$$f(x, y) = a \cdot \sin(\pi x^2) \cdot \sin(2\pi y),$$

которая рассматривалась на существенно более широком диапазоне $[-10; 10] \times [-10; 10]$ в отличие от $[-1; 1] \times [-1; 1]$ из [5]. Параметрическая идентификация новой модели содержала компоненты глобального и локального поисков. Вначале производился поиск на 10000 случайных значениях по методу Монте-Карло, затем применялся метод мультистарта из 20 стартовых точек и использованием метода BFGS для локальной оптимизации. Использовалась процедура `minimize` из пакета `scipy`. Для решения задачи ремоделирования была выбрана ошибка MSE и 4 разных функции. В качестве ремоделирующих рассматривались 4 модели, включая структуру исходной модели (z_1), линейную по параметрам модель (z_2), квадратичную модель без свободного члена (z_3), модель искусственного нейрона (z_4):

$$\begin{aligned} z_1 &= c_1 \cdot \sin(c_2 \cdot x^2) \cdot \sin(c_3 \cdot y); \\ z_2 &= c_0 + c_1 x + c_2 y; \\ z_3 &= c_1 x^2 + c_2 y^2; \\ z_4 &= c_0 + c_1 \cdot \frac{1}{1 + e^{-(c_2 + c_3 x + c_4 y)}}. \end{aligned}$$

Каждый параметр моделей рассматривался на диапазоне $[-10; 10]$. С помощью предложенных методов формировалось множество данных размерами 500, 1000, 1500, 2000 и 2500 строк. Частично фрагмент с результатами экспериментов, где отражены лучшие по качеству результаты, приведен в табл. 1.

Таблица 1

5 лучших результатов вычислительного эксперимента

Объем данных m	Алгоритм генерации множества	Модель z	Ошибка на обучающем множестве
1000	На основе регулярной сетки	z_1	1,0722e-11
500	Комбинаторный	z_1	0,01366
1000	Комбинаторный	z_1	0,02312
2500	Комбинаторный	z_1	0,02992
2500	Комбинаторный	z_4	0,03295

Таблица 2

5 худших результатов вычислительного эксперимента

Объем данных m	Алгоритм генерации множества	Модель z	Ошибка на обучающем множестве
1000	Монте-Карло	z_2	0,06357
1000	Монте-Карло	z_3	0,06389
1500	Комбинаторный	z_4	0,06870
1500	Комбинаторный	z_3	0,06871
1500	Комбинаторный	z_2	0,06871

Вычислительные эксперименты показали следующие результаты:

1) не при всех множествах данных были получены отличные результаты при идентификации самой модели z_1 , при этом не всегда увеличение объема множества приводило к лучшим результатам, что связано с важностью применения качественных методов параметрической идентификации;

2) комбинаторный алгоритм генерации данных для маленького числа переменных приводит к лучшим результатам, при этом худшие результаты также получены при комбинаторном алгоритме;

3) при больших объемах множества данных хорошие результаты показывает адаптивный алгоритм генерации множества, превосходит метод Монте-Карло по качеству генерируемого множества данных.

Заключение

Выбор объема множества данных в математическом ремоделировании вместе с выбором алгоритма для генерации самого множества являются важнейшими факторами для решения задачи. Данные параметры могут приводить к разным результатам, они могут рассматриваться как гиперпараметры ремоделирования и подбираться для каждой задачи отдельно. Вычислительные эксперименты показали целесообразность применения комбинаторного подхода к генерации множества данных для малого числа переменных, адаптивный метод имеет преимущество перед методом Монте-Карло.

Исследование выполняется за счет гранта РНФ № 24-21-00474 «Разработка и исследование методов и автоматизированной системы ремоделирования систем машинного обучения в анализе больших данных», <https://rscf.ru/project/24-21-00474>.

Литература

1. Сараев П. В., Блюмин С. Л., Галкин А. В. Нейросетевое и нейро-нечеткое ремоделирование в управлении металлургическими процессами // Материалы XIII Всероссийской научно-практической конференции с международным участием «Современные проблемы горно-металлургического комплекса. Наука и производство». ТОМ II. – Старый Оскол : СТИ МИСИС, 2016. – С. 102–105.

2. Huang C., Radi B., Hami A. E. Uncertainty analysis of deep drawing using surrogate model based probabilistic method // The International Journal of Advanced Manufacturing Technology. – 2016. – Vol. 86, Iss. 9. – P. 3229–3240.

3. Тихонов А. Н. О некорректных задачах линейной алгебры и устойчивом методе их решения // Доклады Академии наук СССР. – 1965. – Т. 163, № 3. – С. 591–594.

4. Жолен Л., Кифер М., Дири О., Вальтер Э. Прикладной интервальный анализ. – Москва, Ижевск : Институт компьютерных исследований, 2019. – 468 с.

5. Осовский С. Нейронные сети для обработки информации. – М. : Издательство «Финансы и статистика», 2004. – 343 с.

ПРИМЕНЕНИЕ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ДИАГНОСТИКИ ПНЕВМОНИИ НА ОСНОВЕ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Д. С. Сергеев

Воронежский государственный университет

Аннотация. В статье исследуется задача автоматической диагностики заболеваний органов грудной клетки по рентгеновским снимкам. В качестве исходного материала использовался набор данных, разделённый на две категории: снимки здоровых лёгких и изображения с признаками пневмонии. Представлены результаты обучения, анализ точности, а также реализованный веб-интерфейс для практического применения системы. Тестирование показало высокую точность классификации, что открывает перспективы применения метода для более сложных диагностических задач.

Ключевые слова: машинное обучение, глубокое обучение, сверточные нейронные сети, рентгенография, пневмония, медицинские изображения, ResNet50V2, компьютерное зрение.

Введение

В будущем из-за быстрого развития ИИ врачи могут ставить диагнозы иначе, ведь сверточные нейронные сети стали главным инструментом для работы со сложными медицинскими снимками. Было бы очень удобно, если бы программа могла сама находить различные болезни, например видеть проблемы с легкими или переломы костей, при этом делая это с высокой точностью. Это могло бы значительно снизить нагрузку на работников больниц.

Поэтому, развитие искусственного интеллекта и компьютерного зрения дает новые возможности для создания очень точных систем диагностики, но, чтобы сделать модель для нахождения переломов, нужно двигаться по шагам. Сначала лучше решать более простые задачи классификации. Это поможет отладить структуру и методы работы, при этом, нужно еще и сравнить разные сверточные нейронные сети, чтобы выбрать из них наилучшую.

Как первый шаг в большом исследовании по автоматизации диагностики проблем грудной клетки, было выбрана задача по нахождению пневмонии. Это потому, что есть доступные данные, задача бинарной классификации (пневмония есть или нет) относительно проста, и опыт, который будет получен, можно будет использовать для более сложной задачи, а именно нахождения переломов.

1. Материалы и методика исследования

В качестве базовой модели использовалась модель ResNet. Она представляет собой архитектуру CNN, разработанную для снижения влияния затухающего градиента в нейронных сетях и повышения точности результатов [1, 2].

Существует несколько вариантов архитектуры ResNet, одним из которых является ResNet50V2. В качестве базовой модели использовалась модель ResNet50v2, предварительно обученная на наборе данных, подобные обученные модели используют ранее изученные признаки из больших наборов данных, что может улучшить производительность и сократить время обучения на меньших специализированных наборах данных.

EfficientNet входит в семейство сверточных нейронных сетей и спроектирована для высокоэффективной классификации изображений. Архитектура EfficientNet-B0 представлена на рис. 2.

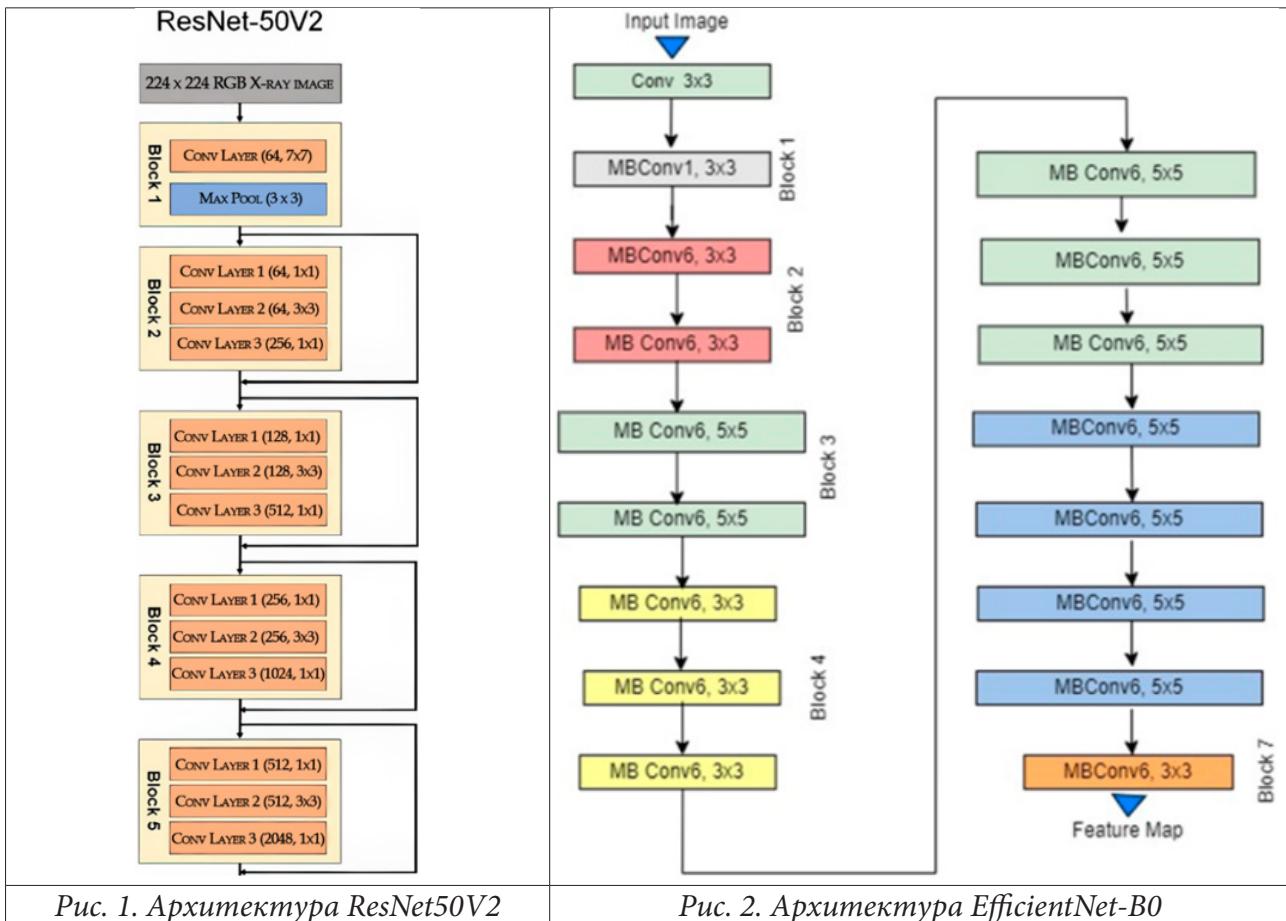


Рис. 1. Архитектура ResNet50V2

Рис. 2. Архитектура EfficientNet-B0

EfficientNet-B0 является базовой моделью линейки EfficientNet и продемонстрировала значительную эффективность в задачах классификации изображений, её архитектура включает метод масштабирования для согласованного изменения размерности и разрешения, обеспечивая современный уровень производительности, также данная модель поддается дообучению для конкретных задач классификации изображений.

Архитектура EfficientNet-B0 построена на основе компонентов мобильной инвертированной узкогорлой свертки (MBConv) с добавлением блоков сжатия и SE-блоков [3]. Такое решение эффективно максимизирует точность при минимизации количества параметров.

В архитектуре применяется глубинно-разделимая свертка, снижающая вычислительную сложность в k^2 раз, где k — размер ядра свертки. Архитектура также включает инвертированные остаточные блоки для сокращения числа обучаемых параметров. В модели EfficientNet-B0 осуществляется масштабирование глубины, ширины и разрешения для корректного масштабирования модели [4].

2. Результаты

В исследовании был выбран набор «Chest X-Ray Images (Pneumonia)», который разделен на тренировочную и тестовую выборки [5].

В каждой из них содержатся изображения, которые разделены на две категории: снимки здоровых легких и снимки с признаками пневмонии. Общий объем набора составляет около 2,5 ГБ.

График слева (рис. 3) показывает изменение точности модели в процессе обучения. Синяя линия на графике показывает точность на обучающей выборке, а оранжевая линия показыва-

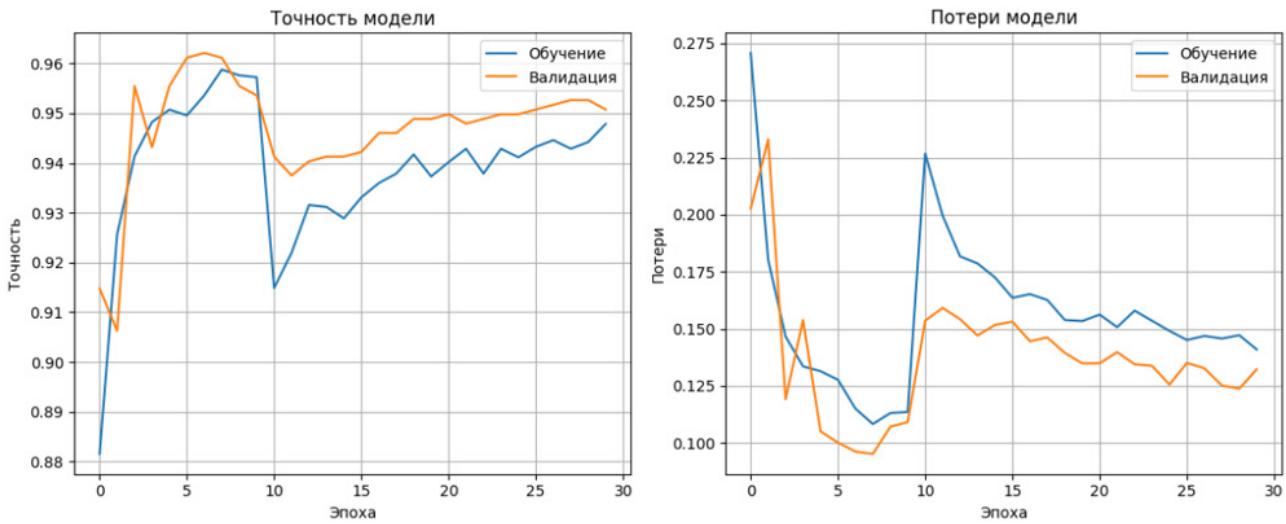


Рис. 3. Графики точность модели и потери модели ResNet50V2

ет точность на валидационной выборке. По оси X — эпохи обучения (от 0 до 30), по оси Y — значение точности (от 88 до 96 %).

График справа показывает изменение функции потерь. Синяя линия, это потери на обучающей выборке, а оранжевая линия, это потери на валидационной выборке. По оси X — эпохи обучения (от 0 до 30), а по оси Y — значение функции потерь (от 0.1 до 0.275)

Данные два графика демонстрируют, что модель достигла высокой точности (более 95 %) на валидационной выборке, также потери стабильно снижаются на протяжении обучения.

Модель была обучена за 156 минут и 44 секунды (2 часа, 36 минут), это довольно долго, ибо модель обучалась на CPU.

Самая точная валидационная точность, которая была достигнута в процессе обучения равняется 93.58 %.

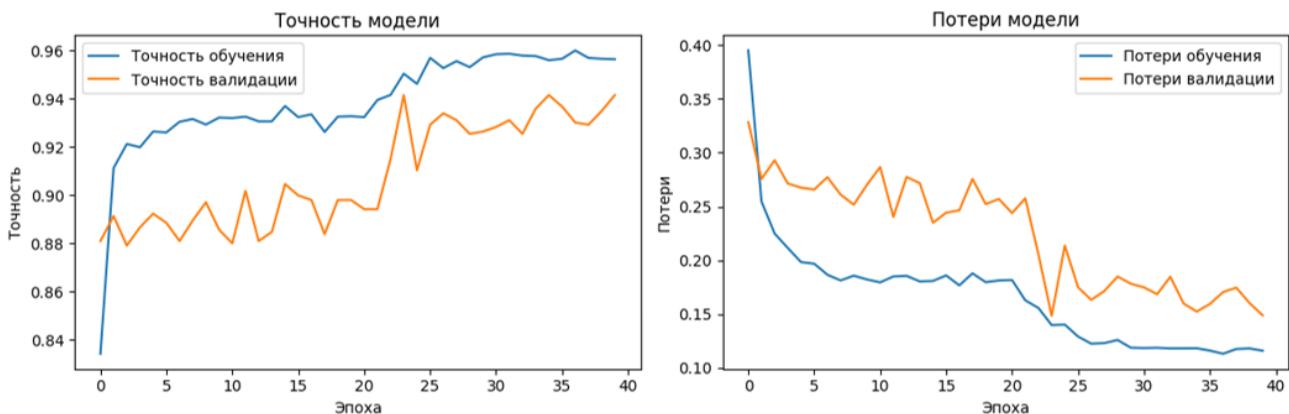


Рис. 4. Графики точность модели и потери модели EfficientNet-B0

На графике видно, что точность растет до примерно 96 %, а валидационная до 94 %. Также видно, что нет переобучения, так как нет признаков взрывного роста валидационных потерь.

Если сравнивать графики у EfficientNet-B0 и ResNet50V2, то можно сделать вывод, что EfficientNet-B0 показывает более стабильное и контролируемое обучение. В это же время, ResNet50V2 демонстрирует хорошую точность в начале, но теряет устойчивость после 8 эпохи.

На графике видно, что EfficientNet-B0 демонстрирует высокую эффективность, модель правильно классифицировала 176 случаев для метки «NORMAL», а также правильно классифицировал 373 случая для метки «PNEUMONIA». Модель не идеально определяется «NORMAL», но в медицинском контексте лучше иметь больше ложны тревог, чем пропущенных диагнозов,

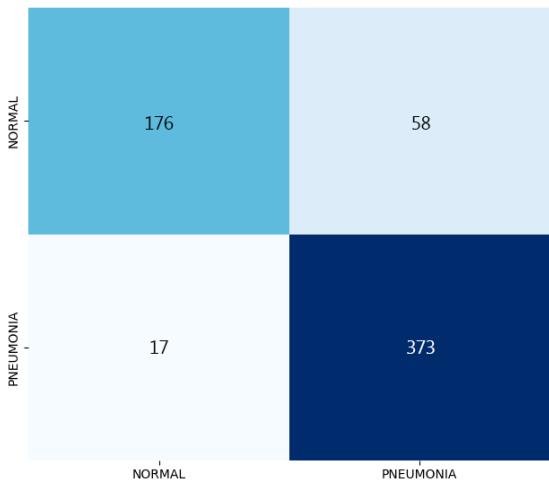


Рис. 5. Матрица ошибок EfficientNet-B0

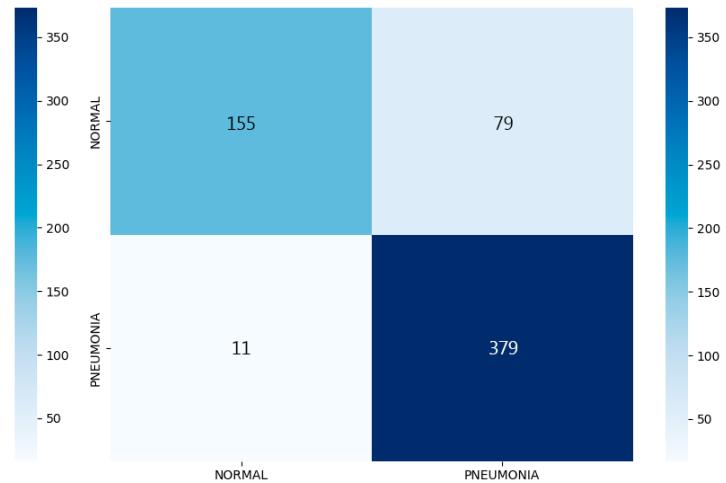


Рис. 6. Матрица ошибок ResNet50V2

то есть лучше ошибиться и сказать, что пневмония есть, когда ее нет, чем сказать, что пневмонии нет, когда она есть. Возможно, что проблема состоит в самом наборе данных, на которых обучалась модель.

Сравнивая матрицы ошибок ResNet50V2 и EfficientNet-B0, можно сделать вывод, что EfficientNet-B0 ошибается меньше на классе NORMAL, чем ResNet50V2. Обе модели показывают хорошие результаты с обнаружением, но EfficientNet-B0 продемонстрировала более стабильное и контролируемое обучение на протяжении всех эпох, она показывает плавный рост точности без признаков переобучения.

Таблица 1

Результаты обучения моделей

Модель	Класс	Точность	Полнота	F1-score
EfficientNet-B0	NORMAL	0.91	0.75	0.82
	PNEUMONIA	0.87	0.96	0.91
ResNet50V2	NORMAL	0.93	0.66	0.77
	PNEUMONIA	0.83	0.97	0.89

Полученные результаты показывают, что модель EfficientNet-B0 демонстрирует более высокие и стабильные показатели по всем ключевым метрикам по сравнению с ResNet50V2 и является более предпочтительным вариантом для практического применения.

Заключение

В ходе сравнения удалось подтвердить, что современные сверточные нейронные сети, такие как ResNet50V2 и EfficientNet-B0, могут демонстрировать высокую точность при решении задачи автоматического выявления пневмонии по рентгеновским снимкам. Важно отметить, что набор данных при этом содержал снимки различного качества, что значительно усложняло задачу, тем не менее обе архитектуры показали способность извлекать устойчивые признаки. Обе модели достигли результатов выше 95% на валидационных данных, однако EfficientNet-B0 показала более стабильное поведение во время обучения и более ровный рост точности, что делает её предпочтительным вариантом для прикладного использования.

Полученные результаты указывают на перспективность применения подобных архитектур в других медицинских системах, что в дальнейшем может использоваться для обнаружения более сложных патологий, а также применена в качестве удобного инструмента для медицинского персонала.

Литература

1. *Kaviya P. A Unified Framework for Monitoring Social Distancing and Face Mask Wearing Using Deep Learning: An Approach to Reduce COVID-19 Risk* / P. Kaviya, P. Chitra, B. Selvakumar // Procedia Computer Science. – 2023. – Vol. 218. – P. 1561–1570.
2. *He K. Deep Residual Learning for Image Recognition* / K. He, X. Zhang, S. Ren, J. Sun // arXiv preprint arXiv:1512.03385. 2015.
3. *Hu J. Squeeze-and-Excitation Networks* / J. Hu, L. Shen, G. Sun, S. Albanie, E. Wu // arXiv preprint arXiv:1709.01507. 2017.
4. *Tan M. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks* / M. Tan, Q. V. Le // arXiv preprint arXiv:1905.11946. 2019.
5. *Mooney P. Chest X-Ray Images (Pneumonia)* / P. Mooney // Kaggle. 2018. [Электронный ресурс]. URL: <https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>.

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДИАГНОСТИКИ ОПУХОЛЕЙ НА МРТ СНИМКАХ ГОЛОВНОГО МОЗГА

С. С. Сергеев

Университет Иннополис

Аннотация. В работе рассматривается задача детекции опухолей на МРТ снимках головного мозга. Для ее решения рассмотрены методы Faster R-CNN и SIFT-CNN. Предложены и протестированы модификации SIFT-CNN, направленные на улучшение качества оценки в ситуациях, где обучение на дескрипторах неэффективно. В ходе исследования наиболее подходящей для решения задачи стала модификация на основе графических HOG дескрипторов, которая показала сбалансированные результаты метрик mAR и mAP по сравнению со стандартной моделью и SIFT, SURF модификациями. На основе выбранной модели реализована REST API-система и предложены практические сценарии её использования.

Ключевые слова: детекция объектов, машинное обучение, REST API, МРТ, опухоль головного мозга, мониторинг и оповещение, разработка систем.

Введение

Компьютерное зрение — область искусственного интеллекта, направленная на интерпретацию визуальной информации с изображений и видеоматериалов. Продукты, использующие достижения области компьютерного зрения, активно внедряются во все сферы общества.

Одним из перспективных направлений является здравоохранение, где компьютерное зрение применяется для диагностики заболеваний по данным КТ, МРТ, рентгенографии и видеозаписям [1–3]. Такой подход повышает точность диагноза и сокращает время его установления. Это, в свою очередь, позволяет выявлять заболевания на ранних стадиях и увеличивает вероятность благоприятного исхода, а также частично компенсирует дефицит кадров.

В данной статье рассматривается метод классификации изображений SIFT-CNN, предлагаются методы, основанные на схожей концепции, проведён эксперимент по задаче детекции опухолей на МРТ-снимках с использованием алгоритма Faster R-CNN. На базе наиболее эффективного алгоритма построена система с архитектурой REST API, обеспечивающая возможность переобучения модели на новых данных, а также функции мониторинга и автоматического оповещения о диагностических результатах.

1. Описание применённых моделей

В качестве модели детекции объектов выбран Faster R-CNN, состоящий из каркасной и головной частей. Каркас представлен свёрточной нейронной сетью (CNN), извлекающей карты признаков из изображений. Головная часть отвечает за обнаружение объектов на полученных картах и реализуется через три компонента:

1. Region Proposal Network выделяет регионы с предполагаемым расположением объектов [4].

2. ROI pooling формирует признаки фиксированного размера из карт регионов [5].

3. Классификатор определяет классы по вычисленным признакам.

Алгоритм обеспечивает отличную точность классификации и локализации и устойчив к масштабированию, ротации и окклюзии. Точность, однако, существенно зависит от выбранного каркаса.

В качестве каркаса изначально планировалось использовать архитектуру SIFT-CNN [6], основанную на обучении сверточной сети на SIFT-дескрипторах [7]. При таком подходе обучение на стандартных дескрипторах невозможно, поскольку число точек интереса различается для каждого изображения. Поэтому автор метода [7] применил плотную реализацию алгоритма, в которой дескриптор вычисляется для каждой точки. Несмотря на возросшие вычислительные затраты, такая реализация совместно с ResNet-18 показала более высокие результаты сопоставимые с ResNet-50 [6].

В задаче детекции опухолей данный подход с ResNet-18 продемонстрировал результаты, примерно в два раза хуже, чем у модели, обученной на оригинальных изображениях. В связи с этим была предложена альтернативная схема, при которой уже ResNet-50 обучается одновременно на изображении и дескрипторах, формируя входной сигнал из двух каналов — одноканального изображения и сжатые дескрипторы. Сжатый дескриптор представляет собой среднее всех элементов исходного дескриптора. Такой подход снижает локальную информативность, но при этом позволяет использовать дескриптор без потерь в качестве общей оценки.

Предложенную схему удалось применить также к альтернативным дескрипторам: SURF [8] и HOG [9]. Для SURF использован метод сжатия дескриптора, а для HOG — его графическое представление из библиотеки scikit-learn.

2. Компьютерный эксперимент

2.1. Описание экспериментальных данных и эксперимента

В качестве экспериментальных данных используются изображения пользователя “Python bots2021” с roboflow.com. В ходе исследования установлено, что снимки идентичны другим наборам данных, но не имеют модифицированных изображений, что позволило составить пользовательский набор аугментаций. Набор состоит из 3038 рисунков различного размера опухолей и разделены на следующие четыре класса:

1. No_tumor — Опухоль отсутствует.
2. Glioma — Глиома.
3. Meningioma — Менингиома.
4. Pituitary — Аденома гипофиза.

Эксперимент состоит из двух этапов: обучение модели и её оценка на тестовых данных. Для обучения использовано 3200 изображений, 1090 из которых получены с помощью дополнения данных, валидационная и тестовая выборки включают 600 и 328 снимков соответственно. Все изображения предварительно нормализованы с помощью метода оценки плотности ядра [10].

Испытание проведено с моделью ResNet-50, предобученной на наборе данных ImageNet. Дообучение производилось на видеокарте RX 6800 XT с 16 Гб видеопамяти с применением платформы ROCm при следующих параметрах обучения:

- Оптимизатор — AdamW.
- Количество эпох — 10.
- Планировщик. Основной — CousineAnnealingLR, холодный старт — LinearLR.
- Batch_size — 18.
- Learning_rate — 0.0001.
- Weight_decay — 0.0002.

2.2. Метрики

В задачах классификации с локализацией и детекции объектов для проверки правильности положения ограничивающей рамки используется отношение площадей ограничивающих рамок:

$$IoU = \frac{S(A \cap B)}{S(A \cup B)}, \quad (1)$$

где A и B — предсказанная и настоящая тестовые рамки [11]. При IoU равном 0 рамки не пересекаются, при 1 рамки наложены идеально. Для данной задачи применены следующие модификаторы метрик mAP и mAR [11]:

1. По порогу IoU .
 - a. 0.5:0.05:0.95 — среднее mAP и mAR , полученных на каждом значении IoU в данном интервале.
 - b. 0.5.
 - c. 0.75.
2. По размеру рамки.
 - a. Малая — площадь меньше $32^2 px^2$.
 - b. Средняя — площадь в диапазоне от 32^2 до $96^2 px^2$.
 - c. Большая — площадь больше $96^2 px^2$.
3. По максимуму детекций на изображение.
 - a. На 1.
 - b. На 10.

Для итоговой модели рассчитывается мульти классовая матрица ошибок.

2.3. Результаты эксперимента

Результаты эксперимента приведены в табл. 1. Анализ полученных данных показал, что наименее эффективным оказался алгоритм SURF, который показал снижение средней точности на 0.06. Метод HOG продемонстрировал наилучшие значения метрики mAR и занял промежуточное положение по показателю mAP между алгоритмами SIFT и базовой реализацией. Алгоритм SIFT обеспечил умеренное повышение общей точности и значительное улучшение при $IoU = 0.75$ для объектов среднего размера, однако привёл к снижению значений mAR и mAP для малых объектов.

Таблица 1

Результаты испытания модификаций алгоритма

Метрика	Обычный	SIFT	SURF	HOG
$mAP 0.5:0.05:0.95 $ все	0.424	0.456	0.380	0.447
$mAP 0.5 $ все	0.860	0.878	0.751	0.860
$mAP 0.75 $ все	0.372	0.450	0.325	0.398
$mAP 0.5:0.05:0.95 $ малые	0.410	0.319	0.242	0.395
$mAP 0.5:0.05:0.95 $ средние	0.423	0.464	0.391	0.448
$mAP 0.5:0.05:0.95 $ большие	0.811	0.796	0.654	0.789
$mAR 0.5:0.05:0.95 $ на 1	0.511	0.550	0.471	0.529
$mAR 0.5:0.05:0.95 $ на 10	0.526	0.561	0.482	0.550
$mAR 0.5:0.05:0.95 $ малые	0.513	0.430	0.348	0.566
$mAR 0.5:0.05:0.95 $ средние	0.525	0.566	0.487	0.547
$mAR 0.5:0.05:0.95 $ большие	0.820	0.800	0.720	0.820

В совокупности полученные результаты позволяют заключить, что модификация на основе дескрипторов HOG является наиболее сбалансированным решением с точки зрения компромисса между точностью и устойчивостью. Матрица ошибок модификации приведена в табл. 2, при этом общая точность классификации составляет 92 %.

Таблица 2

Матрица ошибок модификации НОГ

Истинные	Предсказанные				
	Класс	Без опухолей	Глиома	Менингиома	Аденома гипофиза
Без опухолей	29	7	2	0	
Глиома	0	69	8	0	
Менингиома	1	0	95	1	
Аденома гипофиза	1	1	2	88	

3. Описание разработанной системы

Приложение реализовано в виде сервиса, взаимодействие с которым осуществляется через REST API. Такой архитектурный подход обеспечивает возможность автономного функционирования системы без непосредственного участия оператора. Например, МРТ-терминал может в процессе диагностики передавать снимки пациента, которые обрабатываются системой в реальном времени; затем терминал получает результаты предсказаний и отображает их на экране оператора либо аккумулирует серию предсказаний и связанных метрик, формируя вердикт по завершении обследования. Аналогично может быть организована потоковая диагностика, подобная процедуре флюорографии, когда пациент проходит обследование, а результаты автоматически поступают врачу. Три типовых сценария работы системы представлены на рис. 1.



Компоненты программной инфраструктуры проекта разворачиваются в изолированных Docker-контейнерах, объединенных общей сетью, обеспечивающей их взаимодействие. Управление контейнерами осуществляется с использованием файла docker-compose, позволяющий оперативно развернуть все элементы системы.

Инфраструктурная схема представлена на рис. 2, на ней представлены следующие компоненты:

1. Celery — процесс фреймворка Celery, предназначенный для выполнения задач в реальном времени и фоновом режиме. Данный процесс отвечает за выполнение запланированных задач по обучению модели, генерации предсказаний по пакетам фотографий и за выполнение срочных предсказаний.

2. Flower — админ панель и средство мониторинга за задачами в реальном времени для Celery.

3. Redis — брокер сообщений для Celery.

4. Postgres — контейнер объектно-ориентированной СУБД PostgreSQL, хранящей в себе информацию о поступивших пакетах изображений, токены аутентификации, данные фреймворка Django.

5. Web — приложение на REST API архитектуре, разработанное с помощью фреймворков Django и Django Rest Framework на фреймворке Django, реализующее интерфейс взаимодействия с системой.

6. Nginx — веб сервер, принимающий запросы пользователей и передающий их контейнеру web на обработку, может использоваться для настройки ssl сертификатов, обеспечивающих безопасное соединение с сервером, обслуживания статических файлов или балансирования нагрузки.

7. Loki — система хранения, индексирования и обработки журналов.

8. Promtail — приложение, собирающее с локального расположения журналов и передающее в Loki. Используется для сбора журналов контейнеров и Nginx.

9. Prometheus — система сбора и хранения метрик внутри базы данных временных рядов. Собирает метрики с приложений-экспортёров.

10. Grafana — система визуализации данных и оповещения о событиях. Используется для отображения журналируемых данных и собранных метрик.

11. Alertmanager — система оповещения о показателях системы. Через telepush отправляет информацию в telegram.

12. Telepush — telegram echo-бот, отвечающий за перенаправление сообщений системы в telegram чат.

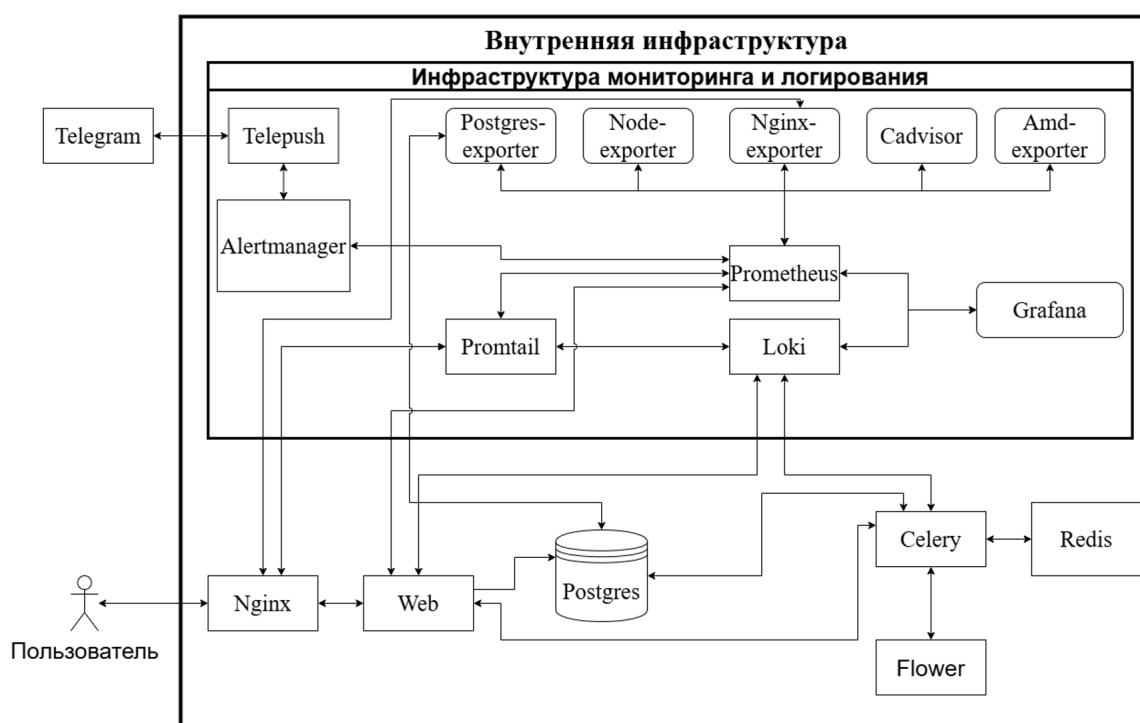


Рис. 2. Инфраструктурная схема системы

13. Telegram — мессенджер, предоставляющий API для взаимодействия с чатами и управления ботами внутри него.
14. Postgres-exporter — система сбора метрик с PosgtreSQL.
15. Node-exporter — система сбора метрик о состоянии сервера.
16. Nginx-exporter — система сбора метрик Nginx.
17. Cadvisor — система сбора метрик docker контейнеров.
18. Amd-exporter — система сбора метрик для видеокарт AMD.

Заключение

В статье предложен метод на основе архитектуры SIFT-CNN и протестирован совместно с алгоритмами SIFT, SURF и HOG на задаче детекции опухолей на МРТ снимках мозга. В результаты эксперимента показали, что модели, базирующиеся на SIFT и HOG, позволяют получить прирост по некоторым критериям, взамен возможного ухудшения по другим. Учитывая данное свойство, метод применим как альтернатива базовым моделям. Для разработанного Web API выбран метод на основе HOG, с общим процентом обнаружения опухолей равным 92, так как предоставляет наиболее сбалансированные результаты. Разработанная система следует стандартам современной web разработки, обеспечивая возможности масштабируемости, журналирования и мониторинга.

В дальнейшем приложение планируется усовершенствовать за счёт внедрения альтернативных методов детекции объектов, что позволит повысить точность и скорость обработки. Также планируется внедрение автоматического подбора параметров обучения, что облегчит настройку модели. Дополнительно планируется исследовать применение метода главных компонент для снижения размерности дескрипторов SIFT, что может улучшить эффективность и уменьшить вычислительные затраты.

Литература

1. *Ashames M. M., Ergin S.* Lung Cancer Tumor Classification With CNN& HOG-SVM Combination [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/367452817_LUNG_CANCER_TUMOR_CLASSIFICATION_WITH_CNN_HOG-SVM_COMBINATION (дата обращения: 07.04.2025).
2. *Babu G. N. K., Peter V. J.* Skin Cancer Detection Using Support Vector Machine With Histogram of Oriented Gradients Features // Ictact Journal on Soft Computing, – 2021. – No 2. – P. 2301–2305.
3. *Kavin Kumar D., Gowsihan R., Naveen A., Praveen B., Ragul K.* Challenges in Information, Communication and Computing Technology. – 1 изд. – London : Taylor & Francis, 2024. – 892 p.
4. *Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun.* Faster R-CNN: Towards Real-Time Object-Detection with Region Proposal Networks [Электронный ресурс]. – Режим доступа: <https://arxiv.org/pdf/1506.01497> (дата обращения: 07.04.2025).
5. *Sumit Bisht, Sunita Joshi, Urvi Rana* Comprehensive Review of R-CNN and its Variant Architectures // International Research Journal on Advanced Engineering Hub (IRJAEH). – 2024. – No 4. – P. 959–966.
6. *Tsourounis D., Kastaniotis D., Theoharatos C., Kazantzidis A., Economou G.* SIFT-CNN: When Convolutional Neural Networks Meet Dense SIFT Descriptors for Image and Sequence Classification // Journal of imaging. – 2022. – No 10. – P. 1–18.
7. *Alcaraz-Chavez J. E., Téllez-Anguiano A.d.C., Olivares-Rojas J. C., Martínez-Parrales R.* Real-Time Tracking and Detection of Cervical Cancer Precursor Cells: Leveraging SIFT Descriptors in Mobile Video Sequences for Enhanced Early Diagnosis [Электронный ресурс]. –Режим доступа: <https://www.mdpi.com/1999-4893/17/7/309> (дата обращения: 07.04.2025).

8. *Bay H., Tuytelaars T., Van Gool L.* SURF: Speeded Up Robust Features // Computer Vision – ECCV 2006 . – Heidelberg: Springer-Verlag, 2006. – P. 404–417.
9. *Dalal N., Triggs B.* Histograms of oriented gradients for human detection // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). – Los Alamitos: IEEE, 2005. – P. 886–893.
10. *Reinhold J. C., Dewey E. B., Carass A., Prince J. L.* Evaluating the Impact of Intensity Normalization on MR Image Synthesis [Электронный ресурс]. – Режим доступа: <https://arxiv.org/pdf/1812.04652.pdf> (дата обращения: 07.04.2025).
11. *Rafael Padilla, Sergio L. Netto, Eduardo A. B. da Silva* A Survey on Performance Metrics for Object-Detection Algorithms [Электронный ресурс]. – Режим доступа: https://www.researchgate.net/publication/343194514_A_Survey_on_Performance_Metrics_for_Object-Detection_Algorithms (дата обращения: 07.04.2025).

СОВРЕМЕННЫЕ МЕТОДЫ ОБРАБОТКИ ИЗОБРАЖЕНИЙ НА ОСНОВЕ ГИБРИДНЫХ МОДЕЛЕЙ ГЛУБОКИХ НЕЙРОННЫХ СЕТЕЙ

А. А. Сирота, Р. Р. Отырба, Н. И. Бережнов

Воронежский государственный университет

Аннотация. Рассматриваются основные проблемы, возникающие при создании и применении современных архитектур глубокого обучения (CNN и ViT) в задачах обработки изображений. Определены направления совершенствования моделей глубокого обучения для снятия существующих ограничений. В рамках указанных направлений представлены результаты исследований, содержащие новые гибридные архитектуры, которые эффективно сочетают преимущества свёрточных сетей и трансформеров для решения целого ряда критически важных задач компьютерного зрения, таких как: классификация, семантическая сегментация, обнаружение изменений, а также разработка методов аугментации для повышения устойчивости к сложным, аппликативным помехам. В ходе многочисленных экспериментов показано, что предложенные гибридные модели демонстрируют конкурентоспособную или превосходящую точность при значительно меньших вычислительных затратах и меньшем числе параметров. Таким образом, работа вносит вклад в создание более эффективных и устойчивых архитектур глубокого обучения для широкого спектра задач обработки изображений.

Ключевые слова: обработка изображений, глубокие нейронные сети, трансформеры, свёрточные сети, гибридные архитектуры, алгоритмы аугментации, аппликативные помехи.

Введение

В последние годы методы обработки изображений достигли значительных успехов благодаря применению глубоких нейронных сетей. Современные модели строятся на принципе автоматического извлечения признаков изображения различного уровня абстракции. Эти многоуровневые иерархические представления признаков обеспечивают универсальную основу для решения разнообразных задач обработки изображений, таких как классификация объектов, детекция, семантическая сегментация, восстановление изображений и обнаружение изменений. На сегодняшний день современные методы обработки изображений широко используют свёрточные нейронные сети (CNN) и визуальные трансформеры (ViT).

Исторически CNN впервые продемонстрировали возможность эффективно извлекать иерархические признаки из изображений, начиная от простых границ и текстур на первых слоях и заканчивая сложными объектными представлениями на глубоких слоях. Благодаря своей архитектуре, включающей свёртки, пулинг и нелинейные функции активации, такие сети позволяют автоматически выявлять значимые признаки для решения задач компьютерного зрения без необходимости их ручного формирования. Однако основной недостаток данного подхода связан с локальностью свёрточных фильтров: CNN плохо учитывают глобальный контекст, что часто является критически важным для понимания объектов, находящихся в различных частях сцены.

1. Анализ современных направлений исследований и решаемые задачи

На фоне стремительного прогресса в обработке естественного языка и появления архитектуры трансформеров [1] в последние годы наблюдается значительный рост интереса к адаптации трансформерной архитектуры, включая механизмы самовнимания, изначально разработанные для языковых моделей, к задачам анализа изображений. Такой переход стал возможен

благодаря тому, что самовнимание позволяет гибко учитывать глобальные взаимосвязи между различными областями входных данных — то, с чем свёрточные сети справляются ограниченно из-за локальности своих фильтров. С момента появления первого Vision Transformer (ViT) [2] трансформерные архитектуры зарекомендовали себя как мощный инструмент для решения широкого спектра задач компьютерного зрения (КЗ).

Однако, несмотря на значительные успехи, современные модели трансформеров всё ещё имеют ряд ограничений.

1. Трансформеры демонстрируют снижение эффективности при работе с небольшими наборами данных. Это связано с тем, что они не обладают так называемой индуктивной предвзятостью (априорными предположениями о характере данных и их зависимостях). В отличие от СНС, которые используют локальные свёртки для обработки взаимосвязей между пикселями, трансформеры полагаются на механизм многоголового самовнимания (Multi-Head Self-Attention, MHSA), требующий значительного объёма данных для обучения. Всё это делает их склонными к переобучению в условиях малой обучающей выборки.

2. Рост точности трансформеров обычно достигается путём увеличения размеров моделей и глубины архитектуры, что приводит к существенному росту вычислительных затрат. Это затрудняет их использование в условиях ограниченных вычислительных возможностей.

3. Трансформеры обрабатывают признаки в рамках определённых масштабов на каждом уровне иерархии сети, но не учитывают возможную вариативность масштабов объектов в рамках слоя внимания. Это приводит к неспособности эффективно захватывать информативные признаки объектов различного размера, а также снижает эффективность голов MHSA, поскольку они будут склонны выявлять схожие зависимости, что уменьшает разнообразие внимания и, в свою очередь, вызывает избыточность вычислений [3].

4. Механизм самовнимания часто формирует множество нерелевантных связей между токенами, порождая избыточное внимание. Кроме того, его квадратичная сложность по числу токенов существенно повышает вычислительные затраты при работе с изображениями высокого разрешения. Эти факторы снижают эффективность моделей и повышают риск переобучения. Попытки упростить внимание уменьшают гибкость модели и могут привести к пропуску важных токенов (ключей и значений).

5. Недавние исследования показывают, что свёрточные нейронные сети также могут достигать конкурентоспособных показателей эффективности в задачах обработки изображений, о чём свидетельствуют результаты, представленные в таких работах, как ConvNeXt [4], MSCAN [5], RepLKNet [6] и SLaK [7].

На практике эти ограничения усугубляются в реальных сценариях применения (например, в системах видеонаблюдения, аэрокосмического мониторинга, медицинской диагностики и др.), где изображения часто содержат аппликативные помехи, такие как атмосферные осадки, артефакты или другие сложные воздействия, что значительно затрудняет извлечение информативных визуальных признаков для решения задач КЗ. В таких условиях особенно важно использовать модели и алгоритмы аугментации, способные генерировать синтетические обучающие данные с реалистичными искажениями, отражающими реальные условия съёмки. Это позволяет воспроизводить погодные эффекты, шумы сенсоров и другие виды помех без необходимости ручного сбора данных в каждом возможном сценарии. Благодаря этому удается компенсировать нехватку или несбалансированность исходных выборок и повысить устойчивость моделей к вариативности входных данных и сложным внешним воздействиям.

В связи с изложенным в качестве важного направления исследований и разработок в области КЗ, как нам представляется, является создание новых гибридных архитектур глубоких нейронных сетей, сочетающие преимущества CNN и ViT, с одновременной модернизацией механизмов внимания для снижения переобучения и вычислительной сложности. В этом плане центральное место занимает научное обоснование принципов построения таких моделей,

особое внимание уделяется кодирующей части, которая служит для извлечения информативных визуальных признаков и может применяться универсально для решения различных задач компьютерного зрения, включая классификацию, сегментацию, восстановление и обнаружение новизны.

В рамках этих исследований нами видится важным решение трех основных научных задач.

Первая задача направлена на теоретическое обоснование новых методов структурной регуляризации механизма внимания, основанных на использовании мультиплексивных и аддитивных стохастических компонентов в трансформерных модулях, создание методов и алгоритмов деформируемого и прореженного внимания, обеспечивающих использование наиболее информативных участков изображения. Новизна заключается в создании алгоритмов, которые выявляют и интегрируют информативные фрагменты в модули внимания напрямую в процессе обработки тензорных потоков.

Вторая задача направлена на создание и исследование новых гибридных архитектур для решения комплекса задач компьютерного зрения: классификации, семантической сегментации и обнаружения новизны. Ядром моделей-кодировщиков станет свёрточный модуль многомасштабного восприятия с параллельной многоветвевой структурой, а также эффективные канальные и пространственные механизмы деформируемого и прореженного внимания. Такой подход позволит объединить сильные стороны CNN — в извлечении локальных признаков разных масштабов, и ViT — в захвате глобальных зависимостей данных, обеспечивая тем самым более качественное извлечение информативных визуальных признаков, что в свою очередь позволит повысить качество анализа изображений.

Третья задача связана с созданием нового подхода к аугментации изображений с целью повышения устойчивости моделей при обработке данных в условиях аппликативных помех, например, атмосферных осадков или артефактов. Для этого могут использоваться гибридные модели с модулем перекрестного внимания для извлечения помех (например, осадков или артефактов) зашумленного изображения и их включении в исходное изображение. Новизна такой постановки заключается также в комплексном подходе, в котором обучение моделей для аугментации и восстановления проводится совместно по циклической схеме, что обеспечивает высокую реалистичность при переносе помех.

2. Предлагаемые модели гибридных нейронных сетей

2.1. Гибридная архитектура для задач классификации и сегментации

Предлагаемая архитектура SegTwice представляет собой гибридное решение, объединяющее трансформеры и свёрточные нейронные сети в рамках общей структуры кодер-декодер (рис. 1).

В основе предлагаемого подхода лежит стремление использовать сильные стороны обоих методов: свёрточные нейронные сети — для эффективного извлечения локальных признаков, а трансформеры — для захвата глобальных зависимостей. Введение свёрточных слоёв дополнительно усиливает индуктивную предвзятость сети, что способствует повышению её эффективности и снижению риска переобучения в условиях малой выборки.

В рамках данной архитектуры предлагается новая кодирующая сеть TWICE-DA (от англ. Transformer With Integrated Multi-Scale Convolutional Extractor and Deformable Attention) с иерархической структурой из 4 уровней (рис. 2).

Гибридность архитектуры достигается путём введения блока трансформера, включающего следующие новые ключевые архитектурные решения:

1. Модуль многомасштабного восприятия (Multi-Scale Perception Unit, MSPU), использует несколько параллельных свёрточных ветвей для извлечения признаков в разных масштабах.

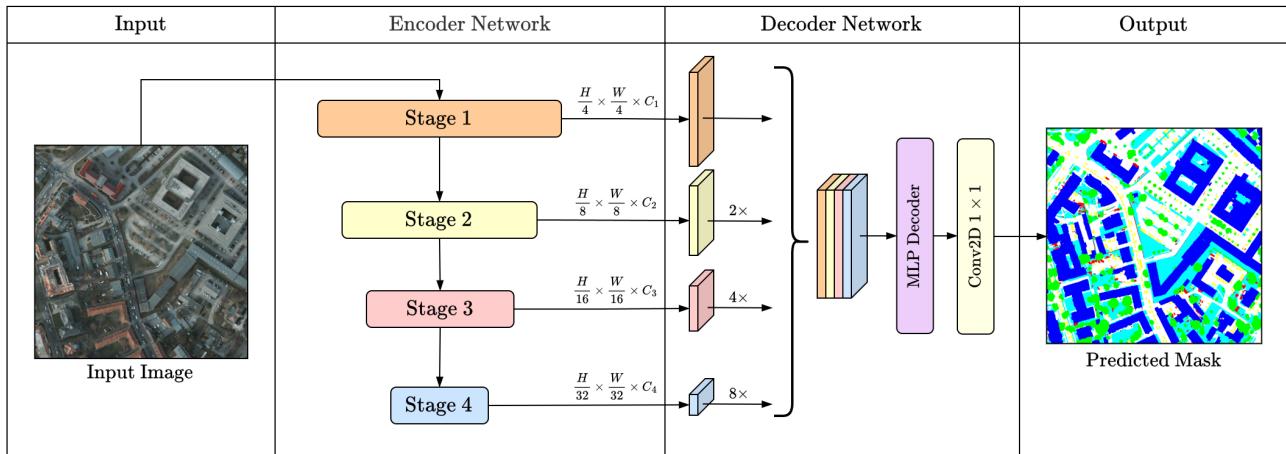


Рис. 1. Общий вид предлагаемой архитектуры SegTwice

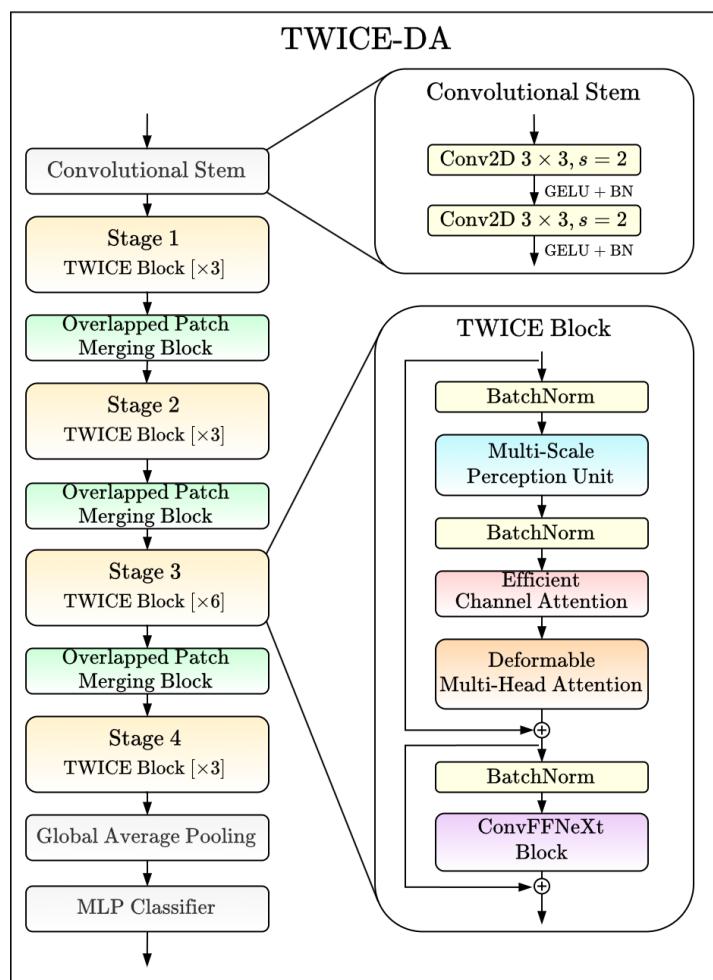


Рис. 2. Архитектура сети TWICE-DA

Размещается перед механизмом MHSA, чтобы усилить позиционную информацию о положении объектов на изображении.

2. Эффективный модуль канального внимания (Effecient Channel Attention, ECA), динамически выделяет наиболее информативные каналы, усиливая важные признаки и подавляя менее значимые.

3. Модуль деформируемого многоголового внимания (Deformable Multi-Head Attention, DMHA) динамически фокусируется только на определённых релевантных областях изображе-

ния, уменьшая таким образом вычислительную сложность модели и улучшая качество внимания. Если традиционный MHSA учитывает всевозможные попарные связи между пикселями, то DMHA производит выбор ограниченного числа ключевых точек для извлечения ключей Q и значений V , чье расположение определяется обучаемыми смещениями.

4. Модуль свёрточной сети прямого распространения (Convolutional Feed-Forward Network, ConvFFNeXt) обеспечивает канальное взаимодействие признаков, улучшая их информативность и выраженность. Для этого реализуется облегчённая версия FFN, в которой перед полно связанными слоями добавляется depthwise-свёртка.

Нами были проведены эксперименты на задаче классификации изображений для оценки обобщающей способности и эффективности извлечения признаков TWICE-DA на данных разной сложности CIFAR-100 и Caltech-256 (табл. 1). Установлено, что TWICE-DA, обладая меньшим количеством обучаемых параметров и вычислительной сложностью, превосходит большинство современных моделей, незначительно уступая только MSCAN-S. Трансформеры MiT-B1, Swin-T, Twins-SVT-S, показывают более низкие результаты, что подтверждает сложность их обучения на относительно небольших и несбалансированных наборах данных.

Таблица 1

Сравнительный анализ точности TWICE-DA и других современных моделей

Модель	Параметры	FLOPs	CIFAR-100	Caltech-256
EfficientNetV2-S	20,5M	2,8G	78,49	72,87
ConvNeXt-T	28,0M	4,4G	74,37	64,44
MSCAN-S	13,6M	2,6G	81,00	75,91
MiT-B1	13,3M	1,6G	77,77	63,49
Swin-T	27,7M	4,3G	76,21	63,92
Twins-SVT-S	23,7M	2,8G	75,94	63,13
CvT-13	19,7M	4,0G	77,10	66,43
TWICE-DA	13,1M	1,8G	80,98	74,41

Тестирование предлагаемой архитектуры SegTwice для задачи семантической сегментации проводилось на популярных аэрокосмических наборах данных LoveDA и Potsdam. Результаты тестирования представлены в табл. 2 и 3. Исходя из проведённых экспериментов, можно сделать вывод, что предлагаемая архитектура SegTwice не уступает в точности традиционным моделям и современным трансформерам, демонстрируя высокие и конкурентоспособные показатели при значительно меньшем количестве обучаемых параметров.

Таблица 2

Сравнительный анализ точности SegTwice и других современных моделей на LoveDA

Модель	Кодер	Параметры	Классы (IoU)							mIoU
			1	2	3	4	5	6	7	
DeepLabV3+	ResNet50	39,6M	43,0	50,9	52,0	74,4	10,4	44,2	58,5	47,62
HRNet	HRNet-W48	75,9M	44,6	55,3	57,4	74,0	11,1	45,3	60,9	49,79
SegFormer	MiT-B1	13,7M	42,2	56,4	50,7	78,5	17,2	45,2	53,8	49,14
UperNet	Swin-T	60,0M	43,3	54,3	54,3	78,7	14,9	45,3	59,6	50,00
AerialFormer-T	Swin-T	42,7M	45,2	57,8	56,5	79,6	19,2	46,1	59,5	52,00
UperNet	ViT-L12×4	80,6M	46,2	60,6	57,3	76,9	16,1	47,5	62,2	52,38
MTP	InternImage-XL	335,0M	46,8	62,6	59,0	82,3	17,5	47,6	63,4	54,17
SegTwice	TWICE-DA	13.5M	43,3	56,2	56,8	79,2	15,2	44,4	63,6	51,27

Таблица 3

Сравнительный анализ точности SegTwice и других современных моделей на Potsdam

Модель	Кодер	Параметры	Классы (F1-Score)						mF1	mIoU
			1	2	3	4	5	6		
DeepLabV3+	ResNet50	39,6M	89,3	92,8	83,3	78,4	88,2	31,6	77,3	66,8
DANet	ResNet18	12,6M	88,5	92,7	78,8	85,7	73,7	43,2	77,1	65,3
SCAttNet V2	ResNet50	26,6M	90,0	94,0	84,1	79,8	89,1	33,6	78,4	68,3
SegFormer	MiT-B1	13,7M	92,9	96,4	86,9	88,1	95,2	58,9	86,4	78,0
UperNet	Swin-T	60,0M	93,5	97,0	87,4	88,6	96,1	56,9	86,6	78,5
AerialFormer-T	Swin-T	42,7M	93,5	96,9	87,2	89,0	95,9	62,5	87,5	79,5
SegTwice	TWICE-DA	13,5M	91,1	95,6	85,1	86,0	91,1	52,2	83,5	74,0

Следует подчеркнуть, что представленные результаты были получены без предварительного обучения TWICE на ImageNet, а также с использованием простейшего MLP декодера (с 0,5M параметров). Однако, несмотря на это, модель успешно справляется как глобальный, так и локальный контекст, эффективно сегментируя как крупные объекты, так и мелкие детали.

2.2. Сиамская гибридная архитектура трансформера для обнаружения изменений

В ходе наших исследований в области создания гибридных архитектур предложена легковесная сиамская архитектура X-ChangeNet на основе иерархического трансформера с усовершенствованным механизмом сопоставления разновременных признаков (рис. 3).

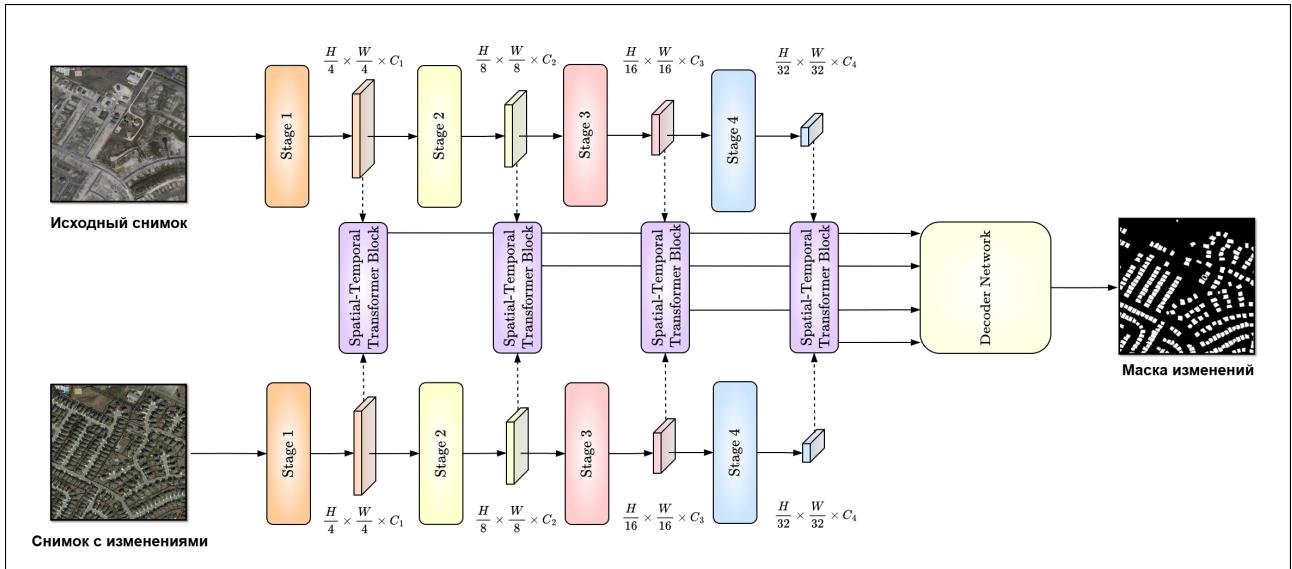


Рис. 3. Общий вид предлагаемой архитектуры X-ChangeNet

Ключевым нововведением в предлагаемой архитектуре X-ChangeNet является механизм сопоставления STTB, который реализует выявление различий между разновременными признаками (рис. 4).

STTB состоит из трёх последовательно расположенных модулей, каждый из которых играет специфическую роль в формировании финальных карт различий.

Модуль многомасштабной попарной корреляции (Multi-Scale Pairwise Correlation Unit, MSPCU) выполняет выявление базовых структурных различий между признаками через по-

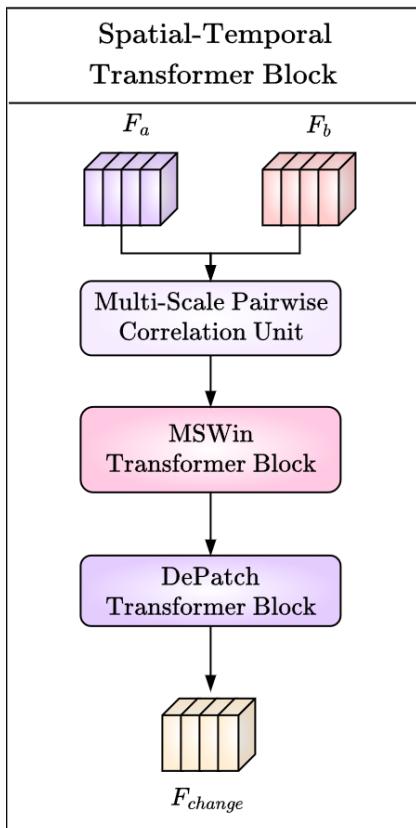


Рис. 4. Модуль сопоставления разновременных признаков STTB

парную конкатенацию и многомасштабную групповую свёртку, обеспечивая пространственно-временной анализ и фокусировку трансформера на наиболее информативных областях.

Многомасштабный трансформер на основе локальных окон (Multi-Scale Window Transformer Block, MSWin Block) усиливает значимые признаки и моделирует сложные локальные и региональные зависимости с помощью механизма многомасштабного оконного внимания, который параллельно обрабатывает несколько масштабов для более точного анализа признаков.

Деформируемый трансформер на основе патчей (Deformable Patch Transformer Block, DePatch Block) применяет деформируемое внимание на уровне патчей, позволяя фокусироваться на значимых структурных изменениях, снижать вычислительную нагрузку и сохранять глобальный контекст.

Тестирование предлагаемой архитектуры X-ChangeNet для задачи обнаружения элементов новизны проводилось на популярных аэрокосмических наборах данных LEVIR-CD и CDD Dataset (season-varying). Результаты тестирования представлены в табл. 4.

Исходя из результатов проведённых экспериментов, можно сделать вывод, что предлагаемая архитектура X-ChangeNet демонстрирует высокие и конкурентоспособные показатели точности, превосходя большинство современных моделей при значительно меньшем количестве обучаемых параметров, что подтверждает эффективность предлагаемого механизма сопоставления STTB.

Таблица 4

Сравнительный анализ моделей на наборах данных LEVIR-CD и CDD Dataset

Модель	Параметры	LEVIR-CD	CDD Dataset
FC-EF	1,4M	83,40	59,20
FC-Siam-diff	1,4M	86,31	69,20
FC-Siam-conc	1,5M	85,86	63,70
UNet++_MOSF	11,0M	85,86	88,31
SNUNet-CD	12,0M	88,16	95,30
IFN	36,0M	88,13	90,30
DASNet	16,3M	79,91	91,19
STANet	12,2M	87,26	91,50
BIT	4,0M	89,31	94,61
ChangeFormer	41,02M	90,40	94,63
FTN	–	91,01	–
Changer	11,3M	92,33	–
TinyCD	0,3M	91,05	–
SMBCNet	10,1M	90,87	–
LightCDNet-large	2,8M	91,43	–
SiamixFormer-5	175.2M	91,58	94,51
X-ChangeNet	5,8M	91,91	97,81

2.3. Модель WeatherTransformer для синтеза условно-реальных изображений на основе трансформера с перекрестным вниманием

Предлагаемая схема модели WeatherTransformer реализует метод «экстракции-включения», который осуществляет перенос applicативных помех и других видов искажений естественного и искусственного происхождения с одного изображения на другое. В ходе исследований фокус был направлен, прежде всего, на синтезировании изображений сцен в условиях различных атмосферных осадков, что часто мешает эффективному применению систем видеонаблюдения.

При подготовке обучающей выборки для выделения помех и нанесения ее на изображение-шаблон проводится решение обратной задачи — восстановления зашумленных изображений с помощью модели SwinIR или другой продвинутой модели. Такой подход позволяет решить проблему отсутствия парных данных. Имея только зашумленное изображение, можно получить восстановленное изображение и шаблон. Архитектура WeatherTransformer сочетает в себе сильные стороны сверточных слоев (локальная инвариантность и иерархичность представлений) и трансформеров (глобальное внимание и адаптивность к контексту). Модель принимает на вход два изображения: исходное чистое изображение и изображение-шаблон (референс) с примером целевого погодного эффекта. На выходе формируется исходное изображение с наложенным эффектом погодного шаблона.

Общая архитектура включает три основных блока: сверточный кодер, трансформер с перекрестным вниманием и сверточный декодер (рис. 5). Кроме того, используется позицион-

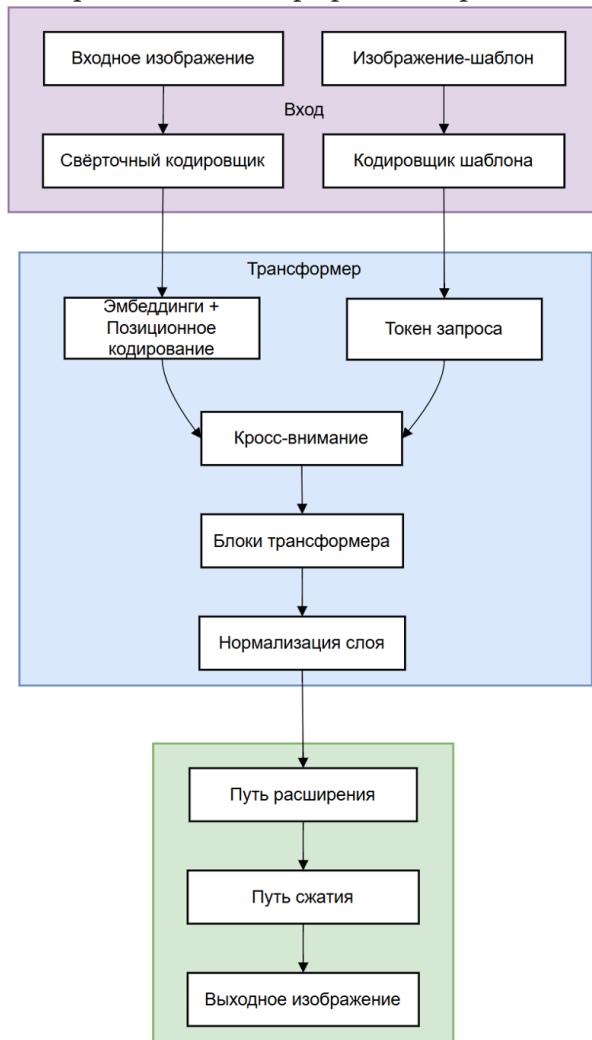


Рис. 5. Общий вид архитектуры WeatherTransformer

ное кодирование для сохранения пространственных связей между частями изображения. Для обучения модели в работе предложено использовать составную функцию потерь `weatherLoss`, учитывающую контентное, структурное и перцептуальное сходство:

$$L_{total} = 0.7L_{content} + 0.2L_{struct} + 0.1L_{perc}.$$

В табл. 5 приведены показатели степени реалистичности получаемых синтетических изображений, полученные при обучении на данных их датасета All-Weather, включающего в себя изображения с дождем (Raindrop), снегом (Snow100K) и туманом (Outdoor-Rain) с соответствующими эталонами; всего около 20 тысяч обучающих изображений и 1 тысяча тестовых. На рис. 6 даны примеры синтезированных изображений. Из результатов видно, что при синтезе искажений модель `WeatherTransformer` улучшила метрики по сравнению с известными архитектурами.

Таблица 5

Сравнение методов при синтезе изображений в плохих погодных условиях

Метод	PSNR (↑)	SSIM (↑)	FID (↓)
CycleGAN	21.604	0.639	47.073
HRIGNet	18.595	0.747	32.111
WeatherTransformer	21.88	0.78	30.47
WeatherTransformer	22.30	0.79	29.80



Рис. 6. Примеры обработки изображений с помощью `WeatherTransformer`
а) — исходное изображение, б) — референс, в) — синтезированное изображение

Заключение

В рамках данной работы были исследованы современные методы обработки изображений на основе глубоких нейронных сетей, выявлены ключевые ограничения существующих архитектур (CNN и ViT). В ответ на эти вызовы были разработаны и экспериментально апробированы новые гибридные архитектуры. Эти модели, сочетающие преимущества CNN и ViT, направлены на повышение эффективности и устойчивости систем компьютерного зрения (КЗ) для решения целого ряда критически важных задач, таких как: классификация, семантическая

сегментация, обнаружение изменений, а также разработка методов аугментации для повышения устойчивости к аппликативным помехам. Полученные результаты демонстрируют конкурентоспособную или превосходящую точность наших гибридных моделей при значительно меньших вычислительных затратах и параметрической сложности.

Исходя из полученных результатов, в ходе дальнейших исследований целесообразно сосредоточиться на масштабировании и унификации разработанных подходов. В первую очередь необходимо провести комплексную оптимизацию гиперпараметров архитектуры TWICE-DA, затем осуществить реализацию нескольких вариантов моделей TWICE-DA различного размера — Tiny (T), Small (S) и Large (L), что позволит адаптировать сеть под разные сценарии применения. Для оценки и дальнейшего повышения обобщающей способности этих моделей планируется проведение их предварительного обучения на крупном наборе данных ImageNet. Завершающим этапом станет ввод TWICE-DA в качестве кодирующей сети для архитектур X-ChangeNet и WeatherTransformer с целью унификации и повышения эффективности извлечения визуальных признаков.

Литература

1. Attention is all you need / A. Vaswani, N. M. Shazeer, N. Parmar [и др.] // arXiv Preprint. – 2017. – URL: <https://arxiv.org/pdf/1706.03762> (дата обращения: 25.11.2025).
2. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale / A. Dosovitskiy, L. Beyer, A. Kolesnikov [и др.] // arXiv Preprint. – 2020. – URL: <https://arxiv.org/pdf/2010.11929> (дата обращения: 25.11.2025).
3. EfficientViT: Memory Efficient Vision Transformer with Cascaded Group Attention / X. Liu, H. Peng, N. Zheng [и др.] // 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2023. – P. 14420–14430.
4. A ConvNet for the 2020s / Z. Liu, H. Mao, C. Y. Wu [и др.] // 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022. – P. 11966–11976.
5. SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation / M. Guo, C. Lu, Q. Hou [и др.] // arXiv Preprint. – 2022. – URL: <https://arxiv.org/pdf/2209.08575> (дата обращения: 25.11.2025).
6. Scaling Up Your Kernels to 31×31: Revisiting Large Kernel Design in CNNs / X. Ding, X. Zhang, J. Han, G. Ding // 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2022. – P. 11953–11965.
7. More ConvNets in the 2020s: Scaling up Kernels Beyond 51x51 using Sparsity / S. Liu, T. Chen, X. Chen [и др.] // arXiv Preprint. – 2022. – URL: <https://arxiv.org/pdf/2207.03620> (дата обращения: 25.11.2025).

РОЛЬ ИНСТРУМЕНТОВ ИНТЕЛЛЕКТУАЛЬНОГО СОПРОВОЖДЕНИЯ СОТРУДНИКОВ

В. В. Смоленцева¹, Н. Н. Тетерин², Е. Г. Бергер²

¹Воронежский государственный университет инженерных технологий

²МИРЭА – Российский технологический университет

Аннотация. В работе рассмотрен комплексный анализ роли инструментов интеллектуального сопровождения в современных организациях в контексте коммуникативных платформ трудового взаимодействия. В условиях растущей роли информационных технологий в организациях и управлении персоналом, проблема организационного взаимодействия сотрудников, развитие компетенций и оптимизации производительности труда является актуальной. Представлены практические решения позволяющие автоматически анализировать коммуникации сотрудников и выстраивать персональные траектории развития на основе данных коммуникативных платформ. Практическая значимость исследования заключается в интеллектуализации процессов управления персоналом за счет использования ситуационного ИИ-центра сопровождения сотрудников.

Ключевые слова: интеллектуальное сопровождение, производительность труда, организационное взаимодействие, компетенции сотрудников, управление персоналом, ситуационный ИИ-центр.

Введение

Современные организации сталкиваются с необходимостью объективной оценки эффективности работы сотрудников. Существующие варианты оценки производительности не в полной мере отражают полноту анализа в контексте причин, связанных с взаимодействием сотрудников организации. Особую значимость приобретают вопросы использования человеческого капитала, что обусловлено ростом конкуренции на трудовом рынке. Актуальность исследования обусловлена наличием существенного разрыва между технологическими возможностями современных платформ трудового взаимодействия и их практическим применением в управленческой деятельности. Включение ситуационных ИИ-центров позволит перейти к объективной оценке эффективной работы персонала [1].

В табл. 1 приведена классификация инструментов интеллектуального сопровождения в современных организациях в контексте корпоративных платформ, представленных в категории инструментов.

Таблица 1

Классификация инструментов интеллектуального сопровождения

Категория инструментов	Основные функции	Примеры реализации
Аналитические платформы	Мониторинг производительности труда	Анализ выполнения КПИ, отслеживание рабочего времени
Системы развития компетенций	Персональная траектория обучения, оценка навыков	Рекомендация обучающего контента, курсы повышения квалификации
Платформы коммуникаций	Анализ взаимодействий между сотрудниками, выявление деструктивного поведения	Анализ тональности сообщений, анализ социальных сетей сотрудников

Объединение вышеописанных направлений в единый ситуационный ИИ-центр позволяет достичь синергетического эффекта, когда результаты каждого модуля усиливают и дополняют друг друга [2].

На рис. 1 представлен концептуальный подход к работе ситуационного ИИ-центра сопровождения сотрудников.

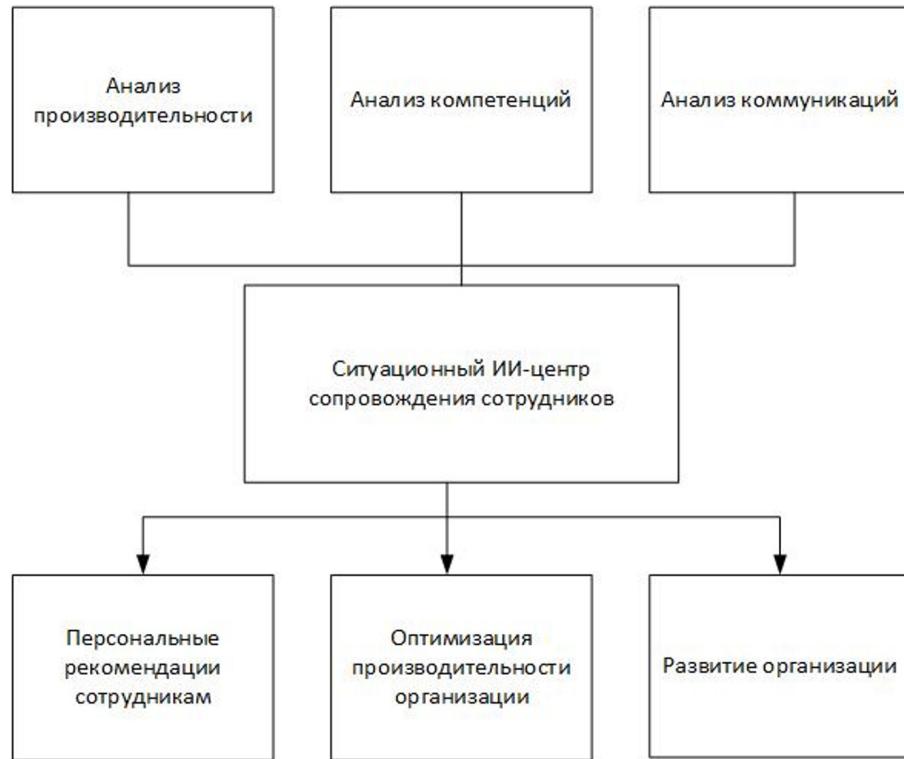


Рис. 1. Ситуационный ИИ-центр сопровождения сотрудников

Ситуационный ИИ-центр анализирует данные из всех модулей и формирует дальнейшие пути развития, учитывающие взаимосвязь между производительностью, компетенциями и коммуникациями сотрудников. Например, если в социальных сетях обнаружены следы деструктивного поведения, то с сотрудником будет проведена профилактическая беседа и предложение пройти корпоративный тренинг по развитию коммуникативных навыков [3].

Компонентами ситуационного центра являются гибридные архитектуры семантического анализа и обработки естественного языка при взаимодействии с сотрудниками организаций. Генерация сценариев может осуществляться архитектурами нейронных сетей и классификация вариантов через алгоритмы машинного обучения [4,5].

В табл. 2 приведена планируемая эффективность трудовой деятельности за счет внедрения ситуационного ИИ-центра в деятельность организаций.

Таблица 2

Планируемое повышение качества и эффективности ведения трудовой деятельности за счет внедрения ситуационного ИИ-центра

Показатель	Основные эффекты
Производительность труда	Рост производительности труда
Обучение новым компетенциям	Рост получаемых новых компетенций
Коммуникации сотрудников	Рост качества коммуникаций персонала
Вовлеченность сотрудников в деятельность организаций	Своевременное выявление и решение проблем

В рамках дальнейшего развития предусмотрено развертывание единого дашборда с визуализацией ключевых показателей, объединение с HR-системами и корпоративными платформами, а также интеллектуальную систему рекомендаций для принятия кадровых решений [6]. Ключевыми причинами для реализации данных улучшений являются: потребность в консолидации данных; снижение когнитивной нагрузки; повышение точности кадровых решений.

Заключение

Рассмотренный вариант классификации инструментов интеллектуального сопровождения и структурное представление компонентов схемы ситуационного ИИ-центра с идентификацией показателей эффективности взаимодействия может иметь прикладное значение при организации взаимодействия сотрудников с последующей поддержкой принятия решений руководством организаций. Данный подход позволяет создать устойчивую систему управления персоналом, где анализ производительности, развитие компетенций и оптимизация коммуникаций создают синергетический эффект для организаций, обеспечивая долгосрочную конкурентоспособность на рынке.

Дальнейшие исследования могут быть направлены на внедрение ситуационных ИИ-центров в различные организационные структуры.

Литература

1. Салухов В.И. Образовательная компонента в формировании и становлении системы распределенных ситуационных центров и центров компетенции / В. И. Салухов, Б. В. Соколов // Стратегические приоритеты. – 2017. – № 2(14). – С. 138–147.
2. Григорьев П. В. Проблемы текущего развития системы распределенных ситуационных центров / П. В. Григорьев // Вестник ЦЭМИ. – 2024. – Т. 7, № 2.
3. Тетерин Н. Н. Концептуальный подход классификации деструктивного поведения с применением технологий искусственного интеллекта / Н. Н. Тетерин, В. В. Смоленцева // Актуальные проблемы прикладной математики, информатики и механики: сборник трудов Международной научной конференции, Воронеж, 02–04 декабря 2024 года. – Воронеж : Научно-исследовательские публикации, 2025. – С. 326–329.
4. Смоленцева Т. Е. Формирование методологической и технологической концепции эффективной обратной связи в цифровой образовательной среде // «Экономика и качество систем связи» 2025. – № 3. – С. 121–134.
5. Смоленцева В. В. Анализ алгоритмов поведения персонала структурных подразделений организаций в ЧС // Актуальные проблемы деятельности подразделений уголовно-исполнительской системы: Сборник материалов Всероссийской научно-практической конференции. В 3-х томах, Воронеж, 24 октября 2024 года. – Воронеж : ООО Издательско-полиграфический центр «Научная книга», 2024. – С. 92–96.
6. Елизаров А. М. Цифровые платформы и цифровые научные библиотеки / А. М. Елизаров, Е. К. Липачев // International Journal of Open Information Technologies. – 2020. – Т. 8, № 11. – С. 80–90.

РАЗРАБОТКА ЧАТ-БОТА ДЛЯ СБОРА И КЛАССИФИКАЦИИ ТУРИСТИЧЕСКИХ ОБЪЕКТОВ

М. А. Студеникин, Е. М. Аристова

Воронежский государственный университет

Аннотация. В статье рассматривается разработка Telegram-бота для сбора и систематизации сведений о туристических объектах. Актуальность проекта связана с ростом потребности в удобных сервисах для быстрого получения структурированной туристической информации. Целью работы является создание эффективного инструмента, способствующего популяризации локального туризма. В рамках исследования представлена архитектура системы на основе Python и фреймворка aiogram, а также базы данных SQLite для хранения информации об объектах. Описывается реализованный функционал, включающий категоризацию мест, предоставление детальных описаний с фотографиями и интеграцию с API Яндекс.Карт. В работе обосновывается эффективность выбранного технологического стека для создания информационных сервисов в туристической отрасли.

Ключевые слова: чат-бот, Telegram Bot API, туризм, информационная система, Python, aiogram, SQLite, база данных, интерактивный гид.

Введение

Современный этап развития информационных технологий характеризуется повсеместным проникновением мессенджеров в повседневную жизнь. Такая платформа, как Telegram, стала не только средством коммуникации, но и мощным инструментом для автоматизации сервисов и предоставления услуг. Одной из перспективных областей применения является создание чат-ботов — программ, имитирующих человеческое общение и выполняющих полезные функции для пользователя.

В контексте туристической отрасли существует устойчивая потребность в оперативном получении структурированной и достоверной информации о достопримечательностях, инфраструктуре и услугах. Традиционные методы (буклеты, сайты) зачастую неудобны для использования «на ходу». Чат-бот, интегрированный в популярный мессенджер, позволяет решить эту проблему, предоставляя пользователю интуитивно понятный интерфейс для поиска интересующих мест, не выходя из мессенджера.

Целью данной работы является разработка и реализация чат-бота, который объединяет информацию о туристических местах Воронежа.

Для достижения поставленной цели необходимо было решить следующие задачи: спроектировать структуру базы данных для хранения информации о туристических объектах; реализовать модуль взаимодействия с базой данных; разработать пользовательский интерфейс бота, включая систему команд и интерактивных меню; интегрировать функционал для предоставления детальной информации о местах, включая фотографии и построение маршрутов; а также обеспечить корректную обработку пользовательских запросов и ошибок.

1. Теоретическая часть

1.1. Чат-боты как класс программного обеспечения

Чат-бот — это программный агент, способный к диалогу с пользователем на естественном языке или с помощью структурированных команд в текстовом канале. В основе работы

большинства современных ботов лежит архитектура «клиент-сервер», где клиентом выступает мессенджер, а серверная логика реализована на стороне разработчика. Telegram предоставляет для этого мощный Bot API, позволяющий получать сообщения от пользователей и отправлять им ответы, используя HTTP-запросы.

1.2. Технологический стек разработки

Для реализации бота был выбран язык программирования Python [4,5], что обусловлено его простотой, читаемостью и наличием богатой экосистемы библиотек. Ключевой библиотекой является aiogram [2] — современный асинхронный фреймворк для Telegram Bot API. Асинхронность позволяет боту эффективно обрабатывать запросы от множества пользователей одновременно без блокировки основного потока выполнения, что критически важно для производительности.

В качестве системы управления базами данных была выбрана SQLite [3]. Это обосновано ее легкостью, встраиваемостью, отсутствием необходимости в отдельном серверном процессе и достаточной производительностью для задач проекта с невысокой нагрузкой. SQLite идеально подходит для прототипирования и для проектов, где все данные хранятся локально в одном файле.

1.3. Архитектура разрабатываемого решения

Архитектура бота следует классической трехзвенной схеме:

1. Уровень представления (Presentation Layer): интерфейс взаимодействия в Telegram, включающий кнопки, команды и сообщения.
2. Уровень бизнес-логики (Business Logic Layer): ядро бота, написанное на Python с использованием aiogram. Этот уровень обрабатывает команды, формирует запросы к базе данных и генерирует ответы.
3. Уровень данных (Data Layer): база данных SQLite, отвечающая за надежное хранение информации о туристических местах.

2. Проектирование и реализация

2.1. Проектирование и реализация базы данных

Для хранения информации о туристических объектах была разработана реляционная модель данных. В ее основе лежит таблица *places*, структура которой представлена в табл. 1:

Структура таблицы *places*

Таблица 1

Название поля	Тип данных	Описание
<i>id</i>	INTEGER	Первичный ключ, уникальный идентификатор записи.
<i>name</i>	TEXT	Название места.
<i>description</i>	TEXT	Развернутое текстовое описание места.
<i>photo_path</i>	TEXT	Путь к файлу с фотографией места в файловой системе сервера.
<i>address</i>	TEXT	Физический адрес места.
<i>category</i>	TEXT	Категория места.

Данная структура позволяет гибко управлять контентом: добавлять новые места, редактировать существующие и осуществлять выборку по категориям. Для обеспечения целостности данных поле `id` объявлено как PRIMARY KEY AUTOINCREMENT, а для оптимизации запросов на выборку по категориям рекомендуется создать индекс по полю `category`.

2.2. Реализация модуля работы с базой данных

Взаимодействие с базой данных осуществляется с помощью стандартной библиотеки Python `sqlite3` [3]. В коде бота реализованы две ключевые функции, инкапсулирующие запросы к базе данных.

Функция `get_places_by_category(category)` предназначена для получения списка всех мест, принадлежащих к определенной категории. Она выполняет SQL-запрос `SELECT id, name FROM places WHERE category = ?`, который возвращает идентификаторы и названия мест, отсортированные по алфавиту. Использование параметризованного запроса (?) предотвращает возможность SQL-инъекций.

Функция `get_place_by_id(place_id)` используется для получения полной информации о конкретном месте по его идентификатору. Запрос `SELECT name, description, photo_path, address FROM places WHERE id = ?` извлекает все необходимые данные для формирования карточки объекта.

Обе функции обеспечивают корректное открытие и закрытие соединения с базой данных, что гарантирует отсутствие утечки ресурсов.

2.3. Реализация пользовательского интерфейса и логики бота

Логика бота построена на обработчиках сообщений и callback-запросов от онлайн-кнопок, что предусмотрено архитектурой Telegram Bot API [1].

Обработчик команды /start. При получении команды `/start` бот отправляет приветственное сообщение и открывает главное меню, реализованное в виде `ReplyKeyboardMarkup`. Это меню содержит четыре кнопки с категориями: «Парки», «Памятники», «Кафе», «Отели». Использование Reply-клавиатуры значительно упрощает взаимодействие для пользователя, избавляя его от необходимости запоминать команды [1].

Обработчик выбора категории. Когда пользователь нажимает на одну из кнопок категорий, срабатывает обработчик, который с помощью словаря `category_map` преобразует текст кнопки в системное имя категории (например, «Парки» в «park» и аналогично с другими категориями). Затем вызывается функция `get_places_by_category(category)`. Если для выбранной категории есть объекты, бот динамически формирует онлайн-клавиатуру (`InlineKeyboardMarkup`), где каждая кнопка соответствует одному объекту. Это демонстрирует принцип адаптивного интерфейса, который формируется на основе содержимого базы данных.

Обработчик выбора конкретного места. При нажатии на онлайн-кнопку с названием объекта генерируется callback-запрос с данными в формате `place_{id}`. Обработчик извлекает `id` объекта, запрашивает полную информацию о нем с помощью функции `get_place_by_id(place_id)` и формирует ответное сообщение.

Сообщение включает:

- название и адрес, оформленные с помощью HTML-тегов (``, `<i>`) для лучшей визуализации;
- текстовое описание;
- фотографию, если файл по указанному в базе данных пути существует (проверка `os.path.exists(photo_path)`). Для отправки фото используется `FSInputFile` из библиотеки `aiogram` [2];

- онлайн-кнопку «Открыть маршрут», которая предоставляет прямую ссылку на Яндекс.Навигатор. Ссылка формируется динамически путем подстановки адреса объекта в URL-шаблон. Это ключевая интеграция с внешним сервисом, которая значительно повышает практическую ценность бота, так как позволяет пользователю мгновенно перейти к построению маршрута.

2.4. Асинхронная модель выполнения

Все обработчики в боте объявлены как асинхронные функции (async def). Запуск бота осуществляется с помощью `dp.start_polling(bot)` в асинхронном цикле, предоставляемом фреймворком aiogram [2]. Это означает, что бот не блокируется при выполнении «долгих» операций, таких как обращение к базе данных или отправка медиафайлов. Вместо этого он переключается на обработку других событий, что обеспечивает высокую отзывчивость системы даже при увеличении числа пользователей.

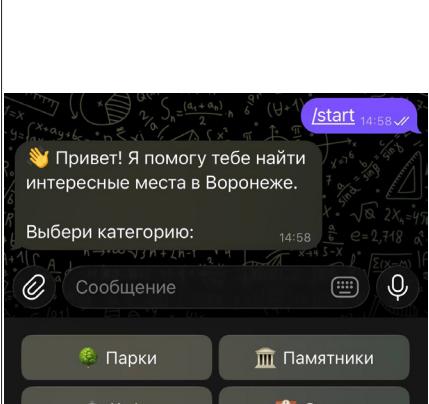
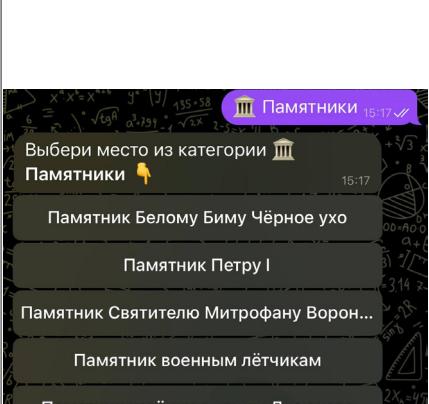
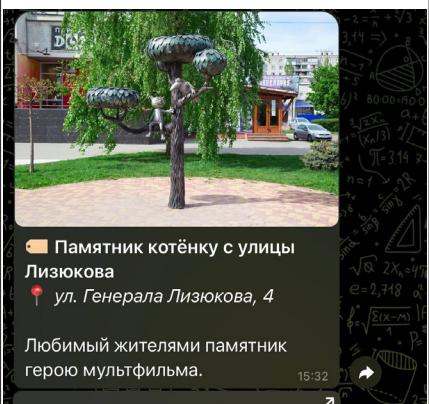
3. Апробация разработанного чат-бота

Для проверки работоспособности разработанного чат-бота и демонстрации его функциональности был проведен ряд тестов. Работа бота была развернута на сервере и протестирована в реальной среде мессенджера Telegram.

Взаимодействие с пользователем начинается с команды `/start`, на которую бот отвечает приветственным сообщением и отображает главное меню в виде Reply-клавиатуры (рис. 1). Данный подход интуитивно понятен и сразу направляет пользователя к основному сценарию использования — выбору категории мест.

После выбора категории, бот динамически формирует и отправляет онлайн-клавиатуру, содержащую список всех доступных мест в данной категории (рис. 2). Список извлекается из базы данных в реальном времени, что обеспечивает актуальность информации и позволяет легко масштабировать систему, добавляя новые места без изменения кода бота.

Ключевым элементом бота является карточка места (рис. 3). При выборе конкретного места из списка пользователь получает исчерпывающую информацию: название, адрес, описание и фотографию. Интеграция с Яндекс.Навигатором через кнопку «Открыть маршрут» предоставляет переход к сервису построения пути, что значительно повышает практическую ценность бота, превращая его из справочника в инструмент для реального путешествия.

		
<p><i>Рис. 1. Главное меню бота после запуска</i></p>	<p><i>Рис. 2. Список мест в категории «Памятники»</i></p>	<p><i>Рис. 3. Карточка места</i></p>

Проведенное тестирование показало, что разработанный чат-бот успешно решает поставленные задачи. Пользовательский интерфейс является интуитивно понятным и не требует дополнительного обучения. Использование асинхронной модели обеспечивает высокую отзывчивость бота даже в условиях одновременных запросов от нескольких пользователей. Интеграция с внешним картографическим сервисом является удачным решением, расширяющим функциональность приложения.

Заключение

В ходе работы была достигнута поставленная цель — разработан и реализован полнофункциональный Telegram чат-бот для предоставления информации о туристических местах. Решение успешно интегрирует в себе базу данных SQLite для хранения структурированной информации и асинхронный Python-модуль на базе фреймворка aiogram для взаимодействия с пользователем.

Основные результаты работы:

1. Спроектирована и создана база данных, обеспечивающая надежное хранение и быстрый доступ к данным о туристических объектах.
2. Реализован интуитивно понятный пользовательский интерфейс с использованием Reply и Inline-клавиатур, что минимизирует усилия пользователя по поиску информации.
3. Обеспечена интеграция с внешним картографическим сервисом (Яндекс.Навигатор) для построения маршрутов, что расширяет функциональность бота за пределы мессенджера.
4. Применена асинхронная модель программирования, заложившая основу для масштабируемости и высокой производительности решения.

Перспективы дальнейшего развития проекта видятся в добавлении новых функций чат-бота, расширении списка категорий, а также внедрении машинного обучения для персонализированных рекомендаций на основе истории запросов пользователя.

Литература

1. Официальная документация Telegram Bot API. – URL: <https://core.telegram.org/bots/api>.
2. Официальная документация библиотеки aiogram (версия 3.x). – URL: <https://docs.aiogram.dev/en/latest/>.
3. Официальная документация по SQLite. – URL: <https://www.sqlite.org/docs.html>.
4. Документация по языку Python. – URL: <https://docs.python.org/3>.
5. Бизли Д. Python. Исчерпывающее руководство / Д. Бизли. – Санкт-Петербург, 2023. – 368 с.

ПРИМЕНЕНИЕ МЕТОДОВ СТАТИСТИЧЕСКОГО АНАЛИЗА И МАШИННОГО ОБУЧЕНИЯ К ИССЛЕДОВАНИЮ ПСИХОЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ПОЛЬЗОВАТЕЛЕЙ СОЦИАЛЬНЫХ СЕТЕЙ

Ф. Г. Типикина

Воронежский государственный университет

Аннотация. В статье рассматривается влияние различных социальных сетей на психоэмоциональное состояние пользователей с использованием методов статистического анализа и машинного обучения. На выборке из 500 респондентов изучены различия в уровне счастья, а также выявлены основные факторы, влияющие на уровень стресса. Результаты показали отсутствие статистически значимых различий в субъективном уровне счастья между пользователями разных платформ, однако ежедневное экранное время оказалось единственным значимым предиктором стресса. Модель случайного леса выявила, что возраст является важнейшим фактором выбора социальной сети, хотя точность прогнозирования платформы была ограничена. Выводы подчеркивают важность контроля времени, проведенного за экраном, для поддержания психологического благополучия, а не фокус на конкретной соцсети.

Ключевые слова: социальные сети, психоэмоциональное состояние, уровень стресса, уровень счастья, статистический анализ, машинное обучение, дисперсионный анализ, множественная регрессия, случайный лес, экранное время, цифровое потребление, психологическое благополучие.

Введение

В условиях стремительного развития цифровых технологий социальные сети стали неотъемлемой частью повседневной жизни миллионов людей по всему миру. Платформы, такие как Сеть 1, Сеть 2, Сеть 3, Сеть 4, Сеть 5 и Сеть 6, не только обеспечивают коммуникацию и обмен информацией, но и влияют на эмоциональное состояние, уровень стресса и субъективное ощущение счастья пользователей. В последние годы всё больше внимания уделяется изучению связи между использованием социальных сетей и психоэмоциональным благополучием, однако результаты исследований зачастую остаются противоречивыми [1].

Настоящее исследование направлено на выявление статистически значимых различий в уровне счастья между пользователями различных социальных платформ, а также на анализ факторов, наиболее сильно влияющих на уровень стресса. В работе применяются методы дисперсионного анализа (ANOVA), множественной линейной регрессии и машинного обучения (случайный лес) для всесторонней оценки взаимосвязей между поведенческими, демографическими характеристиками респондентов и их психологическим состоянием. Анализ проводится на основе данных из выборки из 500 респондентов, содержащей информацию об используемой социальной платформе, индексе счастья (по шкале от 1 до 10), уровне стресса, продолжительности экранного времени, качестве сна, возрасте и других переменных.

Целью исследования является проверка гипотезы о том, что выбор конкретной социальной сети оказывает значимое влияние на уровень счастья, а также определение главных предикторов стресса и возможностей прогнозирования используемой платформы на основе личностных и поведенческих характеристик пользователей.

Нулевая гипотеза (H_0): тип социальной сети не влияет на уровень счастья — средние значения 'Happiness_Index' одинаковы для всех платформ.

Альтернативная гипотеза (H_1): по крайней мере у одной платформы средний уровень счастья статистически отличается от других.

1. Анализ различий в уровне счастья между пользователями социальных платформ

Анализ проведён на выборке из пятисот респондентов из датасета “Mental_Health_and_Social_Media_Balance_Dataset.xlsx”. Для проверки гипотез использован однофакторный дисперсионный анализ (ANOVA), в котором зависимой переменной выступал индекс счастья (‘Happiness_Index (1-10)’), а группирующим фактором — предпочтаемая социальная сеть (‘Social_Media_Platform’).

Результаты ANOVA показали отсутствие статистически значимых различий в уровне счастья между группами пользователей ($F(5, 494) = 1.725, p = 0.127$). Таким образом, на основании имеющихся данных гипотеза H_0 не отвергается. Различия в средних значениях индекса счастья между пользователями Сеть 1, Сеть 2, Сеть 3, Сеть 4, Сеть 5 и Сеть 6 могут быть обусловлены случайной изменчивостью, а не систематическим влиянием выбора платформы.

Analysis of Variance (Mental_Health_and_Social_Media_Balance_Dataset)								
Variable	SS Effect	df Effect	MS Effect	SS Error	df Error	MS Error	F	p
Happiness_Index(1-10)	19,89628	5	3,979256	1139,416	494	2,306510	1,725229	0,127146

Рис. 1. Результаты однофакторного дисперсионного анализа (ANOVA) для сравнения уровня счастья между пользователями различных социальных сетей

Этот вывод указывает на то, что прямая связь между типом используемой социальной сети и субъективным уровнем счастья в данной выборке не подтверждается.

Визуальный анализ распределения уровня счастья по платформам с помощью boxplot показывает, что пользователи Сеть 6 имеют наибольшее среднее значение счастья ($M = 8.65$), за ними следуют Сеть 5 ($M = 8.52$) и Сеть 1 ($M = 8.35$). Пользователи Сеть 2 демонстрируют наименьший средний индекс счастья ($M = 7.99$). Однако, как показали результаты дисперсионного анализа, эти различия не достигают статистической значимости ($p = 0.127$), что указывает на необходимость учета других факторов при изучении связи между цифровыми платформами и психоэмоциональным состоянием.

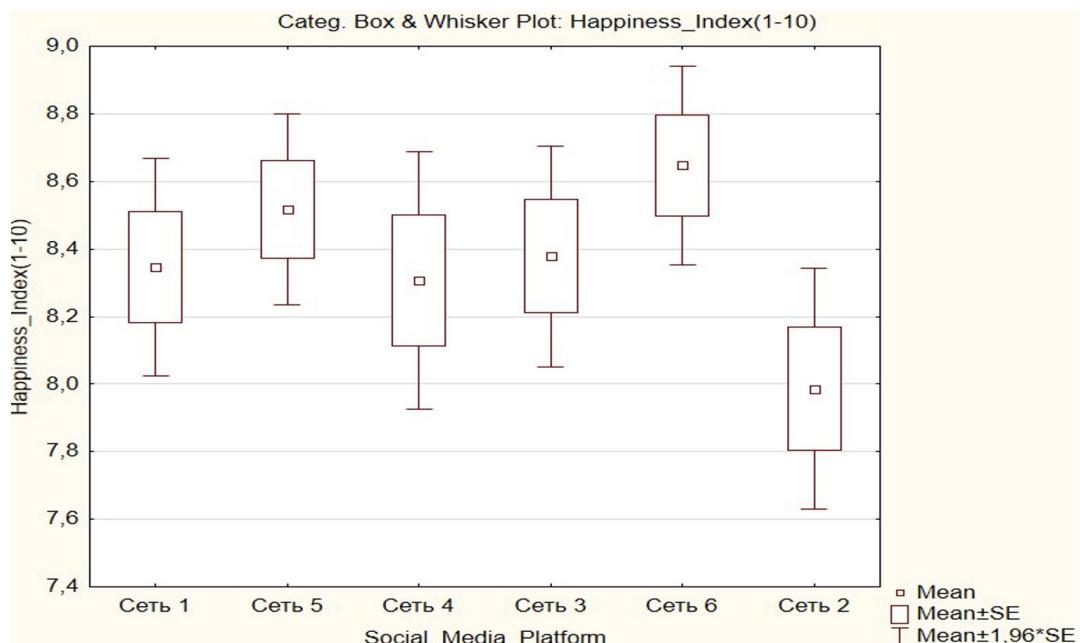


Рис. 2. Ящики с усами: распределение среднего уровня счастья пользователей различных социальных сетей

Таблица описательной статистики подтверждает визуальные наблюдения: средний индекс счастья варьируется от 7.99 (Сеть 2) до 8.65 (Сеть 6). При этом размер выборки для каждой платформы колеблется от 74 (Сеть 2) до 95 (Сеть 3), что позволяет предположить, что незначимость результатов ANOVA может быть обусловлена как недостаточной мощностью теста, так и высокой вариабельностью данных внутри групп.

Breakdown Table of Descriptive Statistics (Mental_Health_and_Social_Media_Balance_Dataset)

N=500 (No missing data in dep. var. list)

Social_Media_Platform	Happiness_Index(1-10) Means	Happiness_Index(1-10) N	Happiness_Index(1-10) Std.Dev.
Сеть 1	8,345679	81	1,476148
Сеть 5	8,517241	87	1,345640
Сеть 4	8,306667	75	1,684374
Сеть 3	8,378947	95	1,625455
Сеть 6	8,647727	88	1,406388
Сеть 2	7,986486	74	1,565845
All Grps	8,376000	500	1,524228

Рис. 3. Описательная статистика индекса счастья по типам используемых социальных сетей

2. Факторы, влияющие на уровень стресса

Для выявления факторов, наиболее сильно влияющих на уровень стресса пользователей социальных сетей [3], был применён метод множественной линейной регрессии. Множественная линейная регрессия [5] показала, что модель объясняет 55.2 % дисперсии уровня стресса ($R^2 = 0.552$, $p < 0.001$). Модель является статистически значимой ($F(5,494) = 121.83$, $p = 0.000000$), что указывает на наличие сильной связи между выбранными предикторами и уровнем стресса. Единственным значимым предиктором оказался ежедневное экранное время ($\beta = 0.709$, $p < 0.001$): каждый дополнительный час использования экрана увеличивает уровень стресса на 0.71 балла. Качество сна, возраст, частота физической активности и количество дней без соцсетей [2] не оказали статистически значимого влияния в данной модели.

Multiple Regression Results: Mental_Health_and_Social_Media_Balance_Dataset		?	X
Multiple Regression Results			
<p>Dependent: Stress_Level(1) Multiple R = ,74309705 F = 121,8309 R? = ,55219322 df = 5,494 No. of cases: 500 adjusted R? = ,54766077 p = 0,000000 Standard error of estimate: 1,037760227 Intercept: 3,270947583 Std.Error: ,5424318 t(494) = 6,0302 p = ,0000</p>			
Age b*=-,01 Daily_Screen_ b*=-,709 Sleep_Quality b*=-,05 Days_Without_ b*=-,026 Exercise_Freq b*=-,053			

Рис. 4. Результаты множественной линейной регрессии для прогнозирования уровня стресса на основе поведенческих и демографических характеристик

Анализ остатков с помощью Normal Probability Plot показал, что остатки распределены нормально, что подтверждает адекватность модели и надёжность полученных результатов.

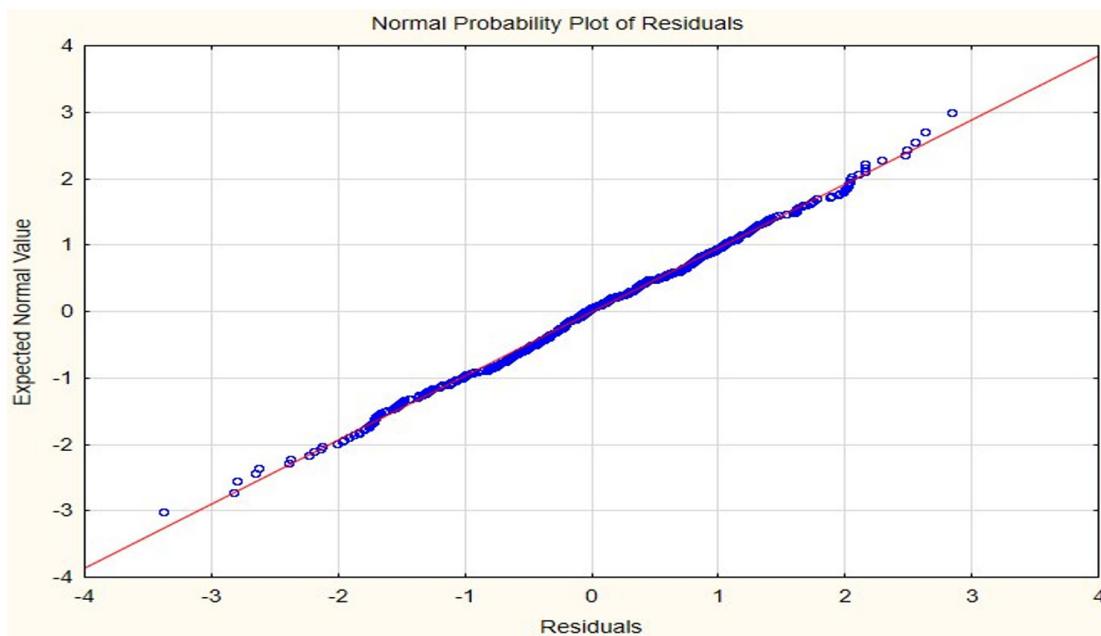


Рис. 5. Нормальный вероятностный график остатков модели множественной линейной регрессии

3. Прогнозирование социальной платформы с помощью случайного леса

Также, в рамках исследования была построена модель классификации на основе метода случайного леса [4] с целью предсказания предпочтаемой социальной сети по поведенческим и демографическим характеристикам респондентов.

Анализ важности признаков показал, что возраст является наиболее значимым предиктором выбора платформы (относительная важность = 1.00), за ним следуют качество сна и ежедневное экранное время. Пол проявил наименьшее влияние, что указывает на доминирующую роль поведенческих факторов над демографическими при принятии решения о выборе платформы.

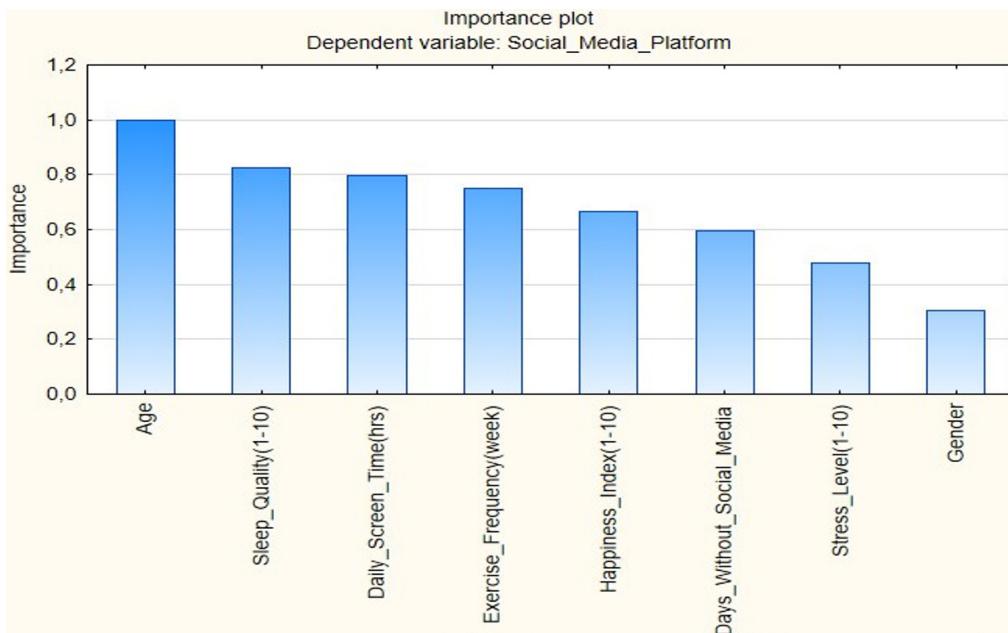


Рис. 6. Важность признаков для прогнозирования используемой социальной сети по методу случайного леса

Матрица классификации демонстрирует распределение ошибок модели на обучающей выборке. Наиболее точно предсказывается платформа Сеть 6 — из 9 наблюдений, все 9 были классифицированы верно. Платформы Сеть 5 и Сеть 4 также показали высокую точность (1 и 2 верных предсказания соответственно). Матрица построена на обучающей выборке, состоящей из 43 наблюдений. В этой подвыборке было 6 наблюдений Сеть 1, из которых модель правильно предсказала 5; из 6-ти наблюдений Сеть 2 — 1 верное предсказание. Из 6 наблюдений Сеть 1 модель правильно предсказала 5, а из 6 наблюдений Сеть 2 — лишь 1. Это может указывать на то, что профили пользователей этих платформ пересекаются с другими или их характеристики недостаточно дискриминативны для данной модели. Общая точность модели на этой выборке составляет $23 / 43 \approx 53.5\%$, что подтверждает сложность задачи классификации социальных сетей на основе поведенческих признаков.

Classification matrix (Mental_Health_and_Social_Media_Balance_Dataset)						
Response: Social_Media_Platform						
Training set sample: Number of trees: 180						
	Class Predicted Сеть 1	Class Predicted Сеть 5	Class Predicted Сеть 4	Class Predicted Сеть 3	Class Predicted Сеть 6	Class Predicted Сеть 2
Observed Сеть 1	5,000000				1,000000	
Observed Сеть 5	1,000000				5,000000	
Observed Сеть 4			2,000000		3,000000	
Observed Сеть 3					5,000000	
Observed Сеть 6					9,000000	
Observed Сеть 2	1,000000				5,000000	

Рис. 7. Матрица классификации для прогнозирования используемой социальной сети с помощью модели случайного леса

Распределение априорных вероятностей [6] показало, что в обучающей выборке наиболее представлены платформы Сеть 5, Сеть 4 и Сеть 6 (по 0.22–0.33), тогда как Сеть 1 и Сеть 2 встречаются реже (0.11 и 0.00 соответственно). Отсутствие наблюдений для Сеть 2 в обучающей выборке может объяснить низкую точность его предсказания, отмеченную в матрице классификации.

Priors (Mental_Health_and_Social_Media_Balance_Dataset)		
Response: Social_Media_Platform		
	Prior probabilities	Class n.
Сеть 1	0,111111	6
Сеть 5	0,222222	6
Сеть 4	0,222222	5
Сеть 3	0,111111	5
Сеть 6	0,333333	9
Сеть 2	0,000000	6

Рис. 8. Априорные вероятности и размеры групп для различных социальных сетей в обучающей выборке модели случайного леса

Заключение

Исследование не выявило статистически значимых различий в уровне счастья между пользователями разных социальных сетей ($F(5,494) = 1.725, p = 0.127$). Средние значения индекса счастья варьировались от 7.99 (Сеть 2) до 8.65 (Сеть 6), однако эти различия могут быть обусловлены случайной изменчивостью, а не влиянием платформы.

В то же время множественная регрессия показала, что ежедневное экранное время — единственный значимый предиктор стресса ($\beta = 0.709, p < 0.001$), объясняя 55.2 % его дисперсии. Остальные переменные (возраст, качество сна, физическая активность и др.) статистически не значимы.

Модель случайного леса для прогнозирования используемой платформы выявила возраст как наиболее важный признак, однако общая точность классификации составила лишь ~53.5 %, что указывает на сложность и многогранность выбора соцсетей.

Таким образом, важнее не то, какую соцсеть использует человек, а сколько времени он проводит за экраном. Для поддержания психологического благополучия целесообразно регулировать интенсивность цифрового потребления, а не фокусироваться на конкретной платформе.

Литература

1. Багдасарьян Н. Г. Цифровое поведение личности в интернет-коммуникациях: культура и риски / Н. Г. Багдасарьян, А. П. Ромашкина // Вестник государственного университета Дубна. Серия: Науки о человеке и обществе. – 2021 – № 1 – С. 61–72.
2. Белинская Е. П., Гавриченко О. В. Самопрезентация в виртуальном пространстве: феноменология и закономерности // Психологические исследования. – 2018. – Т. 11, № 60. – С. 12.
3. Войскунский А. Е. Социальная перцепция в социальных сетях // Вестник Московского университета. Серия 14 Психология. – 2014. – № 2 [электронный ресурс] URL:<https://cyberleninka.ru/article/n/sotsialnaya-pertsepsiya-v-sotsialnyh-setyah> (дата обращения: 23.11.2025).
4. Семериков А. В., Глазырин М. А. Классификация объектов на основе нейронной сети и методами дерева решения и ближайших соседей: учеб. пособие. – Ухта : УГТУ, 2022. – 68 с. URL: <https://e.lanbook.com/book/267857>
5. Флах П. Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных. – М. : ДМК Пресс, 2015. – 400 с. URL: <https://e.lanbook.com/book/69955>
6. Шалев-Шеарц Ш., Бен-Давид Ш. Идеи машинного обучения: учеб. пособие / пер. с англ. А. А. Слинкина. – М. : ДМК. : Пресс, 2019. – 436 с. URL: <https://e.lanbook.com/book/131686> (дата обращения: 17.11.2025)

ПРОГНОЗИРОВАНИЕ РИСКА СМЕРТНОСТИ ПОСЛЕ ИНФАРКТА МИОКАРДА С ИСПОЛЬЗОВАНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Я. Ю. Турусова¹, И. Л. Каширина²

¹Воронежский государственный университет

²МИРЭА – Российский технологический университет

Аннотация. В статье рассматривается задача прогнозирования риска смертности после ИМ с использованием методов машинного обучения. Для исследования сформирована выборка пациентов с клиническими, демографическими и метеорологическими характеристиками, прошедшая этапы очистки и отбора признаков. Для построения прогностических моделей применены логистическая регрессия, деревья решений, ансамблевые методы (Random Forest, градиентный бустинг), а также нейронные. Проведён сравнительный анализ алгоритмов, по результатам которого наиболее высокую эффективность продемонстрировал метод Random Forest, обеспечивший лучшие показатели точности, F1-метрики и AUC. Использование SMOTE позволило повысить чувствительность моделей за счёт балансировки классов. На основе полученных результатов разработан прототип системы поддержки принятия решений для оценки риска летального исхода после ИМ.

Ключевые слова: инфаркт миокарда, смертность, прогнозирование риска, машинное обучение, логистическая регрессия, деревья решений, Random Forest, градиентный бустинг, нейронные сети, клинические данные, метеорологические параметры, SMOTE, дисбаланс классов, AUC, F1-метрика.

Введение

Сердечно-сосудистые заболевания продолжают оставаться одной из ведущих причин смертности в мире, а инфаркт миокарда занимает среди них ключевую позицию. Эти патологии затрагивают не только здоровье отдельных людей, но и оказывают значительное социально-экономическое воздействие, увеличивая нагрузку на системы здравоохранения. Своевременное выявление факторов риска и раннее вмешательство способны существенно улучшить прогноз пациентов и снизить затраты на лечение. В современных условиях развивается все больше технологий, направленных на улучшение диагностики и прогнозирования, что открывает новые возможности для персонализированной медицины. Своевременное прогнозирование риска неблагоприятных исходов после перенесённого инфаркта является важнейшим условием повышения эффективности лечения, оптимизации маршрутизации пациентов и снижения уровня летальности. В последние годы значительное развитие получили методы машинного обучения, позволяющие выявлять скрытые закономерности в больших массивах медицинских данных и улучшать качество прогностических моделей. Однако большинство существующих отечественных клинических систем по-прежнему ограничены традиционными статистическими методами и редко учитывают комплекс факторов, включающих клинико-демографические параметры, особенности состояния пациента и влияние внешней среды. В этой связи возрастаёт необходимость создания современных подходов к анализу рисков после инфаркта миокарда, использующих интеграцию различных типов данных, включая метеорологические и сезонные показатели. Основной целью данного исследования является разработка и оценка эффективности моделей машинного обучения для прогнозирования риска смертности после инфаркта миокарда. Задачи исследования включают формирование и очистку базы данных пациентов, определение ключевых факторов, влияющих на исход заболевания, построение и сравнение нескольких алгоритмов прогнозирования, а также оценку их

точности с применением стандартных метрик качества. Полученные результаты могут стать основой для создания системы поддержки принятия врачебных решений, повышающей точность и своевременность клинических прогнозов. [1–4]

1. Исходные данные

Набор деперсонифицированных данных, предоставленных Воронежским областным клиническим консультативно-диагностическим центром (ВОКДЦ), включает сведения обо всех пациентах, перенесших инфаркт миокарда в Воронежской области в период с 2014 по 2017 год. Датасет содержит следующие показатели (рис. 1): ID — уникальный идентификатор пациента, AGE — возраст, GENDER — пол, INVALID — наличие инвалидности, AG — артериальная гипертензия, IM — инфаркт миокарда, SD — сахарный диабет, FP — факт курения, ONMK — острое нарушение мозгового кровообращения в анамнезе, HOBL — хроническая обструктивная болезнь лёгких, HSN — хроническая сердечная недостаточность, LOCAL — локализация инфаркта, HOSPIT_TIME — время до госпитализации, KILLIP — класс Killip, TLT — тромболитическая терапия, CHKV — чрескожное коронарное вмешательство, MAX_T — максимальная температура воздуха, DELTA_T — суточный перепад температуры, WIND — скорость ветра, PRESSURE — атмосферное давление, HUMIDITY — влажность воздуха, CLOUDINESS — облачность, IS_DEAD — летальный исход.

На конец 2017 года база сведений содержала более 14 тысяч записей, включающих сведения о пациентах — выборка пациентов с клиническими, демографическими и метеорологическими характеристиками, отражающими как состояние здоровья и факторы риска, так и влияние внешних условий, потенциально связанных с исходами инфаркта миокарда.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
1	ID	AGE	GENDER	INVALID	AG	IM	SD	FP	ONMK	HOBL	HSN	LOCAL	HOSPIT	KILLIP	TLT	CHKV	MAX_T	DELTA_T	WIND	PRESSURE	HUMIDITY	CLOUDINESS	IS_DEAD
2	3_2014	65	2	2	1	0	0	0	0	0	2	5	6	1	0	0	11,4	2,1	5,25	766,788	40,5	53,75	0
3	7_2014	53	1	0	1	0	0	0	0	0	2	5	6	1	0	1	19,8	0,4	2,75	757,5	68,4	62,5	0
4	17_2014	85	2	2	1	0	0	0	0	0	2	5	4	2	0	0	29,9	2,9	3,75	766,013	30,8	20	0
5	21_2014	63	1	0	0	0	0	0	0	0	0	5	6	0	0	0	25,6	0,8	4	761,263	42,9	76,25	1
6	40_2014	80	1	2	1	0	0	0	0	0	1	5	6	1	0	0	19,8	3,8	3,25	764,138	68,1	37,5	0
7	45_2014	62	1	0	0	0	0	0	0	0	0	3	2	1	0	0	5,3	1,9	1,63	773,725	77,3	30	0
8	47_2014	41	1	0	0	0	0	0	0	0	0	5	2	1	4	0	-2,4	5	2,75	776,038	66,1	25	0
9	70_2014	37	1	0	1	0	0	0	0	1	3	2	1	0	0	0	2,6	0,6	0,88	763,725	91,3	65	0
10	85_2014	63	1	2	0	0	0	1	0	0	2	5	4	1	0	0	4,2	10,4	4	766,375	62,9	58,75	0

Рис. 1. Фрагмент исходных данных

2. Используемые модели машинного обучения

В исследовании были применены несколько классов моделей машинного обучения, различающихся по сложности, интерпретируемости и способности моделировать нелинейные зависимости. В качестве базовой интерпретируемой модели использовалась логистическая регрессия, позволяющая оценивать вклад отдельных клинических и демографических факторов в риск неблагоприятного исхода. Для моделирования более сложных закономерностей применялись деревья решений, основанные на рекурсивном разбиении признакового пространства и удобные с точки зрения интерпретации. Для повышения устойчивости и точности прогнозов использовались ансамблевые методы. В частности, был реализован случайный лес (Random Forest), объединяющий множество деревьев решений, построенных на различных подвыборках данных и подмножествах признаков. Также применялись алгоритмы градиентного бустинга (включая XGBoost и LightGBM), обучающие деревья последовательно для минимизации ошибки предыдущих моделей, что обеспечивает высокую точность на структурированном медицинском датасете. Дополнительно исследовались нейронные сети, способные моделировать сложные нелинейные зависимости между клиническими, демографическими и метеорологическими параметрами. Использовались полно связанные архитектуры с регуляризацией и подбором гиперпараметров. Для повышения надёжности выводов все модели про-

ходили процедуру кросс-валидации, а оптимизация выполнялась с использованием методов автоматизированного поиска гиперпараметров.

Также в исследовании применялся метод SMOTE (Synthetic Minority Over-sampling Technique), который используется для устранения дисбаланса классов в задачах машинного обучения и позволяет увеличить представленность редкого класса без простого дублирования данных. Он создаёт синтетические объекты редкого класса на основе ближайших соседей существующих примеров, расширяя область их распределения в пространстве признаков. Это помогает улучшить качество классификации, повышает разделяемость классов и снижает риск переобучения на повторяющихся данных.

3. Сравнительный анализ моделей машинного обучения

Перед применением SMOTE данные были сильно несбалансированы: число пациентов, переживших инфаркт, значительно превышало число смертельных исходов. Это приводило к тому, что модели машинного обучения демонстрировали высокую точность по общему числу прогнозов (Accuracy), но относительно низкую чувствительность и F1-score для менее представленного класса. Другими словами, модели часто «пропускали» критические случаи, что снижало практическую ценность прогнозов. В табл. 1 представлены результаты работы пяти алгоритмов на исходных данных без применения SMOTE, оцененные по трём ключевым метрикам: Accuracy, AUC и F1-score.

Таблица 1

Эффективность алгоритмов на исходных данных

Метод	Accuracy	AUC	F1 score
Логистическая регрессия	0.710	0.680	0.420
Алгоритмы градиентного бустинга	0.830	0.780	0.590
Деревья решений	0.790	0.730	0.510
Нейронные сети	0.860	0.880	0.680
Случайный лес	0.890	0.910	0.750

До применения метода SMOTE модели показывают высокую точность, но хуже справляются с выявлением редкого класса (смертельные исходы), что видно по низким значениям F1-score. Это связано с дисбалансом классов в исходных данных. На рис. 2 приведено сравнение по Accuracy, AUC и F1-score до применения метода SMOTE.

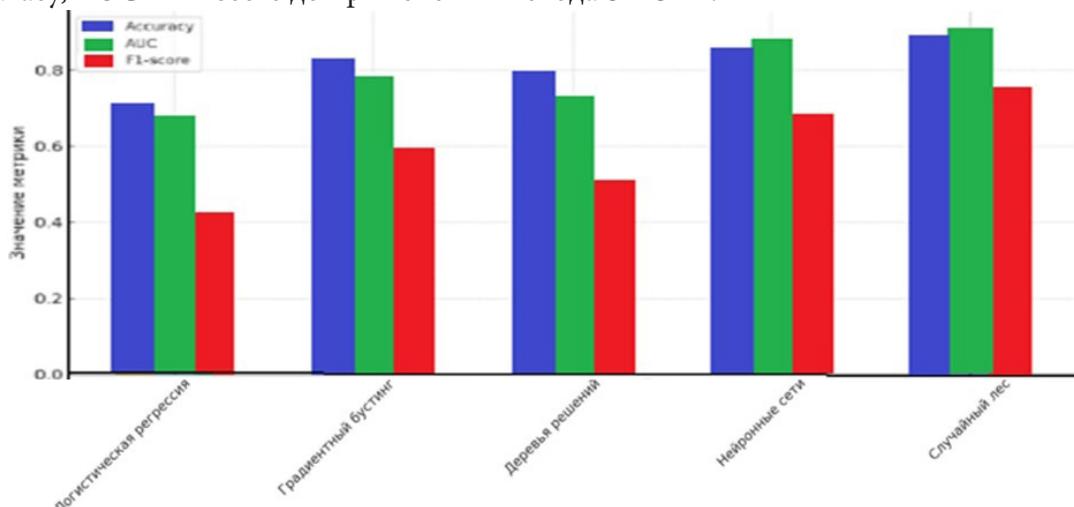


Рис. 2. Сравнение по Accuracy, AUC и F1-score (до метода SMOTE)

Далее для повышения качества моделей и борьбы с дисбалансом классов применялся метод SMOTE (Synthetic Minority Over-sampling Technique), который позволил увеличить количество наблюдений для менее представленного класса и, как следствие, повысить чувствительность моделей. Оценка проводилась по трём ключевым метрикам: Accuracy, AUC и F1-score. Результаты представлены в табл. 2.

Таблица 2

Эффективность моделей на данных, сбалансированных методом SMOTE

Метод	Accuracy	AUC	F1 score
Логистическая регрессия	0.680089	0.752221	0.650367
Алгоритмы градиентного бустинга	0.823266	0.852928	0.792651
Деревья решений	0.771812	0.784199	0.743073
Нейронные сети	0.861298	0.923966	0.831978
Случайный лес	0.901566	0.957367	0.883905

На основании данных таблицы можно сделать вывод, что наилучшую эффективность продемонстрировал алгоритм случайного леса, показавший максимальные значения всех ключевых метрик: Accuracy = 0.901566, AUC = 0.957367 и F1-score = 0.883905. Высокая площадь под ROC-кривой подтверждает устойчивость и надёжность модели. Для наглядного сравнения результаты представлены на рис. 3.

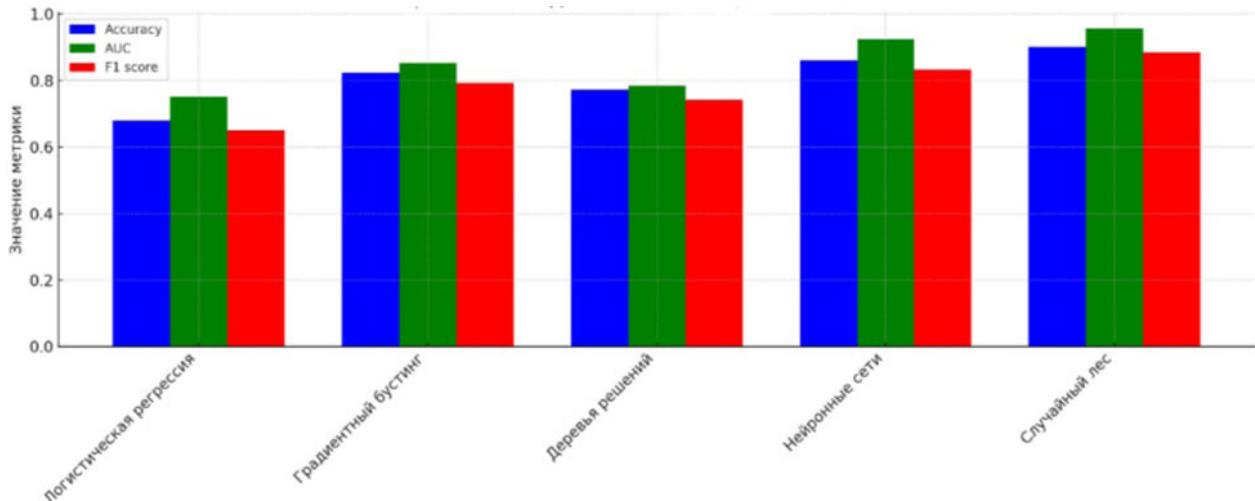


Рис. 3. Сравнение по Accuracy, AUC и F1-score (после метода SMOTE)

Заключение

На основе проведённого анализа данных пациентов, перенесших инфаркт миокарда, построена начальная модель прогнозирования риска летального исхода с использованием методов машинного обучения, включая логистическую регрессию, деревья решений, ансамблевые методы и нейронные сети. Для повышения качества моделей и корректной работы с несбалансированными данными применён метод SMOTE, что позволило улучшить чувствительность моделей и точность прогноза. Наибольшую эффективность продемонстрировал алгоритм случайного леса. Полученные результаты послужили основой для разработки прототипа системы поддержки принятия решений, способной помогать в оценке риска летального исхода и повышать своевременность клинических прогнозов.

Литература

1. *Li W., Yan D., Hu W., Su X., Zhang Z.* Enhancing one-year mortality prediction in STEMI patients post-PCI: an interpretable machine learning model with risk stratification // BMC Cardiovascular Disorders. – 2023. – Vol. 23, No 1. – P. 1–13.
2. *Zhao J., Li Q., Chen Y., Wang H.* Optimized Machine Learning Models to Predict In-Hospital Mortality in STEMI // Therapeutics and Clinical Risk Management. – 2021. – Vol. 17. – P. 101–115.
3. *James G.* An Introduction to Statistical Learning: with Applications in R / G. James, D. Witten, T. Hastie, R. Tibshirani. – 2nd ed. – New York : Springer, 2021. – 607 p. – (Springer Texts in Statistics).
4. *Ломакина Л. С.* Идентификация состояний объектов сложной структуры на основе нейронечётких технологий / Л. С. Ломакина, С. А. Манцеров // Системы управления и информационные технологии. – 2022. – № 1(87). – С. 8–12.

АГЕНТСКИЙ ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ В КИБЕРБЕЗОПАСНОСТИ: КОНЦЕПЦИЯ И СЦЕНАРИИ ПРИМЕНЕНИЯ

М. А. Утенкова, Е. А. Максимова

МИРЭА – Российский технологический университет

Аннотация. Рассматривается применение агентского искусственного интеллекта как перспективного направления для решения задач кибербезопасности. Проведен анализ концепции агентского ИИ, его архитектуры и ключевых отличий от систем на основе сигнатурного и поведенческого анализа. Определены основные сценарии применения ИИ-агентов в кибербезопасности, такие как автономное расследование инцидентов, проактивный поиск угроз, управление уязвимостями и автоматизация отчетности. Выявлены потенциальные риски, связанные с внедрением агентского ИИ, включая компрометацию памяти и несанкционированное использование инструментов.

Ключевые слова: агентский искусственный интеллект, кибербезопасность, информационная безопасность, искусственный интеллект, автономные системы, SOC, расследование инцидентов, управление уязвимостями, OWASP, большая языковая модель.

Введение

Статистика количества утечек информации в мире показывает тенденцию к их увеличению с течением времени. Так, например, количество утечек информации за предыдущие 3 года выросло в 4–5 раз в сравнении с 2021 годом [2]. При этом, эксперты компании Positive Technologies выделяют геополитическую обстановку и темпы цифровизации как ключевые факторы, оказывающие наибольшее влияние на ландшафт киберугроз в России [1].

Количество утечек

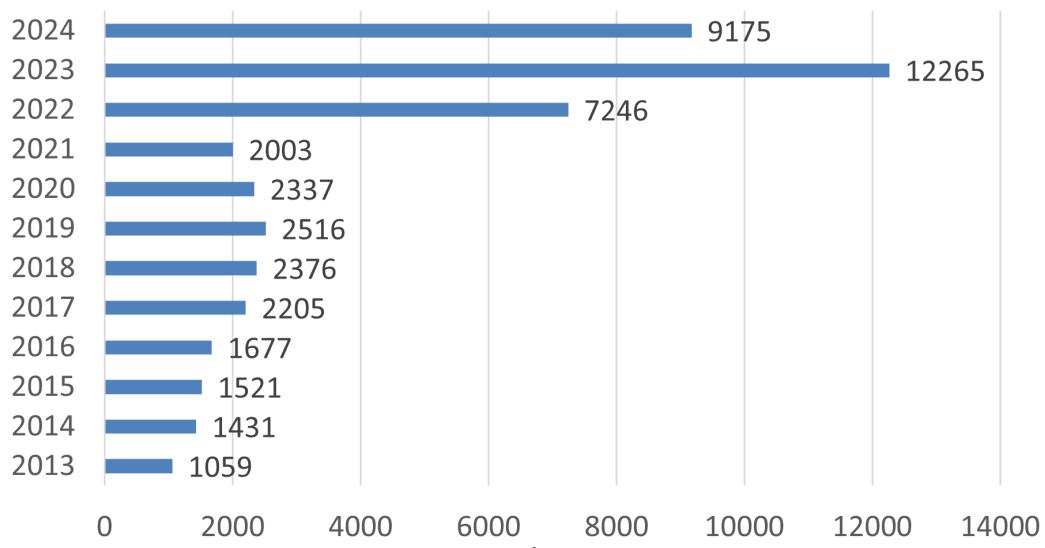


Рис. 1. Количество утечек информации в мире 2013–2024 гг.

Стоит также отметить, что комплексность и сложность кибератак все чаще приводят к комбинированным последствиям, которые включают как утечки данных, так и нарушения в работе бизнес-процессов. В то же время такие атаки сложнее выявить на основе сигнатурных и поведенческих методов. По этой причине появилась потребность в более интеллектуальных, адаптивных и автономных системах защиты информации.

В рамках данного исследования агентский искусственный интеллект (ИИ-агент) рассматривается как возможное решение данной проблемы. ИИ-агент — это автономная система на базе искусственного интеллекта, которая способна самостоятельно анализировать запросы, принимать решения и выполнять действия.

1. Концепция агентского искусственного интеллекта

Предлагается следующее определение: ИИ-агент — это автономная система на базе искусственного интеллекта, которая способна самостоятельно анализировать запросы, принимать решения и выполнять действия. На сегодняшний день ИИ-агенты активно применяются в различных отраслях. Важным отличием агентского ИИ от других видов является то, что он способен не просто проанализировать поставленную задачу, а принять решение и выполнить действия в соответствии с ним. Если, например, LLM (Large language model) способна предложить решение пользователю, то ИИ-агент благодаря API способен напрямую взаимодействовать с другими сервисами и выполнять действия.

Компания OWASP предлагает следующую архитектуру агента искусственного интеллекта (рис. 2) [3] Ключевыми компонентами архитектуры являются: LLM, агент, инструменты, база данных, выполняющая функцию долгосрочной памяти агента.

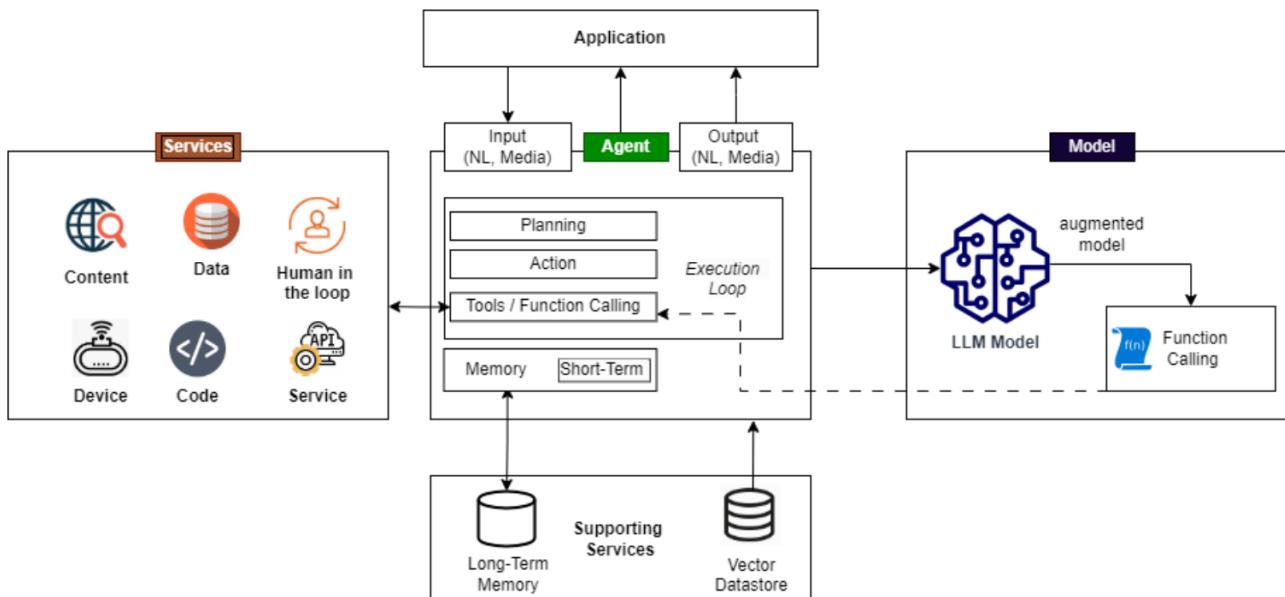


Рис. 2. Архитектура агента искусственного интеллекта

Алгоритм работы агента следующий:

1. Поступает пользовательский запрос, который может представлять собой как инструкцию человека, так и, например, оповещение от SIEM.
2. Агент готовит контекст для LLM, в процессе чего может запросить дополнительную информацию из базы данных и других сервисов.
3. LLM формирует план действий.
4. Агент выполняет план с помощью инструментов (программы или устройства) через API.

Важно отметить, что на практике могут встречаться также многоагентные системы, которые включают два и более агентов, которые обмениваются друг с другом информацией и запросами. При этом, каждый агент может выполнять свой уникальный набор задач. ИИ-агент может быть одним из модулей интеллектуальных СППР [5].

Агентский искусственный интеллект порождает новые киберугрозы [3], перечень которых включает: отравление памяти, неправильное использование инструментов, компрометацию

привилегий, каскадные атаки с галлюцинациями, несогласованное и обманчивое поведение, отказ авторства и невозможность отслеживания и др. В текущих реализациях ИИ-агенты могут некритично полагаться на данные из своей памяти, что делает их уязвимыми к компрометации. При этом, его работу полностью протестировать невозможно, так как логика принятия решения остается неизвестной и ее нельзя отследить. Поэтому на текущем этапе развития технологий рекомендуется применять средства мониторинга по отношению к ИИ-агентам, которые могут отследить нежелательные действия и возможно даже предотвратить их.

2. Сценарии применения агентского искусственного интеллекта в кибербезопасности

В контексте кибербезопасности ИИ-агент может быть использован как один из модулей SOC, что реализовано, например, в организации Radiant Security [4]. Сценарии применения агентского ИИ включают выполнение таких задач, как:

– Автоматическая сортировка и приоритизация оповещений: агент получает оповещение от SIEM, запрашивает дополнительную информацию из логов и данных EDR и определяет приоритетность оповещений и осуществляет их сортировку.

– Автономное расследование инцидентов информационной безопасности и их корреляция: агент получает оповещение от SIEM, запрашивает дополнительную информацию из логов и данных EDR, коррелирует их, выдвигает гипотезы о векторе атаки и предоставляет отчет, включающий рекомендации по устранению.

– Проактивный поиск угроз: агент генерирует гипотезы о потенциальных векторах атак на основе информации об актуальных угрозах, после чего проводит проверки инфраструктуры на устойчивость к данным атакам и составляет отчет об обнаруженных угрозах.

– Управление уязвимостями: в ходе сканирования системы на уязвимости или проактивного поиска угроз агент выявляет новые уязвимости, определяет их критичность [6], при наличии патча, тестирует его на устойчивость к обнаруженным уязвимостям, и устанавливает его в рабочую систему при минимальном вмешательстве со стороны человека.

– Автоматизированное ведение документации и отчетности: получение информации из SIEM и других средств защиты информации и представление ее в формате отчета в соответствии с заданным шаблоном.

ИИ-агенты необходимы для сокращения повторяющихся задач, ускорения процесса детектирования и реагирования, устранения трудоемких задач, для повышения общего уровня безопасности.

Заключение

Таким образом, агентский ИИ является мощным инструментом, который меняет подходы к кибербезопасности. Тем не менее на текущий момент он также является и источником возможных киберугроз. Поэтому к внедрению данной технологии стоит подходить с особым вниманием, чтобы минимизировать потенциальные риски и повысить ее эффективность.

Перспективы дальнейших исследований включают разработку безопасной архитектуры ИИ-агента и методов обеспечения его надежности, создание стандартов в отношении агентского ИИ, исследование методов объяснимого ИИ для агентских систем и др.

Литература

1. CODE RED 2026: Актуальные киберугрозы для российских организаций / Positive Technologies. – 2025. – URL: <https://ptsecurity.com/research/analytics/russia-cyberthreat-landscape-2026/> (дата обращения: 12.11.2025).

2. Утечки информации в мире 2023-2024 годы : [аналитический отчет] / InfoWatch. – 2025. – URL: <https://www.infowatch.ru/sites/default/files/analytics/files/utechki-informatsii-v-mire-2023-2024-gody.pdf> (дата обращения: 12.11.2025).
3. Agentic AI – Threats and Mitigations : [руководство] / OWASP. – URL: <https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/> (дата обращения: 15.11.2025).
4. AI Agents in the SOC: Transforming Cybersecurity Operations / Radiant Security. – URL: <https://radiantsecurity.ai/learn/ai-agents/> (дата обращения: 15.11.2024).
5. Утенкова М. А. Архитектура СППР для оценки рисков информационной безопасности / М. А. Утенкова, А. А. Комков, Е. А. Максимова // Актуальные проблемы прикладной математики, информатики и механики : сборник трудов Международной научной конференции, Воронеж, 02–04 декабря 2024 года. – Воронеж : Научно-исследовательские публикации, 2025. – С. 1402–1406. – EDN FLHVXC.
6. *Utenkova M. Development of an Information Security Risk Assessment Model Through the Calculation of the Vulnerability Danger Factor / M. Utenkova, A. Komkov, E. Maksimova, O. Mezentseva // AISMA-2025: International Workshop on Advanced Information Security Management and Applications : Conference proceedings, Stavropol, 11–15 мая 2025 года. Vol. 1456. – Springer Cham: Springer Nature Switzerland AG, 2025. – P. 434–441. – EDN RSVFLL.*

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ ОПРЕДЕЛЕНИЯ ВЕСОВЫХ КОЭФФИЦИЕНТОВ OWA-ОПЕРАТОРОВ В ЗАДАЧЕ МНОГОКРИТЕРИАЛЬНОГО ВЫБОРА

Д. А. Фефелов, Т. М. Леденева

Воронежский государственный университет

Аннотация. В условиях многокритериального выбора, когда каждый объект характеризуется набором разнородных показателей, возникает задача формирования обобщенной оценки объектов, которая позволяет сравнить и ранжировать заданные объекты по предпочтительности. Для решения такой задачи существуют операции агрегирования, среди которых особое место занимают операторы порядкового взвешенного агрегирования (Ordered Weighted Averaging—OWA). Данная статья посвящена исследованию применения OWA-операторов для построения обобщенных оценок сложных объектов. В ней рассмотрены основные подходы к определению весовых коэффициентов, которые определяют стратегию агрегирования. На практическом примере выбора инвестиционного проекта с помощью различных методов проведен расчет весов и построено итоговое ранжирование проектов на основе полученных обобщенных оценок. Проведенный анализ позволит выявлять наиболее предпочтительные варианты и оценить влияние выбранной стратегии агрегирования на конечный результат выбора.

Ключевые слова: агрегирование, OWA-оператор, обобщенная оценка, весовые коэффициенты, стратегия агрегирования.

Введение

В процессе принятия решений часто приходится сталкиваться с необходимостью оценки и выбора наилучшего объекта из множества альтернативных вариантов, каждый из которых описывается набором частных показателей. Эти показатели могут быть разнородными, иметь различные единицы измерения, что делает их прямое сравнение затруднительным и неэффективным. В этих условиях центральное место занимает задача агрегирования, которая заключается в построении единой, обобщенной (комплексной, интегральной) оценки объекта на основе всей доступной векторной информации.

Под агрегированием понимается переход от векторной оценки размерности n к скалярной величине, которая комплексно характеризует объект [2]. Проблема описания класса средних величин, являющихся ядром процедуры агрегирования, имеет глубокие корни в математике, которые в своей работе описывал А. Н. Колмогоров о понятии среднего [3]. Существуют различные классы операций и функций агрегирования, относящиеся к семейству средних [5], каждый из которых реализует определенную стратегию: конъюнктивную (ориентированную на наихудшие оценки), дизъюнктивную (ориентированную на наилучшие оценки) и компромиссную [2]. Выбор конкретной функции агрегирования является ключевым шагом при построении оценочной системы [4].

Целью данной работы является демонстрация применения одной из наиболее гибких и мощных инструментов агрегирования — оператора порядкового взвешенного агрегирования (OWA), который применяется для решения практической задачи многокритериального (или многоатрибутного) выбора.

1. OWA-оператор и подходы к определению весовых коэффициентов

Особый интерес в семействе базовых классов агрегирования представляет Order Weighted Averaging (OWA) оператор, введенный Р. Ягером [7]. Его особенность заключается в том, что

перед агрегированием компоненты векторной оценки упорядочиваются по невозрастанию, а весовые коэффициенты связываются не с конкретными показателями, а с их порядковыми позициями [1]. Это позволяет напрямую управлять стратегией агрегирования, задавая ту или иную степень оптимизма/пессимизма ЛПР (лица, принимающего решение), и представляет собой развитие идеи о специализированных операторах усиления [6].

n -местный OWA-оператор, ассоциированный с вектором $W = (w_1, \dots, w_n)$, где $w_i \in [0, 1]$ и $\sum w_i = 1$, определяется следующим образом [2]: $\Phi(W, X) = \sum_{i=1}^n w_i y_i$, где y_i — i -й по величине элемент вектора X (т. е. $y_1 \geq y_2 \geq \dots \geq y_n$).

Ключевым этапом в использовании OWA-оператора является определение вектора весов W , который полностью определяет его свойства: стратегию (через показатели *orness* и *andness*), уровень компенсации и равномерного учета аргументов (через энтропию) [1]. Данный этап так же можно рассмотреть, как проблему параметризации функции агрегирования, которая рассмотрена в [4, 5].

Общий подход к построению целенаправленной процедуры агрегирования, в рамках которой формируются обобщенные оценки, включает этап выбора стратегии агрегирования, обоснование необходимости учета компенсационных свойств функции или операции агрегирования, а также учета того, равномерно ли будут учтены частные оценки объектов. Преимуществом OWA-операторов является то, что для оценки перечисленных свойств процедуры агрегирования они имеют количественные показатели, которые определяются вектором весовых коэффициентов.

Рассмотрим несколько подходов к определению весовых коэффициентов, которые могут быть использованы для целенаправленного построения процедуры агрегирования.

1.1. Метод на основе функции квантификации

Данный подход определять весовые коэффициенты на основе функций квантификации, которые формализуют понятие лингвистического квантора («большинство», «по крайней мере половина» и т. п.) [1]. Пусть $Q: [0, 1] \rightarrow [0, 1]$ — непрерывная неубывающая функция квантификации, тогда весовые коэффициенты определяются по формуле [1]:

$$w_i = Q\left(\frac{i}{n}\right) - Q\left(\frac{i-1}{n}\right), \quad \text{для } i = 1, \dots, n.$$

Например, для степенной функции $Q(x) = x^\alpha$, $\alpha > 0$, веса рассчитываются как $w_i = \left(\frac{i}{n}\right)^\alpha - \left(\frac{i-1}{n}\right)^\alpha$. При $\alpha = 1$ получается среднее арифметическое (*orness* = 0.5), при $\alpha < 1$ — оператор будет склонен к конъюнктивной стратегии, при $\alpha > 1$ — к дизъюнктивной [1].

1.2. Метод на основе производной функции квантификации

Этот подход делает веса зависимыми не только от порядка, но и от значений самих аргументов, что позволяет получить более адаптивную процедуру агрегирования. Веса рассчитываются по формуле:

$$w_i = \frac{Q'(1 - y_i)}{\sum_{j=1}^n Q'(1 - y_j)},$$

где Q' — произвольная функция квантификации Q , а y_i — упорядоченные по убыванию значения аргументов. Таким образом, большой вес получают те упорядоченные аргументы, для которых значение производной в точке $(1 - y_i)$ выше.

1.3. Метод на основе редукции экстремальных значений

Этот подход делает веса зависимыми не только от порядка, но и от значений самих аргументов, как в предыдущем случае, однако позволяет сосредоточить внимание на «средних» оценках [1], не принимая во внимание экстремальные. Задаются два параметра $\alpha, \beta \in [0, 1]$, $\alpha < \beta$. Весовые коэффициенты определяются таким образом, что они игнорируют первые $n\alpha$ и последние $n - (n\beta + 1)$ значений, а внимание акцентируется лишь на тех оценках, которые находятся между экстремальными. Данный OWA-оператор называется EVROWA-оператором, для которого всегда $orness = 0.5$ [8], что соответствует сбалансированной, нейтральной стратегии, не склонной к оптимизму, ни к пессимизму, и может быть полезен в контексте поиска устойчивых решений, менее зависимых от выбросов данных.

Выбор конкретного метода определения весов позволяет гибко настраивать процедуру агрегирования в соответствии с семантикой задачи и предпочтениями ЛПР.

2. Пример применения: выбор инвестиционного проекта

В данном разделе приведем практический пример. Рассмотрим задачу выбора наилучшего инвестиционного проекта по благоустройству городской территории из 12 предложенных к рассмотрению. Каждый проект оценивается по следующим шести показателям: P1 — стоимость реализации (млн руб.); P2 — срок окупаемости (лет); P3 — чистый дисконтированный доход (млн руб.); P4 — внутренняя норма доходности (%); P5 — индекс рентабельности; P6 — уровень риска. Исходная информация о проектах представлена в табл. 1 (проекты закодируем буквами алфавита).

Таблица 1

Исходные данные

Проекты	P1	P2	P3	P4	P5	P6
A	18	4,5	25	25	1,8	6
B	12	3	22	28	2,1	4
C	8	2	17	22	1,9	2
D	15	4	30	35	2,5	6
E	15	1,8	28	26	2,3	8
F	20	5	35	30	2	3
G	11	2,8	20	27	2,2	5
H	7	1,5	10	20	1,7	1
I	19	3,5	25	32	2,4	7
J	13	3,2	28	29	2,6	4
K	16	4,2	32	33	2,7	6
L	17	3,9	35	38	2,8	3

На основе исходных данных и использования методов, описанных ранее, получим визуализацию сравнения OWA-оценок по методам расчета весов (рис. 1, 2).

На рис. 2 видно, что в соответствии с различными методами можно получить разные ранжирования, что подчеркивает важность выбора весовых коэффициентов для данного семейства операций агрегирования.

Проведя анализ данных и вычислив среднюю обобщенную оценку, можно выделить 3 лучших варианта инвестиционного проекта, из которых ЛПР должен выбрать наиболее подходя-

щий с учетом дополнительной информации и внешних факторов. Для этого составим общую среднюю OWA-оценку (рис. 3).

В рамках практического примера, получили, что при исходных данных лучшим инвестиционным проектом является проект L. Данный проект хорошо сбалансирован высокими показателями ключевых критериев и его оценка является более устойчивой к изменению стратегий агрегирования.



Рис. 1. Термальная карта OWA-оценок

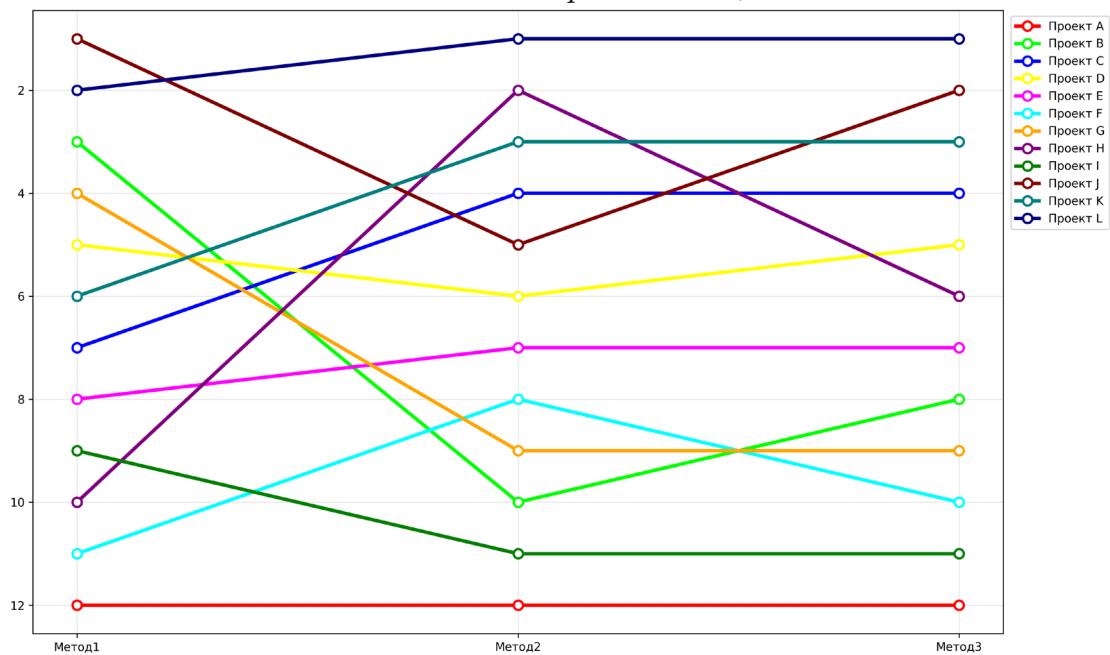


Рис. 2. Динамика изменения рангов



Рис. 3. Лучшие проекты по общей оценке

Заключение

В статье показано, что использование OWA-операторов представляет гибкий и семантически насыщенный аппарат для построения обобщенных оценок в задачах многокритериального (многоатрибутного) выбора. Различные методы определения весовых коэффициентов, основанных на функции квантификации, их производных, редукции экстремальных значений, позволяют целенаправленно формировать стратегию агрегирования, отражающую отношение лица, принимающего решение, к риску и его предпочтения.

Практический пример по выбору инвестиционного проекта показал, что в зависимости от выбранной стратегии, наиболее качественные проекты будут лидировать при любом из рассмотренных подходов. Это свидетельствует об устойчивости решения. Использование нескольких методов расчета весов и последующий сравнительный анализ позволяют принять более обоснованное и взвешенное решение.

Литература

1. Леденева Т. М. Обзор основных классов операторов порядкового взвешенного агрегирования / Т. М. Леденева, И. А. Левкина // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2022. – № 1. – С. 4–8.
2. Леденева Т. М. Агрегирование информации в оценочных системах / Т. М. Леденева, С. Л. Подвальный // Вестник ВГУ. Серия: Системный анализ и информационные технологии. – 2016. – № 4. – С. 2–7.
3. Kolmogorov A. Sur la notion de moyenne / A. Kolmogorov // Atti delle Reale Accademia Nazionale dei Lincei Mem. Cl. Sci. Mat. Natur. Sez. – 1930. – Vol. 12. – P. 323–343.
4. Mayor G. On the representation of some Aggregation functions / G. Mayor, E. Trillas // Proceeding of ISMVL. – 1986. – P. 111–114.
5. Mesiar R. Aggregation Operators / R. Mesiar, M. Komornikova // Proceeding of the XI Conference on applied Mathematics PRIM' 96 / ed. By D. Herceg, K. Surla. – Novi Sad: Institute of Mathematics, 1997. – P. 193–211.

6. *Yager R. R.* Full reinforcement operators in aggregation techniques / R. R. Yager, A. Rybalov // IEEE Transactions on Systems, Man and Cybernetics. – 1998. – Vol. 28. – P. 757–769.
7. *Yager R. R.* A new methodology for ordinal multiple aspect decisions based on fuzzy sets / R. R. Yager // Decision Sciences. – 1981. – Vol. 12. – P. 589–600.
8. *Garcia-Zamora D.* An ordered weighted averaging operator based on extreme values reductions / D. Garcia-Zamora, A. Labello, R. M. Rodriguez, L. Martinez // Atlantis Studies in Uncertainty Modelling. – 2021. – Vol. 3. – P. 290–297.

АНАЛИЗ ИНТЕЛЛЕКТУАЛЬНЫХ ИНСТРУМЕНТОВ ДЛЯ 3D КОНСТРУИРОВАНИЯ В АДДИТИВНОМ ПРОИЗВОДСТВЕ

А. В. Фролов, О. Ю. Марьясин

Ярославский государственный технический университет

Аннотация. В работе представлены результаты экспериментального сравнения современных инструментов на базе искусственного интеллекта для 3D конструирования изделий, предназначенных для аддитивного производства. Проведено тестирование пяти ИИ-платформ различной архитектуры на трех типах объектов различной геометрической сложности. Лучшие результаты для простых геометрических форм продемонстрировала платформа Luma AI, для объектов средней сложности — Kandinsky 3D. Для количественной оценки влияния сложности объектов на качество генерации был введен индекс геометрической сложности. Установлена сильная корреляция между индексом геометрической сложности объекта и точностью генерации. Разработаны практические рекомендации по выбору интеллектуальных инструментов для различных типов проектных задач в аддитивном производстве.

Ключевые слова: аддитивное производство, 3D моделирование, искусственный интеллект, генеративные нейронные сети, диффузионные модели, автоматизация проектирования, индекс геометрической сложности.

Введение

Современное развитие технологий аддитивного производства в сочетании с достижениями в области искусственного интеллекта (ИИ) формирует новую парадигму промышленного проектирования. Согласно аналитическим данным, объем мирового рынка 3D-печати достиг 15,05 млрд долларов в 2024 году, демонстрируя устойчивый рост на 10% в годовом исчислении [1]. Российская Федерация, несмотря на текущую долю 2% в мировом рынке аддитивных технологий, показывает десятикратное увеличение объема рынка за последние восемь лет, что свидетельствует о высоком потенциале интеграции ИИ-технологий в отечественные производственные процессы [2].

Традиционные методы проектирования изделий для аддитивного производства основываются на использовании систем автоматизированного проектирования (Computer-Aided Design — CAD), таких как AutoCAD, SolidWorks, Fusion 360. Однако создание сложных геометрических форм, оптимизированных под специфику аддитивного производства, требует значительных трудозатрат и высокой квалификации специалистов. ИИ-технологии предлагают инновационные решения для оптимизации процессов 3D-печати сложных геометрий и легких структур [3].

Критический анализ современного состояния отрасли выявляет существенный разрыв между теоретическими возможностями ИИ-инструментов для 3D моделирования и их практической применимостью в реальных инженерных задачах. Большинство публикаций сосредоточены на демонстрации технологических возможностей без строгих количественных метрик точности и надежности. Согласно прогнозам экспертов, в течение последующих пяти лет AI-генерированные модели способны достичь качества, сопоставимого с созданными человеком, для приблизительно 60 % базовых применений [4]. Однако отсутствие систематических экспериментальных исследований препятствует объективной оценке готовности данных технологий к промышленному внедрению.

В данной работе представлены результаты экспериментального сравнения современных ИИ-инструментов для 3D конструирования изделий, предназначенных для аддитивного производства.

1. Подготовка и проведение экспериментов

1.1. Выбор экспериментальных объектов

Для обеспечения репрезентативности результатов экспериментов и возможности их экспондукции на широкий спектр промышленных применений были выбраны три типа объектов различной геометрической сложности (рис. 1).

Мяч для гольфа. Простая базовая геометрия типа сферы с регулярной микроструктурированной поверхностью (димпли), требующая высокой точности воспроизведения для обеспечения аэродинамических характеристик.

Рама дронов. Пространственная решетчатая структура с тонкостенными элементами, множественными пересечениями и крепежными узлами. Высокая топологическая сложность с внутренними полостями представляет значительный вызов для ИИ-систем.

Крыльчатка насоса. Аэродинамические профили с лопастями переменного сечения и сложной кривизной. Максимальная геометрическая сложность с критическими требованиями к точности для обеспечения аэродинамической эффективности.

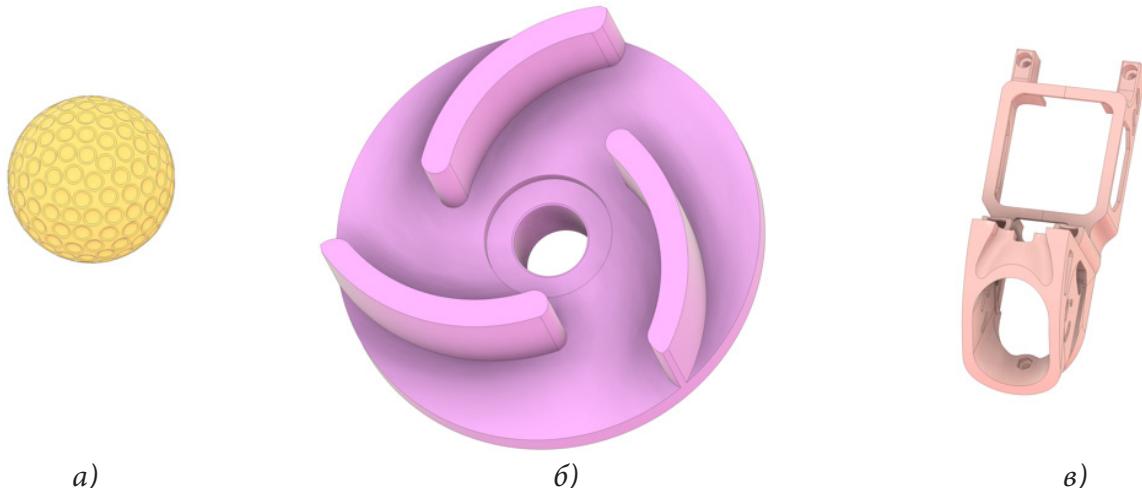


Рис. 1. Объекты для генерации: а) мяч для гольфа, б) крыльчатка насоса, в) рама дрона

1.2. Выбор ИИ-инструментов

В данном исследовании использованы пять современных ИИ-платформ различной архитектуры для автоматизации 3D моделирования. Сравнительная характеристика ИИ-инструментов представлена в табл. 1.

Model Context Protocol (MCP) обеспечивает стандартизированную интеграцию больших языковых моделей с Blender через унифицированный протокол взаимодействия [5]. Система позволяет использовать различные базовые большие языковые модели с возможностью локальной обработки и тонкой настройки.

1.3. Метрики качества

Для объективной оценки качества сгенерированных моделей использована система количественных метрик на основе сравнительного анализа 3D геометрии.

Таблица 1

Сравнение ИИ-инструментов

Инструмент	Архитектура	Ключевые преимущества	Ссылка на сайт
Kandinsky 3D	Diffusion Transformers	Хорошее понимание русского языка, API, формат .fbx	https://ai-forever.github.io/Kandinsky-3/
Hunyuan3D 2.0	Flow-based diffusion	Open source (Apache 2.0), 500K 3D объектов обучения	https://www.hunyuan-3d.com/
Meshy AI	Закрытая архитектура	Высокая скорость генерации, форматы .fbx/.obj/.stl/.gltf	https://www.meshy.ai/
Luma AI Genie	Neural Radiance Fields	Бесплатный доступ, креативная генерация	https://lumalabs.ai/
MCP Blender	Model Context Protocol	Локальная обработка, полный контроль, интеграция с Blender	https://github.com/teslaproduuction/GenesisCore

Средняя абсолютная ошибка (MAE):

$$MAE = \frac{1}{n} \sum_{i=1}^n |d_i|,$$

где d_i — кратчайшее расстояние от точки i генерированной модели до поверхности эталонной модели, n — общее количество точек выборки. Данная метрика обеспечивает оценку типичной величины отклонения без учета направления.

Среднеквадратическая ошибка (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n d_i^2}.$$

Стандартное отклонение (σ):

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \bar{d})^2},$$

где \bar{d} — среднее арифметическое всех расстояний.

Процентильные характеристики: P50 (медиана) характеризует типичное отклонение, P95 (95-й процентиль) значение, ниже которого находится 95 % всех измеренных отклонений, P99 (99-й процентиль) значение, ниже которого находится 99 % всех измеренных отклонений.

1.4. Подготовка эталонных моделей и процедура эксперимента

Эталонные модели получены из коммерческих 3D библиотек, профессиональных CAD-систем (SolidWorks, Fusion 360) и высокоточных 3D сканов с разрешением до 0,1 мм. Все модели нормализованы для исключения влияния положения в пространстве.

Для обеспечения воспроизводимости разработан набор стандартизованных текстовых запросов (промптов). Например, базовый промпт для мяча для гольфа: "Professional golf ball with dimpled surface, white color, regulation size 42.67 mm diameter, 336 dimples, spherical shape, high quality 3D model for manufacturing". Каждый объект генерировался трижды с идентичными параметрами для оценки стабильности.

2. Результаты экспериментального тестирования

2.1. Генерация мяча для гольфа

Результаты генерации мяча для гольфа представлены в табл. 2. Анализ охватывал датасеты от 8759 до 1142786 точек для различных ИИ-инструментов.

Таблица 2

Характеристики точности генерации мяча для гольфа (мм)

Инструмент	Точек	MAE	RMSE	σ	Медиана	P95	P99	Max
Luma AI	24586	0,258	0,398	0,318	0,148	0,801	0,933	1,106
Meshy AI	8759	0,268	0,423	0,331	0,193	0,759	0,995	1,386
MCP Blender	1142786	0,278	0,447	0,362	0,165	0,767	0,873	1,007
Hunyuuan3D 2.0	321910	0,295	0,463	0,369	0,236	0,762	0,961	1,292
Kandinsky 3D	20002	0,333	0,521	0,423	0,269	0,837	1,015	1,293

Лучшие результаты продемонстрировал Luma AI с наименьшим средним абсолютным отклонением 0,258 мм, что на 22,5 % лучше результата Kandinsky 3D (0,333 мм). Kandinsky 3D продемонстрировал стабильные результаты с адекватным воспроизведением базовой сферической формы и корректной интерпретацией промптов на русском языке, но с приблизительным воспроизведением димплов, с упрощенной геометрией. Hunyuuan3D 2.0 продемонстрировал высокое качество с детализированной димпловой структурой и с превосходным топологическим качеством. Meshy AI показал оптимальный баланс между скоростью и качеством с высокой повторяемостью результатов.

2.2. Генерация рамы дронов

Рама дронов представляет существенно более сложную задачу вследствие множественных тонкостенных элементов и требований к топологической корректности. Результаты генерации представлены в табл. 3.

Таблица 3

Характеристики точности генерации рамы дронов (мм)

Инструмент	Точек	MAE	RMSE	σ	Медиана	P95	P99	Max
Kandinsky 3D	19964	0,793	1,267	0,908	0,679	1,867	3,265	4,787
Hunyuuan3D 2.0	127038	1,510	3,024	2,556	0,621	7,405	11,622	15,397

Kandinsky 3D продемонстрировал значительно лучшие результаты (MAE = 0,793 мм) по сравнению с Hunyuuan3D 2.0 (MAE = 1,510 мм). Kandinsky 3D успешно интерпретирует общую концепцию рамы квадрокоптера, но упрощает сложные соединительные узлы и создает монолитные конструкции. Следует отметить, что Hunyuuan3D 2.0 демонстрирует более детализированную генерацию сложных элементов конструкции, что приводит к увеличению локальных отклонений в областях высокой геометрической сложности. Hunyuuan3D 2.0 показывает превосходное качество детализации и лучшее среди тестируемых систем воспроизведение симметрии конструкции, но испытывает сложности с внутренними полостями.

2.3. Генерация крыльчатки

Крыльчатка представляет наивысший уровень сложности среди тестируемых объектов вследствие требований к точности аэродинамических профилей. Результаты представлены в табл. 4.

Таблица 4

Характеристики точности генерации крыльчатки (мм)

Инструмент	Точек	MAE	RMSE	σ	Медиана	P95	P99	Max
Kandinsky 3D	20000	1,054	1,917	1,617	0,505	3,651	5,607	7,893
Hunyuan3D 2.0	212876	2,095	3,437	2,643	1,718	7,047	7,402	9,127

Генерация крыльчатки выявила ограничения существующих ИИ-технологий — оба инструмента показали значительные отклонения, превышающие типичные производственные допуски для аэродинамических поверхностей ($\pm 0,1$ – $0,2$ мм). Все системы демонстрируют тенденцию к упрощению сложных аэродинамических профилей и испытывают сложности в воспроизведении переменного угла атаки лопастей.

3. Анализ результатов

3.1. Влияние геометрической сложности

Для количественной оценки влияния сложности объектов на качество ИИ-генерации авторами, на основе работы [6], был введен индекс геометрической сложности (Geometrical Complexity Index — GCI):

$$GCI = w_1 m_1 + w_2 m_2 + w_3 m_3,$$

где $m_1 = \frac{N_f}{A}$ — характеризует степень детализации поверхности, N_f — количество граней, A — площадь поверхности, $m_2 = \frac{C}{L}$ — характеризует топологическую сложность, C — общая длина кромок, L — характерный размер, $m_3 = \frac{V_c}{V_t}$ — характеризует наличие внутренних полостей и отверстий, V_c — объем полостей, V_t — общий объем.

Весовые коэффициенты $w_i, i=1, \dots, 3$ интерпретируются как доли вклада каждой компоненты в общую сложность объекта:

- $w_1 = 0,0002$ (0,02 %) — плотность сетки. Современные ИИ-модели эффективно обрабатывают детализированные поверхности благодаря работе в латентном пространстве, поэтому данный фактор вносит минимальный вклад в сложность.

- $w_2 = 0,0028$ (0,28 %) — длина кромок. Топологическая сложность влияет на точность воспроизведения геометрических переходов, но её вклад также относительно невелик по сравнению с полостями.

- $w_3 = 0,997$ (99,7 %) — доля полостей. Доминирующий фактор сложности. Анализ дефектов показал, что non-manifold геометрия возникает в 78 % случаев при наличии внутренних полостей. Внутренние полости требуют понимания объемной структуры, что является фундаментальной проблемой для текущего поколения генеративных моделей.

Соотношение весов: $w_3 : w_2 : w_1 = 4985 : 14 : 1$ отражает иерархию сложности, в которой, по мнению авторов, внутренние полости примерно в 5000 раз критичнее плотности сетки, а топологическая сложность в 14 раз важнее детализации.

Геометрические параметры эталонных моделей приведены в табл. 5. Значения параметров N_f , A , C извлечены из STL-файлов эталонных моделей с использованием библиотеки обработки 3D данных.

Значения GCI и усредненные, для различных ИИ-инструментов, значения метрик MAE и RMSE приведены в табл. 6.

Таблица 5

Геометрические параметры эталонных моделей

Параметр	Мяч для гольфа	Рама дрона	Крыльчатка
N_f	~20000	~40000	~95000
$A, \text{ см}^2$	60	380	265
$C, \text{ см}$	450	2300	2050
$L, \text{ см}$	4,27	26	14
$V_c, \text{ см}^3$	0	120	88
$V_p, \text{ см}^3$	40,6	250	175

Таблица 6

Значения CGI и средние характеристики точности

Объект	GCI	Средняя MAE, мм	Средняя RMSE, мм
Мяч для гольфа	0,37	0,285	0,450
Рама дронов	0,75	1,151	2,146
Крыльчатка	0,99	1,575	2,677

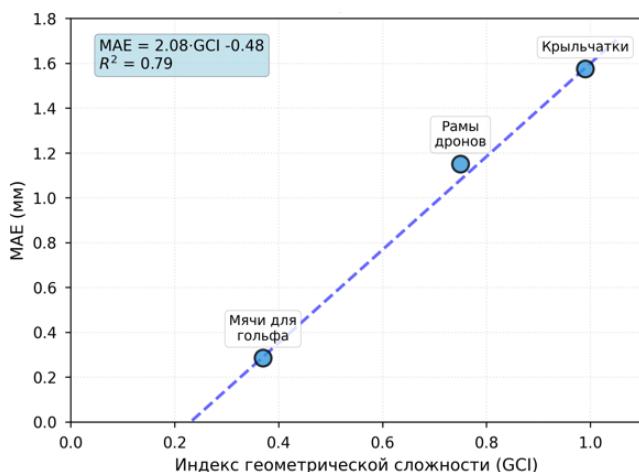
Обнаружена сильная положительная корреляция между индексом геометрической сложности и ошибками генерации: для средней абсолютной ошибки коэффициент корреляции Пирсона $r = 0,89$ ($R^2 = 0,79$, $p < 0,01$), для среднеквадратичной ошибки $r = 0,94$ ($R^2 = 0,89$, $p < 0,01$).

На основе экспериментальных данных получены уравнения линейной регрессии:

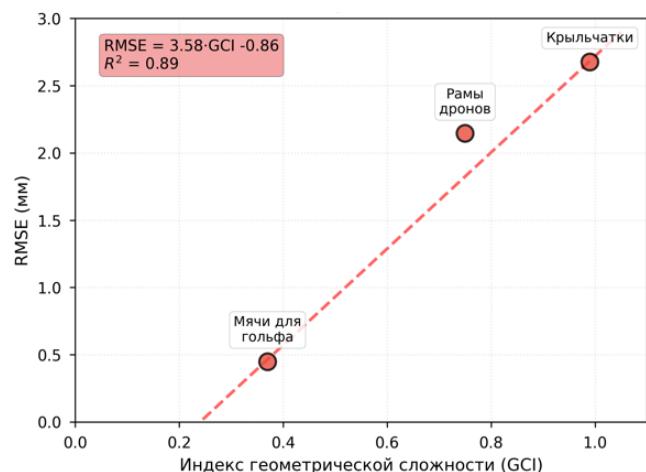
- Для MAE: $MAE = 2,08 * GCI - 0,48$;
- Для RMSE: $RMSE = 3,58 * GCI - 0,86$.

Графическое представление зависимостей метрик MAE и RMSE от GCI показано на рис. 2. Установлены следующие пороговые значения GCI для практического применения:

- $GCI < 0,5$: $MAE < 0,3$ мм (высокая точность, рекомендуется для прямого использования ИИ-генерации);
- $0,5 \leq GCI < 0,8$: $MAE \approx 0,8 - 1,2$ мм (средняя точность, требуется проверка критических размеров);
- $GCI \geq 0,8$: $MAE > 1,5$ мм (низкая точность, необходима CAD-доработка).



a)



б)

Рис. 2. Зависимость между индексом геометрической сложности и точностью генерации:
а) CGI с MAE, б) CGI с RMSE

Полученные результаты демонстрируют, что нормализованный индекс GCI является эффективным инструментом для априорной оценки применимости ИИ-инструментов к конкретным проектным задачам в аддитивном производстве.

3.2. Стабильность и воспроизводимость результатов

Для оценки стабильности и воспроизводимости результатов использовался коэффициент вариации. Коэффициент вариации при повторных генерациях варьировался от 8,9 % (MCP Blender для мячей) до 31,2 % (Hunyuan3D 2.0 для крыльчаток). Наименьшую стабильность продемонстрировали все системы при генерации крыльчаток, что указывает на сложность воспроизведения аэродинамической геометрии.

4. Практические рекомендации

На основе результатов проведенного исследования разработаны рекомендации по применению ИИ-инструментов для различных типов задач (табл. 7).

Таблица 7

Рекомендуемые ИИ-инструменты по типам задач

Тип задачи	Рекомендуемый инструмент	GCI	Ожидаемая точность, мм
Простые геометрические формы	Luma AI	< 0,5	±0,25
Технические детали средней сложности	Meshy AI	0,5–0,7	±0,27
	Kandinsky 3D	0,5–0,8	±0,33–0,79
Высокодетализированные модели	Hunyuan3D 2.0	0,6–0,8	±0,30
Критические размеры	MCP Blender	0–1	±0,28

Заключение

В результате проведенного экспериментального исследования получены следующие результаты:

- Для простых геометрических форм ($GCI < 0,5$) лучшие результаты достигаются при использовании Luma AI ($MAE = 0,258$ мм) и Meshy AI ($MAE = 0,268$ мм). Для объектов средней сложности ($0,5 \leq GCI < 0,8$) оптimalен Kandinsky 3D ($MAE = 0,793$ мм).
- Установлена сильная корреляция между индексом геометрической сложности и точностью генерации ($r = 0,89$, $p < 0,01$).
- Стабильность генерации варьируется от 8,9 % до 31,2 %, что является приемлемым для промышленного применения при соответствующем контроле качества.
- Отечественные ИИ-решения (Kandinsky 3D) демонстрируют приемлемое качество с преимуществами в области информационной безопасности и технологической независимости.

Для промышленного применения рекомендуется гибридный подход — ИИ-генерация базовой формы с последующей CAD-доработкой. Экспериментально установлена средняя экономия времени 35–40 % при использовании такого подхода по сравнению с традиционным проектированием при сохранении требуемого качества. Рекомендуемая постобработка включает топологическую очистку, проверку минимальных размеров элементов и оптимизацию ориентации для печати.

Результаты исследования могут быть использованы предприятиями при выборе ИИ-инструментов для автоматизации процессов проектирования в аддитивном производстве.

Литература

1. TAdviser. 3D-печать (Аддитивное производство). – URL: [https://www.tadviser.ru/index.php/Статья:3D-печать_\(Аддитивное_производство\)](https://www.tadviser.ru/index.php/Статья:3D-печать_(Аддитивное_производство)) (дата обращения: 15.10.2025)
2. Группа «ДЕЛОВОЙ ПРОФИЛЬ». Рынок технологий 3D-печати в России и мире. – 2024. – URL: <https://delprof.ru/press-center/open-analytics/> (дата обращения: 15.10.2025)
3. Soori M. Additive manufacturing modification by artificial intelligence, machine learning, and deep learning: A review / M. Soori, F.K.G. Jough, R. Dastres [et al.] // Additive Manufacturing Frontiers. – 2025. – Vol. 2. – P. 1–17. – DOI: <https://doi.org/10.1016/j.amf.2025.200198>
4. Westpha E. Generative Artificial Intelligence: Analyzing Its Future Applications in Additive Manufacturing / E. Westpha, H. Seitz // Big Data and Cognitive Computing. – 2024. – Vol. 8 (7). – P. 1–21. – DOI: <https://doi.org/10.3390/bdcc8070074>
5. Anthropic. Introducing the Model Context Protocol. – URL: <https://www.anthropic.com/news/model-context-protocol> (дата обращения: 15.10.2025)
6. Greco A. Complexity-driven product design: part 1 –methodological framework and geometrical complexity index / A. Greco, P. Mancuso, M.B. Russo [et al.] // International Journal on Interactive Design and Manufacturing. – 2024. – Vol. 18. – P. 5441–5456.

МЕТОДЫ АНАЛИЗА РАБОЧЕЙ НАГРУЗКИ ТАБЛИЦ И ИДЕНТИФИКАЦИЯ КАНДИДАТОВ ДЛЯ ПАРТИЦИОНИРОВАНИЯ В РАСПРЕДЕЛЕННЫХ РЕЛЯЦИОННЫХ СУБД

Ю. А. Фролов, И. Е. Воронина

Воронежский государственный университет

Аннотация. В условиях роста объемов данных ключевой задачей становится эффективное управление рабочей нагрузкой в распределенных реляционных СУБД. В работе представлены специализированные алгоритмы автоматического идентификации таблиц-кандидатов для партиционирования, образующие основной элемент. Актуальность решения подтверждается возможностью существенного повышения производительности обработки данных через рациональное распределение нагрузки и адаптацию к изменяющимся паттернам доступа.

Ключевые слова: анализ рабочей нагрузки, автоматическое партиционирование, распределенные реляционные СУБД, большие объемы данных, распределение нагрузки, производительность СУБД.

Введение

Обработка постоянно растущих объемов данных представляет серьезную проблему для современных распределенных реляционных систем управления базами данных. Одним из наиболее эффективных подходов к повышению производительности таких систем является партиционирование таблиц, позволяющее распределять данные между узлами кластера. Однако традиционные методы партиционирования, основанные на ручном проектировании и экспертной оценке, оказываются недостаточно эффективными в условиях динамически изменяющихся рабочих нагрузок и сложных распределенных архитектур.

Основной задачей является анализ методов и алгоритмов для автоматического анализа рабочей нагрузки и идентификации кандидатов для партиционирования в распределенных реляционных СУБД, направленных на повышение эффективности обработки данных за счет рационального распределения нагрузки.

Актуальность работы определяется растущей потребностью в разработке автоматизированных систем партиционирования, способных адаптироваться к изменяющимся паттернам доступа и вариативным нагрузкам в распределенных реляционных СУБД. Практическая ценность подтверждается возможностью прямой интеграции алгоритмов в промышленные СУБД для построения системы автоматизированного управления данными, которая снижает операционные затраты за счет минимизации ручного администрирования и оптимизации использования вычислительных мощностей.

1. Анализ существующих решений

В ходе работы был проведен анализ современных распределенных реляционных СУБД и расширений для PostgreSQL, а именно CockroachDB, YugabyteDB, TiDB и Citus [1–4].

CockroachDB предоставляет встроенные механизмы партиционирования на основе диапазонов ключей, отслеживание базовых метрик производительности и сбор статистики выполнения запросов. Однако система не реализует интеллектуальную оптимизацию схемы партиционирования и не обеспечивает адаптивности партиций к нагрузкам.

YugabyteDB предлагает инструменты для настройки партиционирования с поддержкой различных стратегий распределения данных и мониторинг производительности. Система обладает возможностями ручной ребалансировки, но не обеспечивает автоматической адаптации партиций к изменяющимся нагрузкам.

TiDB обеспечивает развитые механизмы партиционирования и комплексный анализ выполнения запросов. Система предоставляет элементы интеллектуальной оптимизации, однако не поддерживает автоматическое изменение схемы партиционирования при изменении паттернов доступа, что ограничивает её адаптивность.

Citus реализует партиционирование для PostgreSQL с поддержкой сбора метрик производительности. Система требует ручного управления партициями и не предоставляет возможностей автоматической адаптации к изменяющимся рабочим нагрузкам.

На основании анализа можно сделать вывод, что ни одна из рассмотренных систем не предлагает комплексного решения для автоматической адаптации схемы партиционирования на основе анализа рабочих нагрузок.

В табл. 1 представлены результаты сравнения функциональности существующих распределенных реляционных СУБД.

Таблица 1

Анализ существующих распределенных реляционных СУБД

Название СУБД	Поддержка партиционирования	Анализ рабочей нагрузки	Интеллектуальная оптимизация	Адаптивность партиций к нагрузкам
CockroachDB	+	+	-	-
YugabyteDB	+	+	-	-
TiDB	+	+	+	-
Citus	+	+	-	-

2. Методология анализа рабочей нагрузки

Методология, ориентированная на автоматическое определение кандидатов для партиционирования в распределенных реляционных СУБД, включает трехэтапный подход к сбору и анализу метрик производительности:

1) Сбор и агрегация метрик — этап, на котором система отслеживает: временные характеристики запросов, частотные метрики, метрики распределения и ресурсные показатели.

2) Анализ и идентификация паттернов — этап, на котором происходит корреляционный анализ между метриками, выявление временных паттернов (часовых, дневных, недельных), обнаружение узких мест производительности, прогнозирование роста нагрузок на основе исторических данных.

3) Принятие решений и рекомендации — этап, на котором система рассчитывает показатель необходимости партиционирования, сравнивает показатели с пороговыми значениями для принятия решений, разрабатывает план для бесшовного внедрения партиций [5].

Ключевые метрики для анализа рабочей нагрузки:

1) Метрики производительности запросов:

- время выполнения запросов;
- частота обращений к таблицам;
- количество операций чтения/записи;
- распределение типов запросов (SELECT, UPDATE, INSERT, DELETE).

2) Метрики распределения данных:

- размер таблиц и индексов;

- распределение данных по узлам кластера;
- коэффициент селективности запросов;
- количество уникальных значений в колонках.

3) Метрики доступа к данным:

- паттерны временных рядов;
- корреляция запросов;
- выявление «горячих» данных;
- частота JOIN-операций между таблицами.

3. Архитектура системы сбора метрик

Система сбора метрик состоит из семи основных компонентов: агент сбора метрик, хранилище временных рядов, аналитический движок, модуль принятия решений, распределенная реляционная СУБД, база рекомендаций, Apache Kafka (рис. 1).

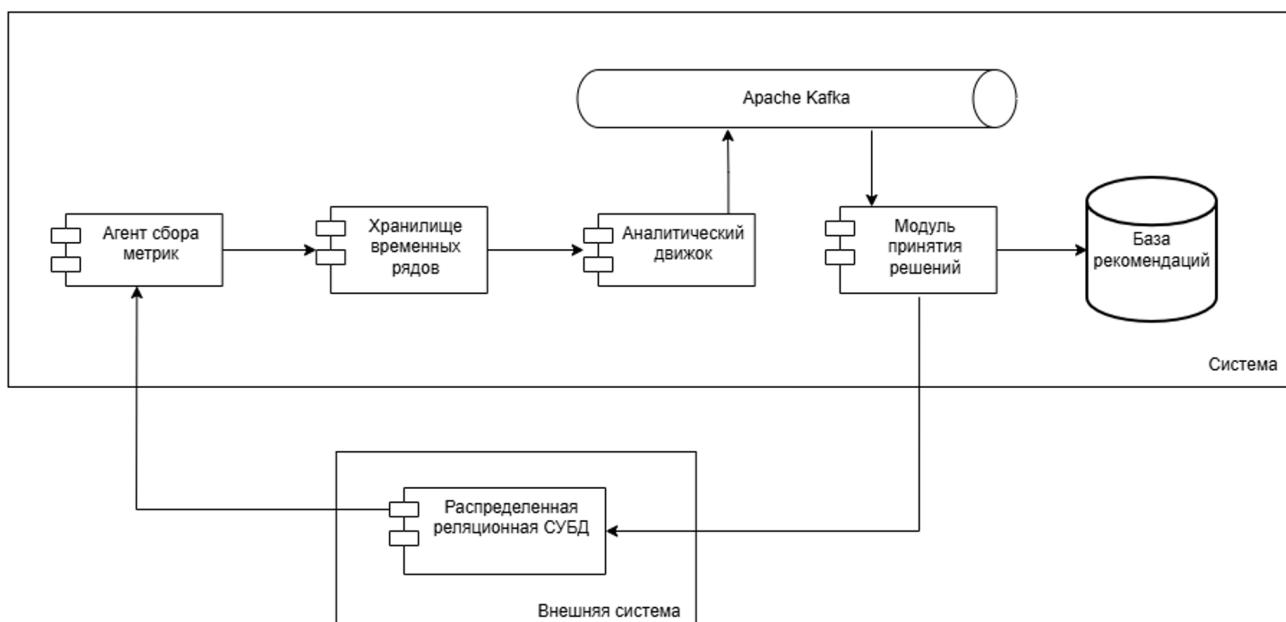


Рис. 1. Архитектура системы сбора метрик

В архитектуре представлены компоненты:

1. Агент сбора метрик — предназначен для сбора метрик производительности в реальном времени.
2. Хранилище временных рядов — отвечает за долгосрочное хранение и агрегацию исторических данных [6].
3. Аналитический движок — отвечает за обработку и анализ собранных метрик.
4. Модуль принятия решений — формирует интеллектуальные рекомендации по партиционированию.
5. Распределенная реляционная СУБД — целевая система по оптимизации партиционирования.
6. База рекомендаций — требуется для хранения и управления генерированными рекомендациями.
7. Apache Kafka — обеспечивает надежную асинхронную передачу потоковых данных.

4. Алгоритм идентификации кандидатов для партиционирования

Выведен алгоритм автоматического выявления таблиц для партиционирования, основанный на многокритериальном анализе производительности.

Многоуровневая система оценки включает анализ эффективности выполнения запросов, оценку характеристик распределения данных и изучение паттернов доступа. Для каждого направления рассчитываются специализированные метрики с адаптивными весовыми коэффициентами, которые автоматически настраиваются под специфику рабочей нагрузки и особенности конкретной распределенной СУБД [7].

Все таблицы автоматически распределяются по четырем категориям приоритета на основе анализа таких параметров, как частота обращений, равномерность распределения данных и объем хранимой информации.

Автоматический выбор стратегии партиционирования выполняется для таблиц с высоким и средним приоритетом. Система анализирует структуру данных, шаблоны запросов и особенности распределения нагрузки, чтобы рекомендовать оптимальный тип партиционирования: диапазонное для временных данных, хэш-распределение для равномерной нагрузки и списочное для категориальной информации.

Адаптивный механизм обеспечивает постоянную оптимизацию процесса принятия решений. Система непрерывно оценивает эффективность реализованных изменений, сравнивая метрики производительности до и после партиционирования, и корректирует пороговые значения и весовые коэффициенты для улучшения точности будущих рекомендаций.

Заключение

Проведенный анализ подтвердил актуальность разработки алгоритма для автоматического партиционирования в распределенных реляционных СУБД. Анализ существующих решений показал, что современные системы, включая CockroachDB, YugabyteDB, TiDB и Citus, не обеспечивают автоматизацию процесса партиционирования на основе анализа рабочих нагрузок.

В работе предложена методология анализа рабочей нагрузки, включающая трехэтапный подход: сбор и агрегацию метрик, идентификацию паттернов и принятие решений. Спроектирована архитектура системы сбора метрик, состоящая из семи взаимосвязанных компонентов, обеспечивающих непрерывный мониторинг производительности и формирование рекомендаций.

Ключевым результатом работы является вывод алгоритмов идентификации кандидатов для партиционирования, основанных на многокритериальной оценке таблиц. Предложенная система автоматически анализирует метрики производительности запросов, распределения данных и паттернов доступа, классифицирует таблицы по приоритетам и определяет оптимальную стратегию партиционирования.

Литература

1. CockroachDB Documentation [Интернет-портал] URL: <https://www.cockroachlabs.com/docs>. (дата обращения 10.11.2025)
2. YugabyteDB Documentation [Интернет-портал] URL: <https://docs.yugabyte.com/>. (дата обращения 10.11.2025)
3. TiDB Documentation [Интернет-портал] URL: <https://docs.pingcap.com/>. (дата обращения 11.11.2025)
4. Citus Documentation [Интернет-портал] URL: <https://docs.citusdata.com/en/v13.0/>. (дата обращения 11.11.2025)

5. Таненбаум Э. Распределенные системы. Принципы и парадигмы / Э. Таненбаум, М. ван Стейн. – 2-е изд. – СПб.: Питер, 2003. – 877 с.
6. Петров А. Распределенные данные. Алгоритмы работы современных систем хранения информации / А. Петров. – 1-е изд. – СПб.: Питер, 2021. – 336 с.
7. Фоккинк У. Распределенные алгоритмы. Интуитивный подход / У. Фоккинк. – 1-е изд. – СПб.: Питер, 2017. – 272 с.

АНАЛИЗ И СРАВНЕНИЕ СОВРЕМЕННЫХ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ ДЛЯ ДЕТЕКЦИИ ТРАНСПОРТНЫХ СРЕДСТВ

А. В. Холкин

Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ

Аннотация. Настоящая работа посвящена сопоставительному анализу современных детекторов объектов, применимых для задачи детекции транспортных средств в видеопотоках городских камер. Рассматриваются архитектуры RT-DETR, D-FINE-X и YOLOv10. Сформирована единая методика оценки, включающая метрики точности mAP, задержку вывода и показатели вычислительных ресурсов, а также проведён натурный эксперимент на часовой видеозаписи городского перекрёстка. По совокупности метрик наилучший практический баланс для многопоточной обработки видеопотоков показали варианты YOLOv10-s и YOLOv10-m. Результаты и рекомендации могут быть использованы при проектировании муниципальных систем мониторинга дорожной обстановки.

Ключевые слова: детектирование объектов, транспортные средства, RT-DETR, D-FINE-X, YOLOv10, интеллектуальные транспортные системы, ИТС, городской видеопоток.

Введение

В последние годы городские ИТС (интеллектуальные транспортные системы) эволюционируют от ретроспективного анализа к оперативному мониторингу дорожной обстановки. Центральным компонентом таких систем является модуль детекции транспортных средств, определяющий границы и классы объектов на кадрах видеопотока. В литературе представлено свыше двухсот архитектур детекции/сегментации, причём наиболее результативные направления развивают идеи семейства DETR и линий YOLO, ориентируясь на лучшее соотношение «точность-скорость» и отказ от подавления немаксимумов на этапе вывода.

Цель настоящей статьи — провести целостное сравнение трёх современных подходов к детекции транспортных средств (RT-DETR, D-FINE-X и YOLOv10) на основании единых метрик и экспериментальных условий, выявить сильные и слабые стороны и сформулировать практические рекомендации для развертывания в городских ИТС.

1. Модели и их архитектурные особенности

Согласно работам [1] и [2], существует более 200 различных архитектурных подходов в области глубокого обучения, направленных на решение задач детекции и сегментации объектов. Наиболее эффективные из них развиваются изначальные архитектурные принципы, заложенные первыми версиями моделей DETR и YOLOv1. К таким относятся RT-DETR, D-FINE-X и YOLOv10, которые и будут рассмотрены в данной работе.

RT-DETR. Эта модификация семейства DETR нацелена на работу в реальном времени. Архитектура использует гибридный блок кодирования признаков: объединяются как взаимодействия внутри одного масштаба, так и слияние признаков между масштабами [3]. Для запуска блока декодирования применяется отбор запросов с минимальной неопределённостью, что улучшает начальную инициализацию. Существенная особенность — «сквозная» постановка обучения и вывода, позволяющая обходиться без этапа подавления немаксимумов. Тем самым модель выдаёт конечный набор рамок и классов без дополнительной постобработки [4].

D-FINE-X. В этой линии регрессия ограничивающих рамок формулируется в вероятностной форме: для каждого края рамки поддерживается распределение, которое поэтапно уточ-

няется слоями декодирования (детализированное уточнение распределений). Чтобы ранние слои получали максимально корректный сигнал о положении объекта, используется самоди-стилляция, ориентированная на глобально оптимальную локализацию. В результате снижаются ошибки позиционирования при умеренной вычислительной нагрузке [5].

YOLOv10. Ключевая идея — согласованное «двойное сопоставление» на разных стадиях: при обучении используется схема «один-ко-многим», а при выводе — «один-к-одному» [6]. Это делает возможным отказ от подавления немаксимумов без потери устойчивости предсказаний. Дополнительно применены облегчённый выходной блок классификации, раздельное уменьшение разрешения по пространственным и канальным направлениям, а также ранжированное проектирование вычислительных блоков. В старших вариантах модели используются свёртки с большим ядром и частичное самовнимание для расширения поля восприятия при контролируемой цене вычислений [7].

2. Методика сравнения

Оценка точности детекторов объектов осуществляется с использованием различных метрик. В рамках данной работы для сопоставления базовой точности использовались отчётные значения mAP на наборе данных COCO, где mAP — это среднее по классам значение AP (Average Precision — средняя точность, равная площади под кривой «точность–полнота»), усреднённое по порогам коэффициента пересечения-на-объединение (IoU) от 0,50 до 0,95 с шагом 0,05 [8]. Параллельно оценивалась задержка вывода (мс).

Для практической оценки проведён прогон часовых фрагментов видеозаписей городского перекрёстка в условиях недостаточной освещённости и средней загруженности, с измерением FPS, использования памяти/ядра графической карты (GPU) и загрузки центрального процессора (CPU).

Сопоставление выполнялось на рабочей станции с CPU Intel Core i5-13500 и GPU NVIDIA RTX 4070 Ti 12 ГБ; конфигурации моделей выбирались в стандартных вариантах (от n до x/l для YOLOv10; от n до l для D-FINE; RT-DETR-R50).

3. Результаты

3.1. Сравнение точностей детекторов

Табл. 1 демонстрирует сравнительную оценку производительности популярных моделей детекторов на наборе данных COCO [2–7].

Таблица 1

Результаты оценки точности моделей

Модель	Параметры (M)	FLOPs (G)	mAP (IoU = 50-95)	Задержка (latency), мс
D-FINE-n	4,0	7,0	42,8	2,12
D-FINE-s	10,0	25,0	48,5	3,49
D-FINE-m	19,0	57,0	52,3	5,62
D-FINE-l	31,0	91,0	54,0	8,07
RT-DETR-R50	42,0	136,0	53,1	9,2
YOLOv10-n	2,3	6,7	39,5	1,84
YOLOv10-s	7,2	21,6	46,8	2,49
YOLOv10-m	15,4	59,1	51,3	4,74
YOLOv10-l	24,4	120,3	53,4	7,28
YOLOv10-x	29,5	160,4	54,4	10,7

Для городских систем, работающих в реальном времени, важно выбрать модель, которая балансирует между точностью, задержкой и вычислительной нагрузкой. Модели с высокой точностью, такие как D-FINE-1 (54,0 % mAP) и YOLOv10-x (54,4 % mAP), обеспечивают лучшие результаты, но имеют большую задержку и требуют больше вычислительных ресурсов. Для минимизации задержки и быстрого реагирования лучше подходят модели, как D-FINE-n (2,12 мс) и YOLOv10-n (1,84 мс), которые при этом показывают немного меньшую точность. Примеры работы детекторов показаны на рис. 1.

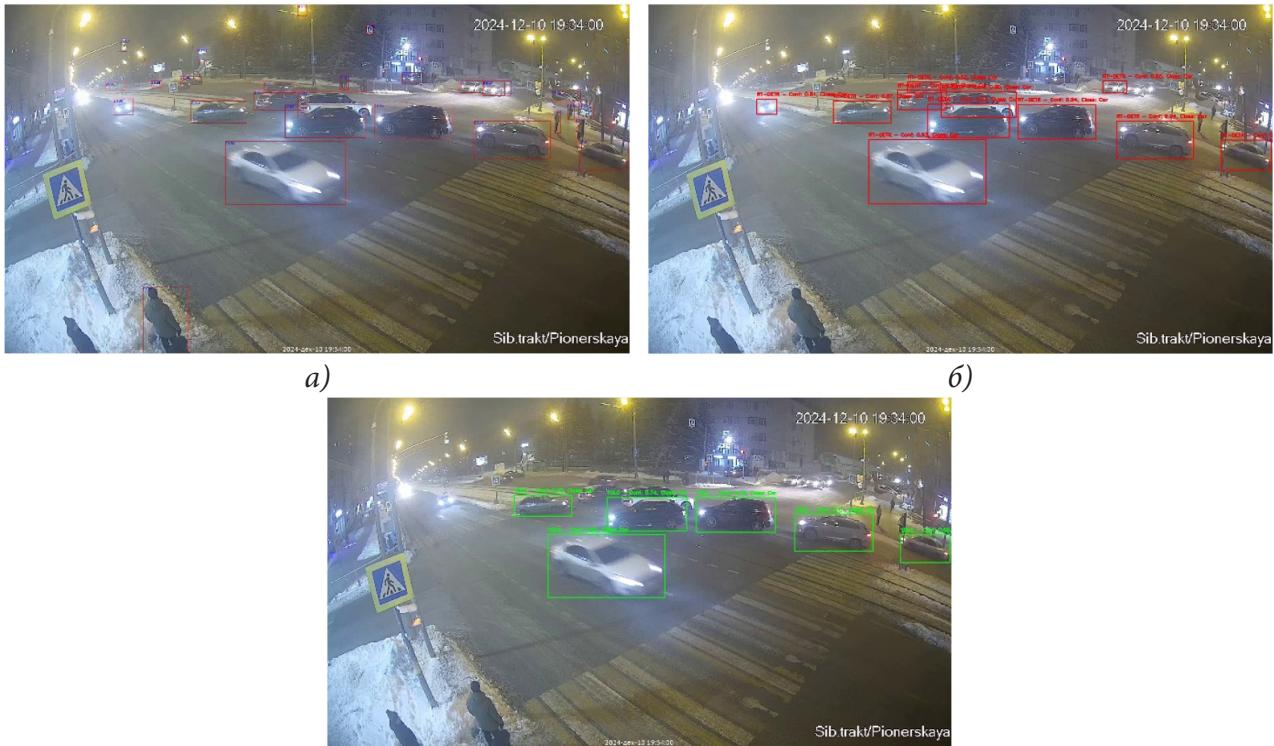


Рис. 1. Примеры работы детекторов: а) D-FINE-1, б) RT-DERT-R50 в) YOLOv10-1

Для дальнейшего использования выбранной модели для параллельной обработки кадров множества видеопотоков также важны требования к вычислительным ресурсам, используемым при анализе изображений.

3.2. Сравнение производительностей детекторов

Для сравнения требований каждой модели к вычислительным ресурсам проведем тестирование каждого решения из списка на часовых фрагментах видеозаписей с городской камерой, установленной на перекрестке в условиях плохой освещенности и среднем количестве транспортных средств. Оценка происходит по средним количеству кадров в секунду (FPS), показателям потреблениям ресурсов графического (GPU) и центрального (CPU) процессоров. Результаты тестирования представлены в табл. 2.

С учетом результатов тестирования для использования в рамках городских ИТС с возможностью масштабирования на множество параллельно обрабатываемых видеопотоков оптимальными являются модели с высокой производительностью (FPS) и умеренным потреблением ресурсов. Модели YOLOv10-s и YOLOv10-m демонстрируют наилучший баланс между высокой частотой кадров (28,0 FPS и 29,6 FPS соответственно) и сравнительно низким потреблением GPU памяти (650,07 Мб и 763,71 Мб), что делает их предпочтительными для использования в системах с ограниченными вычислительными мощностями и для масштабирова-

Таблица 2

Результаты оценки производительности моделей

Модель	Средний FPS	Среднее потребление памяти GPU, Мб	Среднее потребление GPU, %	Среднее потребление CPU, %
D-FINE-n	14,2	1387,22	14,75	6,92
D-FINE-s	12,6	1685,81	18,76	6,16
D-FINE-m	11,2	2134,96	21,39	7,3
D-FINE-l	9,6	2166,28	24,06	6,4
RT-DERT-R50	15,8	796,57	24,97	9,03
YOLOv10-n	30,9	496,96	16,02	9,09
YOLOv10-s	28,0	650,07	20,53	7,69
YOLOv10-m	29,6	763,71	14,15	7,92
YOLOv10-l	25,1	847,43	25,53	8,89
YOLOv10-x	22,2	942,25	24,95	8,55

ния на несколько видеопотоков. Несмотря на меньшие требования к GPU и CPU YOLOv10-n демонстрирует значительно более низкую точность в сравнении с старшими моделями. В то же время, модели D-FINE-n и RT-DERT-R50 показывают хорошие результаты по FPS, но их потребление памяти и CPU может ограничить масштабируемость на большем числе потоков.

Заключение

Сопоставительный анализ RT-DETR, D-FINE-X и YOLOv10 показал, что в задаче детекции транспортных средств для городских ИТС наилучший компромисс «точность-скорость-ресурсы» демонстрируют YOLOv10-s/m: они сохраняют близкую к старшим моделям точность при существенно меньших издержках и позволяют масштабировать систему на множество видеопотоков. При необходимости предельной точности можно выбирать YOLOv10-l/x или D-FINE-l, однако это повышает требования к оборудованию. Полученные рекомендации обеспечивают практическую основу для развёртывания муниципальных систем мониторинга инцидентов на дорогах.

Литература

1. *Chen L., Li S., Bai Q. [et al]* Review of image classification algorithms based on convolutional neural networks // *Remote Sensing*. - 2021. - №22.
2. *Remote Sensing Object Detection in the Deep Learning Era—A Review* // *ResearchGate* URL: https://www.researchgate.net/figure/Object-detection-methods-performance-on-COCO-test-dev-dataset-from-2015-to-2023_fig4_377367955 (дата обращения: 05.11.2025).
3. *Yian Zhao, Wenyu Lv, Shangliang Xu [et al.]* DETRs Beat YOLOs on Real-time Object Detection. – URL: <https://arxiv.org/abs/2304.08069> (дата обращения 05.11.2025).
4. *Ultralytics. Baidu's RT-DETR: A Vision Transformer-Based Real-Time Object Detector.* URL: <https://docs.ultralytics.com/models/rtdeatr/> (дата обращения 30.10.2025).
5. *Yansong Peng, Hebei Li, Peixi Wu [et al.]* D-FINE: Redefine Regression Task in DETRs as Fine-grained Distribution Refinement. – URL: <https://arxiv.org/abs/2410.13842> (дата обращения 30.10.2025).

6. *Ao Wang, Hui Chen, Lihao Liu [et al.]* YOLOv10: Real-Time End-to-End Object Detection. – URL: <https://paperswithcode.com/paper/yolov10-real-time-end-to-end-object-detection> (дата обращения 30.10.2025).
7. Ultralytics. YOLOv10: Real-Time End-to-End Object Detection. – URL: <https://docs.ultralytics.com/ru/models/yolov10/> (дата обращения 30.10.2025).
8. Ultralytics. Insights on Model Evaluation and Fine-Tuning. – URL: <https://docs.ultralytics.com/guides/model-evaluation-insights/> (дата обращения 29.10.2025).

ПРОСТРАНСТВЕННО-ВРЕМЕННОЕ ВОССТАНОВЛЕНИЕ МЕТЕОРОЛОГИЧЕСКИХ ДАННЫХ НА ОСНОВЕ МОДИФИЦИРОВАННОЙ АРХИТЕКТУРЫ SAITS С МЕХАНИЗМОМ МЕЖСТАНЦИОННОГО ВНИМАНИЯ

А. Д. Худобин, И. Е. Воронина

Воронежский государственный университет

Аннотация. Рассматривается метод восстановления пропущенных значений в данных метеорологических наблюдений, основанный на расширении архитектуры SAITS (Self-Attention-based Imputation for Time Series). Предложенные модификации включают механизм межстанционного внимания для явного моделирования пространственных зависимостей между станциями и систему ближайших соседей для включения локальной пространственной информации. В отличие от базового SAITS, который разработан для одиночных многомерных временных рядов, модифицированная архитектура разработана для явно структурированных групп станций.

Ключевые слова: метеорология, восстановление пропущенных значений, SAITS, Time Series, Transformer, Self-Attention, Cross-Attention, DMSA, GRIN, RNN.

Введение

Метеорологические наблюдения составляют основу современных систем прогнозирования погоды, климатического моделирования и мониторинга окружающей среды. Но данные метеостанций неизбежно содержат пропуски, которые возникают вследствие технических неисправностей оборудования, неблагоприятных погодных условий, сбоев в системах связи, планового технического обслуживания. Некоторые параметры могут иметь более 50 % пропусков, что сильно снижает ценность данных для последующего анализа и моделирования.

Традиционные статистические методы восстановления пропущенных значений (линейная интерполяция, метод ближайшего соседа, заполнение средними значениями, кригинг, многомерное восстановление данных цепными уравнениями) демонстрируют ограниченную эффективность при работе с метеорологическими данными. Эти методы либо неспособны моделировать сложные временные зависимости, либо вычислительно неэффективны для крупномасштабных сетей с длительными временными рядами. Кроме того, большинство традиционных подходов либо не учитывают пространственную структуру, либо рассматривают пространственные и временные зависимости раздельно, что не отражает природу атмосферных процессов, характеризующихся тесной пространственно-временной связностью.

В последние годы методы глубокого обучения показали заметные результаты в задачах анализа временных рядов. Рекуррентные нейронные сети (RNN) особенно их модификации LSTM и GRU, могут моделировать долгосрочные зависимости во временных данных. Архитектуры, которые основаны на механизме самовнимания, в частности Transformer, эффективно моделируют долгосрочные зависимости за счет способности механизма самовнимания напрямую связывать любые элементы последовательности. Архитектура SAITS представляет современный подход к восстановлению пропущенных данных во временных рядах. SAITS использует двухветвевую структуру с механизмом самовнимания для высокоточного восстановления пропущенных значений на основе временного контекста.

Однако базовая архитектура SAITS создана для обработки одиночных многомерных рядов и не содержит механизмов для эффективного использования пространственной структуры сети метеорологических станций. Метеорологические данные характеризуются сильной пространственной зависимостью: атмосферные явления одновременно охватывают множество

станций, создавая зависимости, которые можно использовать для восстановления данных. Применения базового SAITS к многостанционным данным либо игнорирует эту пространственную информацию при независимой обработке станций, либо требует объединения данных всех станций в один вектор признаков огромной размерности, это практически неосуществимо вычислительно и смешивает пространственные и временные зависимости в едином механизме самовнимания.

1. Обзор литературы

Архитектура SAITS, предложенная в 2023 году, основана на применении двухветвевой структуры трансформера для восстановления данных временных рядов. Первая ветвь использует диагональное маскирование в механизме самовнимания, что заставляет модель извлекать временные паттерны из контекста, исключая возможность тривиального копирования входных значений. Вторая ветвь получает комбинацию реальных наблюдений и предсказаний первой ветви, обеспечивая механизм уточнения результатов. Выходы обеих ветвей объединяются через обучаемую сеть взвешивания, динамически определяющую оптимальное соотношение предсказаний для каждого признака. Диагональное маскирование в механизме самовнимания предотвращает использование информации из той же временной позиции, стимулируя модель к обучению истинных временных зависимостей [1–2].

Исследования в области пространственно-временного моделирования предлагают различные подходы к совместному учету пространственной и временной структуры. Графовые нейронные сети явно моделируют топологию сети через графовое представление, применивая операции агрегации по ребрам графа для распространения информации между узлами. Сверточные рекуррентные архитектуры комбинируют сверточные операции для обработки локальных пространственных зависимостей с рекуррентными механизмами для моделирования временной эволюции. Однако эти подходы либо ограничены необходимостью предварительно заданной пространственной структуры (регулярной решетки или графа), либо имеют ограничения последовательной рекуррентной обработки, включая проблемы с градиентами и невозможность эффективной параллелизации.

Методы пространственно-временного восстановления данных для сенсорных сетей, такие как GRIN, используют графовые нейронные сети для явного моделирования пространственных связей между узлами сети. Однако эти методы основаны на рекуррентных архитектурах, что ограничивает их применимость к длинным временным последовательностям, характерным для метеорологических наблюдений. Существующие подходы либо полагаются на фиксированную предопределенную структуру пространственных связей, либо пытаются обучить эту структуру из данных, игнорируя доступную географическую информацию о расположении станций [3].

Ключевой нерешенной проблемой остается эффективное комбинирование преимуществ трансформерных архитектур для временного моделирования с явным учетом пространственной структуры сети станций при сохранении вычислительной эффективности и физической интерпретируемости результатов.

2. Постановка задачи

Рассмотрим сеть из N метеорологических станций с известными географическими координатами. Каждая станция имеет F параметров с равномерным временным шагом на протяжении T временных отсчетов. Данные представлены трехмерным тензором наблюдений и соответствующей булевой маской, указывающей наличие или отсутствие измерения в каждой пространственно-временной точке. Задача заключается в восстановлении отсутствующих

значений для всех позиций с нулевой маской так, чтобы восстановленные данные минимизировали ошибку на известных наблюдениях и сохраняли физическую правдоподобность, включая пространственную связь между соседними станциями и временную гладкость.

Метеорологические данные характеризуются разнообразной пространственной структурой корреляций. На локальном масштабе в десятки километров наблюдается высокая корреляция, на региональном масштабе в сотни километров корреляция остается умеренной, а на дальних расстояниях (больше пятисот километров) корреляция становится слабой, потому что станции попадают под влияние различных синоптических режимов. Эта разнообразная пространственная структура должна учитываться при проектировании системы восстановления данных.

3. Реализация

3.1. Стратегия региональных моделей

Для обеспечения вычислительной эффективности и учета разнообразной пространственной структуры корреляций сеть станций разбивается на географически компактные региональные кластеры. Разбиение выполняется через пространственную сортировку станций методом построения цепочки ближайших соседей с последующим разделением упорядоченного списка на группы с ограничением максимального количества станций в них. Алгоритм сортировки начинается с выбора начальной точки и итеративно добавляет ближайшую непосещенную станцию к текущему концу цепочки. Это обеспечивает формирование последовательности станций с минимальными пространственными скачками между соседними элементами. Кроме ограничения на максимальное количество станций в кластере есть ограничение на максимальное общее внутрикластерное расстояние.

Стратегия региональных моделей является решением, которое обеспечивает практическую применимость подхода к крупномасштабным сетям. Ключевые методологические инновации, описываемые далее, относятся к архитектуре модели, применяемой внутри каждого регионального кластера.

3.2. Механизм межстанционного внимания

Центральной инновацией предложенной архитектуры является введение механизма межстанционного внимания, работающего ортогонально временному самовниманию базовой архитектуры SAITS. В отличие от базовой SAITS, обрабатывающей данные в формате одиночной многомерной временной последовательности признаков, модифицированная архитектура работает с явно структурированными группами станций. Данные организованы в виде четырехмерного тензора, где измерения соответствуют обучающим примерам, времени, станциям и признакам.

Для каждой целевой станции, требующей восстановления данных, формируется группа, включающая саму станцию и несколько ближайших соседей из этого кластера. После преобразования входных данных в пространство скрытых представлений модели к ним применяется последовательность блоков межстанционного внимания.

Архитектура блока следует современным практикам проектирования трансформеров с предварительной нормализацией и остаточными связями [1]. После операции внимания между станциями идет позиционная нейронная сеть прямого распространения с промежуточным расширением размерности и нелинейной активацией. Обе операции окружены остаточными связями и нормализацией, обеспечивая стабильность обучения глубоких сетей. Маска заполнения станций гарантирует корректную обработку случаев, когда реальное количество соседей меньше максимального размера группы.

Физическая интерпретация весов межстанционного внимания: они отражают степень влияния одной станции на другую в данный момент времени для всей совокупности параметров.

3.3. Система географических соседей

Параллельно с механизмом межстанционного внимания в архитектуру включена система явного учета географических соседей. Для каждой целевой станции определяется фиксированное множество ближайших соседей с максимальным фиксированным расстоянием до целевой станции. Количество соседей выбирается достаточно малым для вычислительной эффективности, но достаточно большим для корректного представления локального пространственно-го контекста.

Информация о системе соседей передаётся в модель несколькими способами:

1. Добавление данных соседних станций к входному представлению явным образом. Позволяет модели непосредственно наблюдать текущие значения параметров на окружающих станциях.

2. Вычисление нормализованных весов обратного расстояния для каждого соседа. Используются для формирования пространственного кодирования.

3. Преобразование этих весов и расстояний в дополнительные скалярные признаки, добавляемые к входному представлению каждой станции.

Ключевое различие между системой соседей и механизмом межстанционного внимания заключается в природе связей. Система соседей определяет жесткую структуру, в которой географически близкие станции должны иметь схожие метеорологические условия. Механизм внимания обучает мягкие адаптивные веса на основе паттернов данных, которые могут отклоняться от простых правил расстояния в зависимости от конкретных метеорологических условий. Синергия жестких географических ограничений и обучаемых весов обеспечивает быструю сходимость и физическую правдоподобность результатов.

Заключение

Система пространственно-временного восстановления метеорологических данных существенно расширяет возможности базовой архитектуры SAITS за счет введения явного моделирования пространственной структуры сети станций. Ключевые инновации включают механизм межстанционного внимания для обучения адаптивных пространственных зависимостей между станциями, систему географических ближайших соседей для включения физически обоснованных пространственных предположений и стратегию региональных моделей для обеспечения вычислительной эффективности и климатической специализации при работе с крупномасштабными сетями.

Фундаментальное преимущество предложенного подхода заключается в ортогональном разделении пространственного и временного моделирования. Механизм межстанционного внимания работает по пространственному измерению на каждом временном шаге, обучая веса взаимного влияния станций друг на друга. Механизм временного самовнимания из базовой архитектуры SAITS фокусируется на временной эволюции, не смешиваясь с пространственными паттернами.

Синергия жестких географических ограничений через систему ближайших соседей и обучаемых мягких весов через механизм внимания критически важна. Географические соседи предоставляют предположение о локальности корреляций, ограничивающее пространство поиска и предотвращающее переобучение. Механизм внимания обучает оптимальные веса внутри этого пространства, адаптируясь к специфике паттернов данных и метеорологических условий. Экспериментальные результаты демонстрируют, что исключение любого из этих

компонентов приводит к существенному ухудшению качества, это подтверждает их взаимодополняемость.

Стратегия разбиения на региональные модели решает проблему масштабируемости к крупным сетям и улучшает результаты через специализацию моделей под климатические особенности различных регионов.

Предложенная система готова к практическому применению в операционных системах контроля качества метеорологических данных, предобработке для численных моделей прогноза погоды и климатических исследованиях. Модульная архитектура обеспечивает высокую расширяемость. Поэтому модель может адаптироваться к другим типам пространственно-временных данных сенсорных сетей. Перспективные направления развития включают иерархическое межклusterное внимание для граничных станций и улучшение автоматического разбиения на кластеры.

Литература

1. *Du W. SAITS: Self-Attention-Based Imputation for Time Series* / W. Du, D. Côté, Y. Liu // Expert Systems with Applications. – 2023. – 22 с.
2. *Tashiro Y. CSDI Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation* / Y. Tashiro [и др.] // Advances in Neural Information Processing Systems. – 2021. – 30 с.
3. *Cini A. Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks* / A. Cini, I. Marasca, C. Alippi // International Conference on Learning Representations. – 2022. – 20 с.

ПОДХОДЫ К ИНТЕРПРЕТАЦИИ РЕЗУЛЬТАТОВ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

А. А. Худяков¹, А. А. Арзамасцев^{1, 2}

¹Воронежский государственный университет

²Тамбовский филиал Межотраслевого научно-технического комплекса «Микрохирургия глаза»
имени академика С. Н. Федорова

Аннотация. Данная статья представляет собой обзор существующих подходов и инструментов для интерпретации процесса принятия решений в нейросетевых моделях. Обсуждается проблема «черного ящика» и важность интерпретации результатов работы моделей в медицине и других областях, где от принятого решения зависит здоровье и жизнь человека. Рассматриваются основные подходы к повышению интерпретируемости сверточных нейронных сетей — Grad-CAM и его вариации, модели со встроенной интерпретируемостью и на основе прототипов, гибридные методы.

Ключевые слова: искусственный интеллект, нейронные сети, сверточные нейронные сети, компьютерное зрение, офтальмология.

Введение

Офтальмология является одной из областей медицины, где методы глубокого обучения получили особенно широкое распространение. Сверточные нейронные сети (СНС), а также модели на основе трансформеров, при наличии достаточного объема данных для обучения, могут использоваться для решения задач классификации и постановки диагноза по изображениям с различных медицинских приборов, достигая высокого показателя точности и упрощая работу врачей-офтальмологов.

Однако, несмотря на высокое качество результатов, достаточное для возможности применять такие модели на практике, большинство нейросетевых моделей представляют собой «черные ящики», ценность которых заключается лишь в конечных предсказаниях без возможности детальнее изучить процесс принятия решения внутри модели и сделать выводы о том, какие фрагменты исходных данных внесли в данное решение наибольший вклад. Если для других сфер применения алгоритмов машинного обучения эта особенность не столь существенна, то для медицины она наиболее критична — врачу и пациенту важно понимать, почему модель приняла то или иное решение и можно ли ему доверяться, особенно когда речь идёт о жизни человека, необходимости назначить опасное или дорогостоящее лечение или в условиях ограниченного времени.

Стоит отметить, что некоторые алгоритмы, такие как линейная регрессия или деревья принятия решений, изначально являются хорошо интерпретируемыми, однако, они не представляют для нас особого интереса, так как либо не могут быть использованы для анализа изображений, либо показывают неудовлетворительные результаты.

Объяснимый искусственный интеллект (ХАІ) — это область изучения и создания методов, инструментов и подходов, которые стремятся сделать современные модели машинного обучения более интерпретируемыми и понятными для людей, что в свою очередь повышает степень доверия к результатам данных моделей и возможности их практического применения. Такие инструменты позволяют визуализировать внутренние механизмы модели, анализировать вклад отдельных признаков входных данных в итоговое решение, чем также помогают выявлять и устранять ошибки и неточности в реализации алгоритмов и улучшать их. В данной статье рассмотрены некоторые из современных подходов повышения интерпретируемо-

сти нейросетевых моделей, которые могут использоваться в задачах классификации офтальмологических изображений.

1. Grad-CAM

Grad-CAM [1] (Gradient-weighted Class Activation Mapping) — один из наиболее широко используемых методов интерпретации СНС в компьютерном зрении, позволяющий локализовать конкретные области изображения, которые внесли наибольший вклад в итоговое решение модели.

Сверточные слои в СНС сохраняют пространственную структуру изображения, а это значит, что градиенты последних полносвязных слоев модели относительно последних сверточных слоев показывают не просто важность абстрактных признаков, которые модель выделила в исходных данных, а важность конкретных фрагментов изображения, что можно использовать для создания тепловой карты.

На рис. 1 показана визуализация [2] работы метода Grad-CAM на примере изображений сетчатки для автоматического выявления и классификации различной степени диабетической ретинопатии. Используя градиенты относительно каналов последнего сверточного слоя, была создана тепловая карта, которая затем интерполируется и накладывается на исходное изображение.

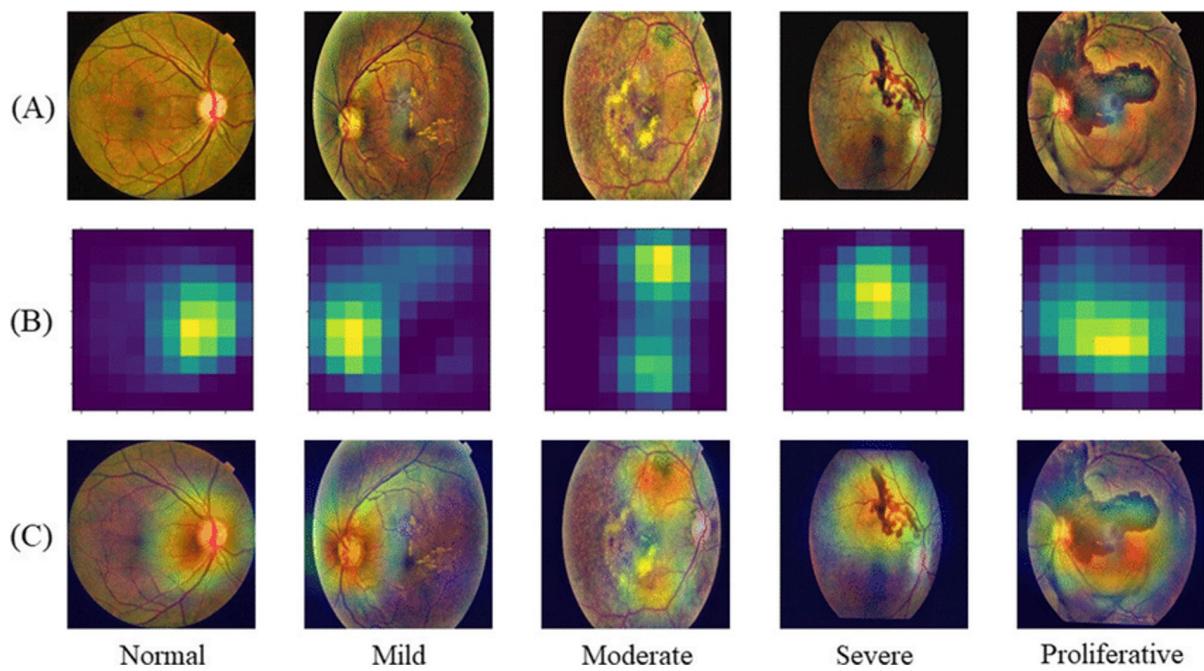


Рис. 1. Работа метода Grad-CAM на примере изображений сетчатки

Основным преимуществом метода является его простая интеграция в уже существующие и обученные СНС — он не требует изменения архитектуры или переобучения модели.

Из недостатков данного метода можно выделить зависимость от размерности карты признаков в последних сверточных слоях, что непосредственно влияет на размер и пространственную точность полученной тепловой карты, а также проблему однозначного определения на что именно модель обратила своё внимание в случае наличия в подсвеченном регионе большого количества мелких деталей. Различные вариации и улучшения метода, такие как Guided Grad-CAM [3], позволяют получить более детализированные тепловые карты, а подход SmoothGrad [4] позволяет уменьшить шум за счет усреднения сразу нескольких карт, построенных на специально зашумленных версиях входных данных.

2. Модели со встроенной интерпретируемостью

Данная группа методов повышения интерпретации модели предполагает изначальное построение архитектуры с целью получения высокой объяснимости результатов. Уже упомянутые ранее деревья принятия решений и линейная регрессия являются хорошими примерами, однако их выразительность и возможность решать сложные задачи, особенно задачи компьютерного зрения, сильно ограничена.

ProtoPNet [5] (Prototype Network for Interpretable Image Classification) — яркий представитель семейства моделей, изначально построенных с целью повышения интерпретируемости результатов. Классические СНС состоят из нескольких сверточных блоков в начале, выделяющих из изображения определенные признаки, и полно связных слоев в конце. Однако определить, что означает и за что отвечает каждый выделенный признак, в классических СНС достаточно сложно. ProtoPNet добавляет еще один тип слоя — слой прототипов. Каждый прототип — вектор в пространстве признаков, выделенных ранее в сверточных слоях, где каждому признаку ставится в соответствие некоторый коэффициент. После обучения модели каждому прототипу ставится в соответствие ближайший к нему по метрике L2 фрагмент одного из изображений в обучающем наборе. Тогда можно сказать, что каждый прототип — это не просто набор чисел, а буквально область в одном из изображений обучающего набора. При принятии решения модель проверяет поступающее на вход изображение на соответствие каждому прототипу, и такое решение можно довольно легко интерпретировать — модель сделала определенный выбор, так как нашла в исходном изображении фрагменты, имеющие визуальное сходство и пространственную локальность с соответствующими фрагментами из изображений обучающего набора, для которых было сделано такое же решение, например — поставлен такой же диагноз, так как определенная область содержит некое затемнение, и похожие затемнения встречались в изображениях обучающего набора с таким же диагнозом.

Принцип работы ProtoPNet для диагностики легочной пневмонии [6] показан на рис. 2.

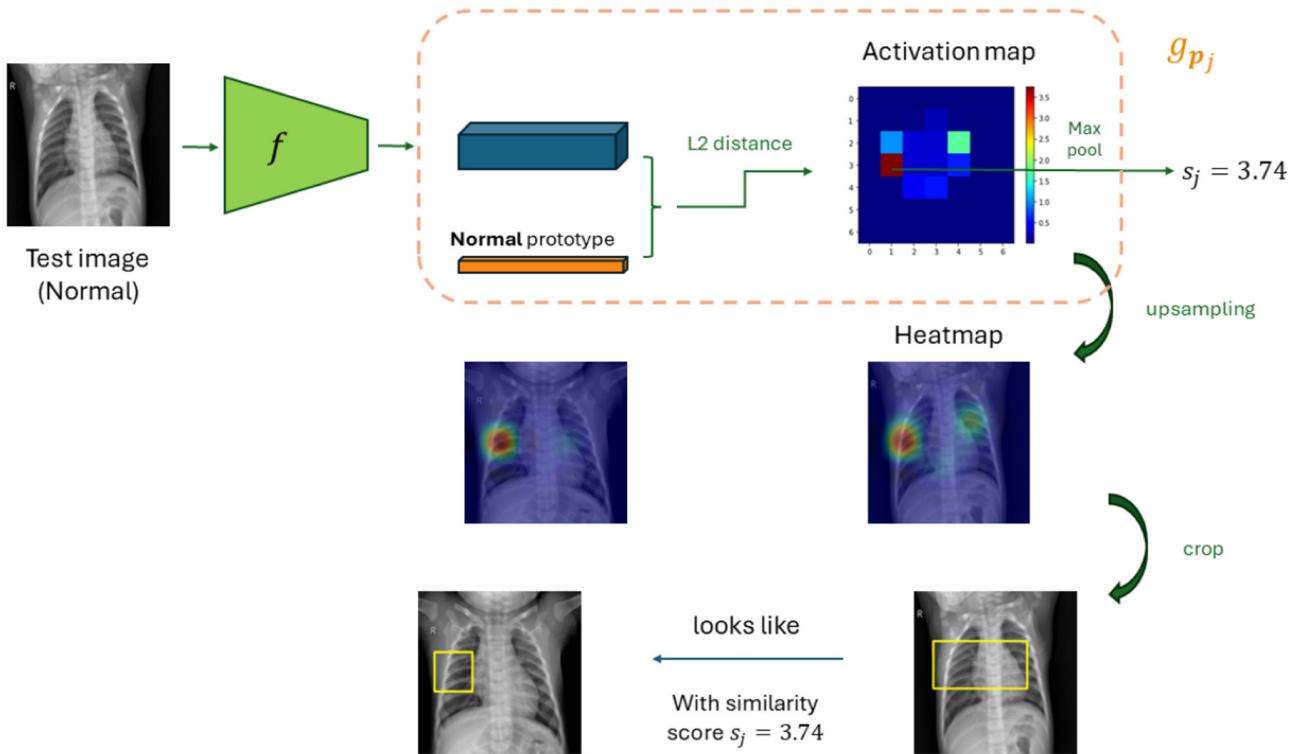


Рис. 2. Работа метода ProtoPNet на примере изображений грудной клетки

3. Гибридные методы

Использование специальных архитектур для улучшения интерпретируемости модели эффективно, но накладывает ограничения на возможности развития и изменения моделей для повышения точности, что обычно предпочтительнее — если хорошо интерпретируемая модель намного больше ошибается по сравнению более точным «черным ящиком», в её использовании нет никакого практического смысла.

Гибридные методы предполагают совместное использование двух типов моделей одновременно — высокоточной классической СНС (или, например, модели на основе визуальных трансформеров (ViT)) и простой, но интерпретируемой и объяснимой модели, обученной аппроксимировать поведение более сложной. Таким образом, вторая модель не заменяет первую, а служит инструментом для анализа её решений.

Примером такого метода может служить LIME [7] (Local Interpretable Model-Agnostic Explanations). Для каждого поступающего на вход основной сети изображения создается несколько модификаций, которые также подаются на вход основной сети. По полученным ответам строится более простая модель (линейная регрессия или дерево принятия решений), которая позволяет понять, какие именно фрагменты (суперпиксели) в исходном изображении повлияли на полученный результат.

Заключение

Интерпретация результатов работы искусственных нейронных сетей — важная задача, которая позволяет повысить доверие человека к сделанным алгоритмами выводам. Она особенно важна в таких областях, как медицина, где от принятого решения может зависеть судьба человеческой жизни. Помимо постановки диагноза, интерпретация может также использоваться для обучения персонала и автоматического создания и разметки сопутствующих наборов данных для обучения других алгоритмов машинного обучения. Современные подходы к повышению объяснимости нейронных сетей позволяют расширить сферу их применения и упростить внедрение моделей в нашу повседневную жизнь.

Благодарности

Работа выполнена в соответствии с договором о научно-техническом сотрудничестве Воронежского государственного университета и Федерального государственного автономного учреждения «Национальный медицинский исследовательский центр «Межотраслевой научно-технический комплекс «Микрохирургия глаза» имени академика С.Н. Федорова», Тамбовского филиала от 28.11.2022.

Литература

1. *Selvaraju R. R. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization* / R. R. Selvaraju, M. Cogswell, A. Das [и др.]. – URL: <https://arxiv.org/abs/1610.02391> (дата обращения 01.11.2025).
2. *Hinge attention network: A joint model for diabetic retinopathy severity grading* / N. S. Shaik, T. K. Cherukuri // *Applied Intelligence*. – 2022. – №52. – С. 15105–15121.
3. *Selvaraju R. R. Grad-CAM: Why did you say that?* / R. R. Selvaraju, A. Das, R. Vedantam [и др.]. – URL: <https://arxiv.org/abs/1611.07450> (дата обращения 01.11.2025).
4. *Smilkov D. SmoothGrad: removing noise by adding noise* / D. Smilkov, N. Thorat, B. Kim [и др.]. – URL: <https://arxiv.org/abs/1706.03825> (дата обращения 01.11.2025).

5. *Chen C. This Looks Like That: Deep Learning for Interpretable Image Recognition / C. Chen, O. Li, C. Tao [и др.]. – URL: <https://arxiv.org/abs/1806.10574> (дата обращения 02.11.2025).*
6. *De Santi L. Part-Prototype Models in Medical Imaging: Applications and Current Challenges / L. De Santi, F. Piparo, F. Bargagna [и др.]. – URL: <https://www.mdpi.com/2673-7426/4/4/115> (дата обращения 02.11.2025).*
7. *Ribeiro M. T. «Why Should I Trust You?»: Explaining the Predictions of Any Classifier / M. T. Ribeiro, S. Singh, C. Gustrin. – URL: <https://arxiv.org/abs/1602.04938> (дата обращения 03.11.2025).*

ПРИМЕНЕНИЕ НЕЧЕТКИХ КОГНИТИВНЫХ КАРТ В ОЦЕНКЕ НАДЕЖНОСТИ СИСТЕМЫ ИНТЕЛЛЕКТУАЛЬНОГО ВИДЕОМОНИТОРИНГА

Е. Г. Царькова

МИРЭА – Российский технологический университет

Аннотация. Целью исследования является рассмотрение вопросов применимости нечетких когнитивных карт (НКК) для анализа надежности сложных человеко-машинных систем (ЧМС), а также разработка алгоритмов оценки факторов влияния на показатель надежности системы с учетом экспертных оценок. На примере анализа надежности системы интеллектуального видеомониторинга охраняемого объекта демонстрируется построение нечеткой когнитивной карты, приводится метод расчета индексов важности и коэффициентов совместного влияния факторов для определения интегрального показателя надежности системы. Разработан вычислительный алгоритм, приведены результаты его программной реализации. Выделены факторы, оказывающие наибольшее влияние на целевую переменную, рассчитаны и ранжированы факторы совместного влияния.

Ключевые слова: человеко-машинные системы, надежность, нечеткая логика, нечеткие когнитивные карты, экспертные оценки, факторы влияния, видеомониторинг.

Введение

При исследовании эффективности систем обеспечения комплексной безопасности, рассматриваемых как сложные человеко-машинные системы, возникает объективная потребность в выработке подходов к их моделированию с учетом факторов, не поддающихся прямому измерению, включающих не только технические характеристики надежности системы, но и человеческий фактор (состояние человека-оператора, уровень компетенций, степень утомления, скорость реакции). Для оценки надежности подобных систем, состояние которых зависит от большого количества взаимозависимых параметров, применение классических подходов, например, регрессионного анализа или традиционных систем нечеткого вывода, становятся малоэффективными, в том числе, в связи с наличием зависимостей между факторами влияния, отсутствием возможности количественной оценки показателей и др.

Альтернативным вариантом в оценке надежности сложных человеко-машинных систем (ЧМС), можем считать применение нечетких когнитивных карт (НКК). Теория НКК получила свое развитие в работах Р. Аксельрода как продолжение идей Б. Коско [1, 2]. Эффективность НКК при исследовании ЧМС во многом определяется возможностью моделировать динамику взаимного влияния факторов в случае использования качественных оценок, выявлять факторы, оказывающие наиболее сильное влияние на целевой показатель системы, осуществлять сценарное моделирование для принятия решений с учетом выявленных управляющих воздействий, за счет которых возможно наибольшее положительное воздействие на систему. В работе рассматривается применение нечетких когнитивных карт для ранжирования силы влияния множества факторов на надежность системы интеллектуального мониторинга охраняемого объекта, рассматриваемой как человеко-машинная система. Целью исследования является применение когнитивного анализа для принятия решений по выбору наиболее перспективных направлений повышения эффективности системы видеомониторинга охраняемого объекта.

1. Метод решения

1.1. Подход к построению нечеткой когнитивной карты

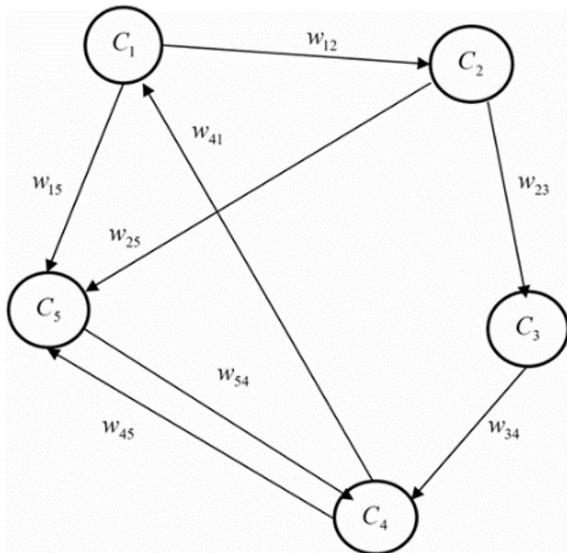


Рис. 1. Вид нечеткой когнитивной карты

Применение НКК основывается на построении ориентированного взвешенного графа, где вершинами являются концепты C_i . Концепты служат ключевыми переменными модели. Взвешенные дуги графа $w_{ij} \in [-1;1]$ отражают силу и направление причинно-следственных связей между концептами. При этом вес дуги w_{ij} характеризует интенсивность воздействия изменения концепта C_i на динамику концепта C_j . Общий вид НКК приведен на рис. 1.

При использовании НКК в сложных ЧМС источником информации о весах графа становятся экспертные оценки, выраженные в лингвистической форме: «повышает-понижает», «усиливает-ослабляет», «растет-снижается» и т. д. с выбором соответствующего знака связи (положительный — отрицательный) [3].

Для описания нечеткой когнитивной карты ис-

пользуется граф $G = (C, E)$, где $C = \{C_1, C_2, \dots, C_k\}$ — множество узлов (вершин) графа (концепты). Множество всех возможных значений, которые может принимать степень влияния концепта, обозначим через L , множество ребер графа — через E . Отображение $W : (C_i, C_j) \rightarrow w_{ij}$ ставит в соответствие паре концептов (C_i, C_j) ненулевое значение веса направленного ребра w_{ij} при $i \neq j$ и 0 при $i = j$. Таким образом, отображение $W(C \cdot C)$ представляется матрицей связи, имеющей нули на основной диагонали.

1.2. Нечеткая когнитивная карта системы видеомониторинга

Рассмотрим пример построения НКК для оценки надежности ЧМС «оператор — система видеомониторинга — охраняемый объект». В качестве концептов выделим следующие факторы, влияющие на общую эффективность системы:

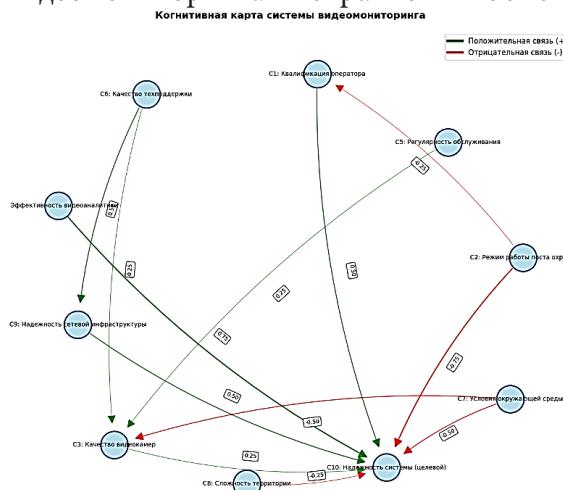


Рис. 2. Нечеткая когнитивная карта системы видеомониторинга

квалификация и внимательность оператора (C_1), режим и интенсивность работы поста охраны (C_2), разрешающая способность и качество видеокамер (C_3), эффективность алгоритмов видеоаналитики (C_4), регулярность профилактического обслуживания оборудования (C_5), качество и скорость технической поддержки (C_6), условия окружающей среды (C_7), сложность и уязвимость охраняемой территории (C_8), надежность и пропускная способность сетевой инфраструктуры (C_9), общая надежность и эффективность системы видеомониторинга (целевой концепт, C_{10}). На рис. 2 приведен вид нечеткой когнитивной карты, построенной с использованием введенных концептов и модельных

экспертных оценок весов дуг ориентированного графа. Для определения w_{ij} введена шкала оценок, позволяющая оценить количественно связи факторов влияния с учетом лингвистических термов (табл. 1).

Таблица 1

Оценки силы влияния концептов

Лингвистической выражение оценки	Количественное значение
Максимальное положительное влияние фактора	1
Положительное влияние фактора, выше среднего значения	0,75
Положительное среднее влияние фактора	0,5
Положительное влияние фактора, ниже среднего значения	0,25
Влияние отсутствует	0
Отрицательное влияние фактора, ниже среднего значения	-0,25
Отрицательное среднее влияние фактора	-0,5
Отрицательное влияние фактора, выше среднего значения	-0,75
Максимальное отрицательное влияние фактора	-1

В случае необходимости учета мнения нескольких экспертов групповая оценка величины w_{ij} может вычисляться через взвешенное среднее:

$$w_{ij} = \frac{\alpha_1 w_{ij}^1 + \alpha_2 w_{ij}^2 + \dots + \alpha_m w_{ij}^m}{\alpha_1 + \alpha_2 + \dots + \alpha_m},$$

где w_{ij}^p — оценка силы влияния концептов p -м экспертом; α_p — весовой коэффициент, отражающий вес мнения p -го эксперта, $p = 1, 2, \dots, m$, m — количество экспертов [4].

В рассматриваемом случае матрица связей W имеет вид:

$$W = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 \\ -0,25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,75 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,25 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0,75 \\ 0 & 0 & 0,25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,25 & 0 & 0 & 0 & 0 & 0,5 & 0 \\ 0 & 0 & -0,5 & 0 & 0 & 0 & 0 & 0 & -0,5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,25 \\ 0 & 0,25 & 0 & 0 & 0 & 0 & 0 & 0 & 0,5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Для вычисления значений концептов в динамике введем соотношение:

$$A_i^{k+1} = f \left(\sum_{\substack{j=1 \\ j \neq i}}^k A_j^k w_{ij} + c A_i^k \right), \quad k = 0, 1, 2, \dots, \quad (1)$$

где A_i^{k+1} — величина концепта C_i на шаге $k+1$, A_i^k, A_j^k — величины C_i, C_j на шаге k , соответственно, w_{ij} — величина влияния C_i на C_j , $c \in [0, 1]$ — вклад значения концепта на предыдущем шаге (предыстория), f — пороговая функция для нормировки значений в единичный отрезок. Под величиной концепта понимаем степень его выраженности на шаге k . В качестве пороговой функции выберем правую часть гиперболического тангенса:

$$f(x) = \begin{cases} \tanh(x), & x \geq 0 \\ 0, & x < 0 \end{cases}, \quad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (2)$$

1.3. Расчет индексов влияния

Пусть A^k, A^{k+1} — векторы значений концептов системы на шагах $k, k+1$, соответственно, W_0 — квадратная матрица значений взаимного влияния концептов нечеткой когнитивной карты. Тогда можем представить (1) в матричном виде [5]:

$$A_i^{k+1} = f(A^k W_0 + c A^k), \quad k = 0, 1, 2, \dots, \quad (3)$$

где

$$W_0 = \begin{pmatrix} 0 & w_{12} & \dots & w_{1n} \\ w_{21} & 0 & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & 0 \end{pmatrix}. \quad (4)$$

Заменяя значения главной диагонали матрицы (4) на коэффициент c :

$$W = \begin{pmatrix} c & w_{12} & \dots & w_{1n} \\ w_{21} & c & \dots & w_{2n} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & c \end{pmatrix}, \quad (5)$$

приходим от преобразования (4) к следующему виду:

$$A_i^{k+1} = f(A^k W), \quad k = 0, 1, 2, \dots \quad (6)$$

Для учета значений концептов на нулевом шаге вводится вектор начальных значений концептов $A^0 = [A_1^0, A_2^0, \dots, A_n^0]$. Вычисление целевого показателя эффективности системы сводится к нахождению значения выходного концепта C_n , равного $A_n \in [0; 1]$. Величина значения концепта C_n на первом шаге является функцией от элементов вектора $A^0 = [A_1^0, A_2^0, \dots, A_n^0]$, т. е. $A_n^l = F(A_1^0, A_2^0, \dots, A_n^0)$. В качестве A_n^l принимаем значения C_n в стационарном режиме (такой шаг l , на котором A_n^l близко к A_n^{l-1}).

Пусть $F(1_j, 0)$ — значение функции из (6) при $A_j^0 = 1$ и равенстве нулю всех остальных аргументов, $F(0)$ — значение функции из (6) при равенстве нулю всех аргументов, $F(1_j, 1_k, 0)$ при $A_j = A_k = 1$ и равенстве нулю всех остальных аргументов.

Для определения важности концепта C_j и совместной важности концептов C_j, C_k вводятся, соответственно, величины $I(C_j)$, $I(C_j, C_k)$, для вычисления которых согласно [6, 7] могут быть использованы соотношения:

$$\begin{aligned} I(C_j) &= \frac{\partial A_n^l}{\partial A_j} = \frac{F(1_j, 0) - F(0)}{(1-0)} = F(1_j, 0), \\ I(C_j, C_k) &= \frac{\partial^2 A_n^l}{\partial A_j \partial A_k} = \frac{F(1_j, 1_k, 0) - F(1_j, 0) - F(0, 1_k) + F(0, 0)}{(1-0)(1-0)} = F(1_j, 1_k, 0). \end{aligned} \quad (7)$$

Алгоритм вычисления интегрального целевого показателя на основе нечеткой когнитивной карты сводится к выполнению последовательности шагов:

1. Задание начального вектора A^0 .

Для индекса важности $I(C_j)$:

$$A^0 = [A_j^0 = 1, A_i^0 = 0, i = 1, 2, \dots, n, i \neq j], \quad (8)$$

для $I(C_j, C_k)$:

$$A^0 = \left[A_j^0 = A_k^0 = 1, A_i^0 = 0, i = 1, 2, \dots, n, i \neq j \right]. \quad (9)$$

2. Применяя соотношения (7), находим вектор состояния системы

$$A^l = \left[A_1^l, A_2^l, \dots, A_n^l \right], \quad (10)$$

такое, что $|A_i^l - A_i^{l-1}| < \varepsilon$, где $\varepsilon > 0$ — мало, $i = 1, 2, \dots, n$ (установившийся режим).

В качестве значений индекса важности $I(C_j)$ принимаем элементы A_n^l из (10), полученные с использованием (8), для определения индекса важности $I(C_j, C_k)$ используем элементы A_n^l из (10), полученные с использованием (9).

2. Результаты программной реализации

Для автоматизации расчетов индексов влияния факторов и сценарного моделирования согласно приведенному алгоритму разработано программное средство на языке программирования Python. Инструменты визуализации, реализованные в программном средстве, обеспечивают возможность выявления факторов, оказывающих наибольшее (наименьшее) влияние на общую надежность рассматриваемой системы. С использованием приведенного алгоритма вычислены индексы влияния факторов на целевой концепт системы (рис. 3).

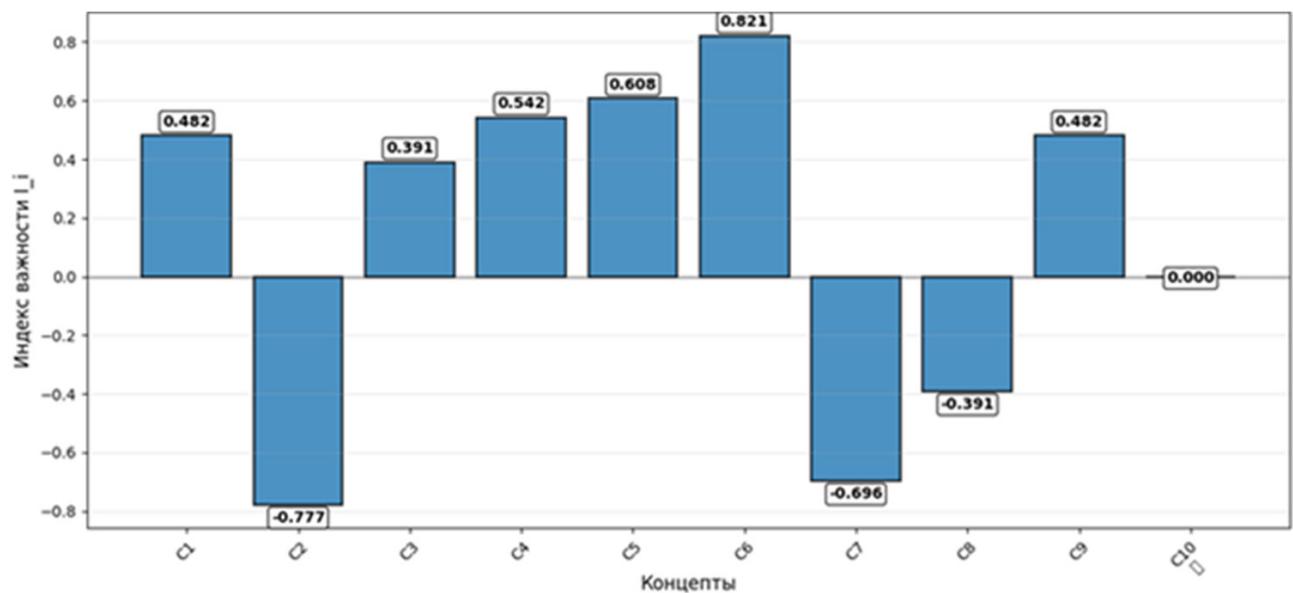


Рис. 3. Результат ранжирования факторов влияния

Согласно полученным результатам наибольшее влияние на эффективность системы видеомониторинга оказывают факторы C_2 (режим и интенсивность работы поста охраны, особенности графика работы, загруженность оператора, количество одновременно обрабатываемых видеопотоков), C_6 (качество и скорость технического обслуживания оборудования), C_7 (эффективность алгоритмов видеоаналитики, качество программного обеспечения, его эффективность при автоматической детекции событий), C_6 (негативные условия окружающей среды, плохая видимость).

На рис. 4 приведены результаты расчета факторов совместного влияния концептов на эффективность системы. Согласно полученным результатам наибольший синергетический эффект дают одновременное воздействие на систему факторов C_2 и C_7 , C_4 и C_6 , C_1 и C_6 .

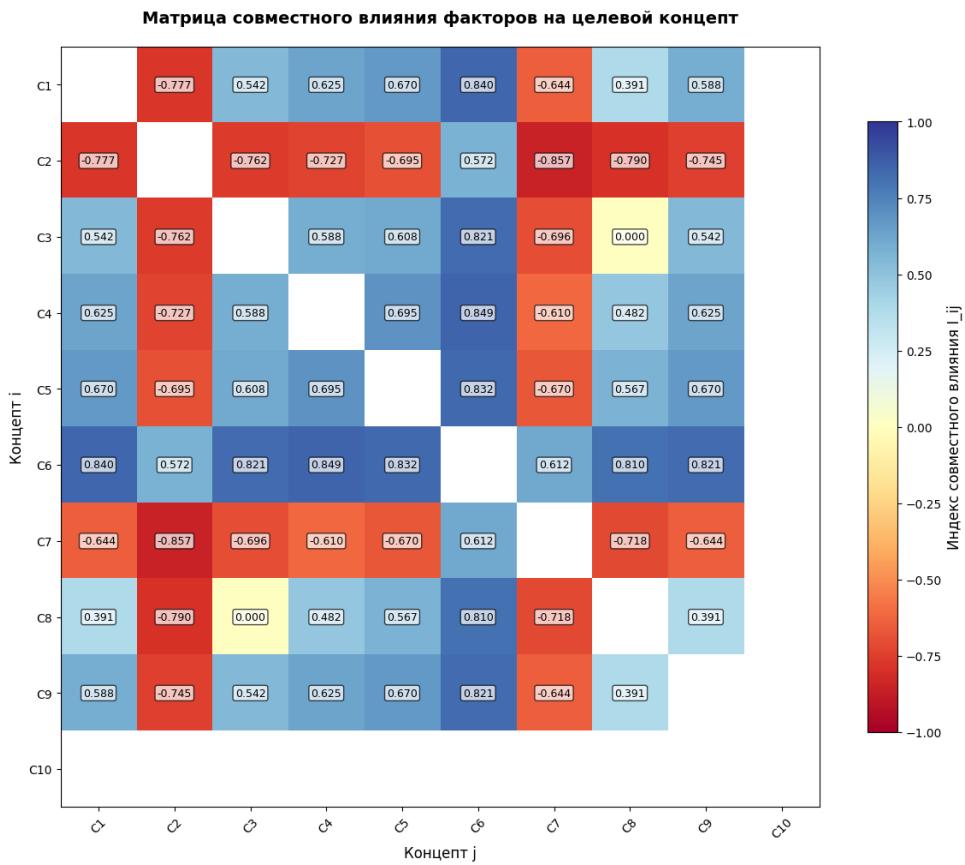


Рис. 4. Результат ранжирования факторов совместного влияния

Заключение

В ходе проведенного исследования показано, что предложенный метод анализа ЧМС на основе нечетких когнитивных карт может быть применен для определения «слабых звеньев» системы, определения наиболее актуальных направлений модернизации процессов и средств видеомониторинга на охраняемой территории. Разработан вычислительный алгоритм ранжирования факторов влияния на общую надежность системы. Разработанный алгоритм позволяет определять направления повышения качества системы видеомониторинга. Так, согласно полученным результатам, можно сделать вывод о необходимости корректировки факторов организационного характера (режим охраны, необходимость непрерывного обучения, повышения квалификации персонала), совершенствование инженерно-технического обеспечения охраняемого объекта, инвестиции в качественное программное обеспечение. Согласно полученным результатам предложенный метод к анализу надежности ЧМС может быть использован в качестве эффективного инструмента исследования сложных слабоструктурированных систем, характеристики которых определяются совокупностью взаимозависимых факторов, оцениваемых преимущественно экспертными процедурами.

Литература

1. Yorkulov B. A., Sulyukova L. F. Quality assessment models based on fuzzy cognitive maps for educational information system // Проблемы вычислительной и прикладной математики. – 2024. – № 4(58). – Р. 148–157.
2. Гончарова А. А., Храмов В. Ю. Оценка риска информационной безопасности систем обработки информации с использованием нечетких продукционных когнитивных карт // Труды

молодых ученых факультета компьютерных наук ВГУ: Сборник научных трудов. – Воронеж : Воронежский государственный университет, 2025. – С. 53-60.

3. *Podvesovskii A. G., Isaev R. A., Kopeliovich I. A.* An Approach to Generating Formal Fuzzy Cognitive Maps for Experimental Studies in Cognitive Modeling // Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications. – 2024. – Vol. 34, No 3. – P. 665–672.

4. *Петухова А. В.* Решение обратной задачи моделирования для предприятия розничной торговли с использованием теории нечетких когнитивных карт // Инженерный вестник Дона. – 2023. – № 3(99). – С. 135–146.

5. *Gutiérrez Buitrago, Aguilar J., Ortega A., Montoya E.* Using fuzzy cognitive maps to evaluate the innovation in micro, small and medium-sized enterprises // Management Decision. – 2024.

6. *Nápoles G., Grau I., Jastrzebska A., Salgueiro Ya.* Learning-based aggregation of Quasi-Nonlinear Fuzzy Cognitive Maps / // Neurocomputing. – 2025. – Vol. 626. – P. 129611.

7. *Kamal Kumar Gola* Security analysis of fog computing environment for ensuring the security and privacy of information // Transactions on Emerging Telecommunications Technologies. – 2023. – Vol. 34, Iss. 10. – P. 112–117.

РАЗРАБОТКА ЧАТ-БОТА КЛИЕНТСКОЙ ПОДДЕРЖКИ ДЛЯ КОММЕРЧЕСКОГО БАНКА НА ОСНОВЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

К. Е. Чекина

Воронежский государственный университет

Аннотация. В статье проведен обзор существующих чат-ботов клиентской поддержки, используемых в российских коммерческих банках, с привлечением отдельных международных примеров. Анализ осуществлен по критериям: актуальность внедрения, функционал, технологическая база, отличительные характеристики, пользовательские оценки и выявленные недостатки. Рассматриваются решения крупнейших банков — Сбербанка, ВТБ, Альфа-Банка и Т-Банка, а также международные — Bank of America, Capital One. Показано, что чат-боты являются одним из ключевых инструментов цифровизации банковской сферы, однако требуют дальнейшего развития в направлении персонализации, расширения функционала и совершенствования диалоговых возможностей.

Ключевые слова: чат-бот, машинное обучение, искусственный интеллект, коммерческие банки, клиентская поддержка, NLP, клиентское обслуживание, автоматизация, бизнес-процессы, языковые модели, цифровизация.

Введение

Современная банковская сфера переживает активный процесс цифровой трансформации. В условиях высокой конкуренции и роста ожиданий клиентов финансовые организации вынуждены внедрять новые технологические решения, способные обеспечить круглосуточную доступность, персонализацию обслуживания и снижение операционных затрат. Одним из таких решений стали чат-боты, которые благодаря использованию методов машинного обучения (ML) и обработки естественного языка (NLP) способны заменить или дополнить традиционные каналы коммуникации — колл-центры и офисные консультации.

Разработка чат-ботов в банковской сфере началась относительно недавно, однако уже сегодня они охватывают значительную часть клиентских запросов. По данным аналитических агентств, чат-боты позволяют автоматизировать до 70–80 % стандартных обращений, что ведёт к сокращению нагрузки на операторов и снижению затрат на обслуживание. При этом сохраняется проблема ограниченности диалога и недостаточной гибкости при работе с нестандартными запросами.

Целью данной статьи является проведение обзорного анализа существующих решений в области чат-ботов коммерческих банков по ключевым параметрам: актуальность, функционал, технологическая основа, отличительные характеристики, пользовательские оценки и выявленные направления для доработки.

1. Актуальность

Актуальность внедрения чат-ботов в коммерческих банках обусловлена рядом факторов. Во-первых, изменилось поведение клиентов: современный пользователь требует быстрого и круглосуточного доступа к информации и услугам, а традиционные контактные центры не способны в полной мере удовлетворить этот запрос. Во-вторых, банки сталкиваются с необходимостью оптимизировать издержки на обслуживание, что становится особенно важным в условиях высокой конкуренции и экономической турбулентности. Автоматизация клиентских обращений позволяет значительно сократить расходы на содержание больших колл-центров.

Кроме того, использование чат-ботов повышает конкурентоспособность банка. В условиях, когда финансовые продукты во многом схожи, качество обслуживания становится ключевым фактором привлечения и удержания клиентов. Возможность решить проблему в несколько кликов через чат-бот создаёт положительный пользовательский опыт и укрепляет лояльность клиентов.

2. Функционал

Современные банковские чат-боты предоставляют широкий спектр функций. Форум «Финополис-2025» подтвердил курс финансового сектора на технологическую независимость и интеграцию искусственного интеллекта во все уровни клиентского сервиса. К таким функциям относятся предоставление информации о состоянии счёта и продуктах банка, возможность совершения простых операций (например, блокировка карты, переводы между собственными счетами, оплат коммунальных или иных регулярных платежей), навигация по услугам банка, оперативное получение справочной информации, а также интеграция с различными каналами доступа — мобильное приложение, мессенджеры, веб-чат. Эти функции сегодня можно считать базовыми для большинства ведущих банковских решений, поскольку они закрывают вос требованные сценарии самообслуживания и уменьшают нагрузку на контакт-центр. При этом важно отметить, что хотя базовые функции у многих чат-ботов схожи, их реализация и глубина отличаются: некоторые решения ограничиваются справочно-информационной ролью, другие поддерживают полноценные транзакции и сами инициируют действия по запросу клиента.

Значимый аспект — применение машинного обучения (ML) и технологий искусственного интеллекта (ИИ) в работе чат-ботов банков. Такие технологии позволяют реализовывать автоматическое распознавание намерения пользователя, анализ истории взаимодействий, прогнозирование потребностей клиента, а также адаптацию ответа бота под конкретного пользователя. Например, в одном из российских банков выяснено, что чат-бот по текстовым диалогам обрабатывает до 46 % обращений с использованием около 150 тематик в рамках 124 сценариев. При этом банки отмечают применение моделей машинного обучения при распределении обращений клиентов по категориям и оптимизации работы контакт-центра.

Машинное обучение также позволяет повысить точность автоматического ответа, улучшать классификацию новых запросов и снижать долю операций, требующих вмешательства живого оператора. Таким образом, ИИ-компонент становится ключевым элементом зрелости чат-бота: от простого «бота-меню» к «умному ассистенту», способному вести контекстный диалог и решать задачи клиента.

В последние годы развивается функционал проактивных сервисов: чат-бот может сам инициировать коммуникацию с клиентом, напоминая о платежах, предупреждая о подозрительных операциях или предлагая персонализированные финансовые решения. Здесь ярко проявляется применение машинного обучения, позволяющего анализировать поведение клиента и формировать прогнозы.

Исходя из этого, при сравнении функционала чат-ботов разных банков можно выделить основные различия и акценты, на которые ориентируются отдельные коммерческие учреждения. Во-первых, глубина транзакционной автоматизации: одни банки оставляют бота преимущественно в справочной роли, другие позволяют завершать операции целиком в чате, без перехода в мобильное приложение или сайт. Во-вторых, каналовая мультиформатность: часть банков делает акцент на текстовом чате, другая — на голосовом интерфейсе или мультимодальных сценариях (распознавание изображений, голосовое управление). В-третьих, уровень персонализации и проактивности: некоторые решения просто отвечают на запросы, тогда как другие предвидят потребности клиента, предлагают рекомендации и автоматически уведомляют об изменениях или рисках. Кроме того, банки по-разному подходят к интеграции с эко-

системой-сервисов: у одних чат-бот ограничен клиентским интерфейсом, у других — часть крупного цифрового окружения с данными, продуктами и сервисами партнёров. Таким образом, различия в функционале отражают стратегические приоритеты банка: безопасность и стабильность, скорость непрерывного обслуживания, эмоциональная вовлечённость клиента или расширение экосистемы.

3. Описание существующих решений

Сбербанк. Один из наиболее продвинутых чат-ботов в российской банковской сфере. В последние годы Сбер внедряет решения на базе генеративных моделей и собственной платформы GigaChat. Чат-бот интегрирован в экосистему банка, что позволяет клиенту решать широкий круг задач — от финансовых до бытовых.

ВТБ. Чат-бот «Помощник ВТБ» способен охватывать более 1900 тем, обеспечивая автоматизацию до 85 % обращений клиентов. Основное преимущество заключается в широте функционала и способности обрабатывать запросы различной сложности.

Альфа-Банк. Чат-бот банка реализован с использованием технологий больших языковых моделей (LLM), что обеспечивает поддержку более чем 50 языков. Это позволяет обслуживать не только российских клиентов, но и международную аудиторию.

Т-Банк. Акцент сделан на голосовых технологиях и использовании собственной платформы VoiceKit. Чат-бот сочетает текстовое и голосовое взаимодействие, предоставляя клиентам более гибкий канал коммуникации.

Международные решения. Для сравнения можно выделить Erica (Bank of America), которая анализирует транзакции и формирует персональные финансовые советы, а также Eno (Capital One), акцентирующую внимание на безопасности платежей и контроле расходов. Эти примеры демонстрируют более высокий уровень персонализации и проактивности по сравнению с большинством российских аналогов.

4. Отличия

Основные отличия существующих решений заключаются в применении технологий и приоритетах развития.

- Сбербанк делает ставку на генеративные модели и экосистемный подход.
- ВТБ стремится охватить максимально широкий спектр клиентских запросов.
- Альфа-Банк выделяется мультиязычной поддержкой.
- Т-Банк активно развивает голосовое взаимодействие.

Международные решения отличаются большей ориентацией на персонализацию и прогнозирование поведения клиента, тогда как в российских реалиях акцент делается на автоматизацию стандартных процессов.

5. На чем основаны

Технологической основой чат-ботов выступают методы обработки естественного языка (NLP), машинного обучения и нейросетевых моделей.

- Для распознавания речи применяются акустические модели и алгоритмы ASR (automatic speech recognition).
- Для понимания и генерации текста используются трансформерные архитектуры, включая большие языковые модели (LLM).
- Для персонализации сервисов задействуются алгоритмы машинного обучения (ML), анализирующие транзакционную активность и поведенческие паттерны клиентов.

– Для выделения имен, дат, сумм, номера счета и адресов внедряется технология распознавания данных (NER) и сокращается количество шагов в диалоге.

С 1 марта 2023 года вступил в силу Федеральный закон от 29.12.2022 № 584-ФЗ, который ввёл запрет для российских банков и финансовых организаций на использование иностранных программных продуктов в сфере коммуникаций. Ограничение распространяется, в том числе, на обмен финансовыми и персональными данными. В перечень запрещённых решений вошли такие сервисы, как Telegram и Microsoft Teams. Соответственно, в банковской отрасли недопустимо применять, например, Telegram-ботов для обслуживания клиентов, если в процессе затрагивается персональная или финансовая информация.

Для соблюдения требований закона рекомендуется переходить на отечественные программные решения. Некоторые банки разрабатывают собственные решения (например, Сбер — GigaChat, Тинькофф — VoiceKit), что позволяет снизить зависимость от зарубежных технологий и адаптировать их под локальные условия.

6. Оценки пользователей

Пользователи в целом положительно оценивают внедрение чат-ботов, отмечая скорость и удобство получения информации. Например, в 2018 году доля запросов, требующих живого оператора, составляла около 70 %, а к 2024 году благодаря технологии машинного обучения эта цифра снизилась до 35 %. Согласно исследованиям, уровень удовлетворённости клиентов достигает 70–80 %, что сопоставимо с традиционным обслуживанием через операторов.

Однако сохраняются и проблемные зоны. Нередко клиенты выражают недовольство при столкновении с ограничениями чат-бота: непонимание сложных или нестандартных запросов, невозможность вести свободную беседу, избыточные переходы к оператору. Это указывает на необходимость дальнейшего совершенствования технологий обработки языка.

7. Что требует доработки

Несмотря на достигнутый прогресс, чат-боты в банковской сфере всё ещё требуют доработки по ряду направлений:

- развитие диалоговых возможностей, включая понимание сленга, сокращений и многоступенчатых запросов;
- углубление персонализации, что предполагает анализ истории клиента и формирование индивидуальных предложений;
- расширение мультиязычной поддержки;
- интеграция с внешними сервисами, выходящими за рамки финансовой сферы;
- повышение уровня доверия за счёт обеспечения конфиденциальности и безопасности коммуникации.

Заключение

Проведённый анализ показал, что российские коммерческие банки активно внедряют чат-боты, которые уже сегодня выполняют значительную часть функций клиентской поддержки. Наиболее сильными сторонами являются скорость обслуживания, снижение издержек и круглосуточная доступность. Однако в сравнении с международными решениями российские чат-боты пока отстают в части персонализации и проактивности.

Дальнейшее развитие связано с интеграцией больших языковых моделей, совершенствованием технологий NLP и машинного обучения, а также расширением функционала за пределы стандартных операций. Это позволит чат-ботам не только обслуживать клиентов, но и стано-

виться полноценными финансовыми ассистентами, способными прогнозировать потребности и предлагать индивидуальные решения.

Литература

1. Редакция [developers.sber.ru](https://developers.sber.ru/help/salutebot/opportunity). О SaluteBot. Возможности чат-бота – 2024. – URL: <https://developers.sber.ru/help/salutebot/opportunity> (дата обращения: 25.09.2025)
2. ВТБ. Как мы обучили чат-бота вести диалог с помощью искусственного интеллекта – 2025. – URL: <https://vc.ru/ai/1988947-obuchenie-chat-bota-vtb-s-iskusstvennym-intellektom> (дата обращения: 26.09.2025)
3. Паршин К. В Альфа-Банке заявили о внедрении агента для общения на иностранных языках – 2025. – URL: <https://lenta.ru/news/2025/07/10/v-alfa-banke-zayavili-o-vnedrenii-agenta-dlya-podderzhki-obscheniya-na-inostrannyh-yazykakh/> (дата обращения: 25.09.2025)
4. Т-Банк : официальный сайт. – 2025. – URL: <https://www.tbank.ru/software/voicekit/> (дата обращения: 25.09.2025)
5. Aldridge A. A Decade of AI Innovation: BofA's Virtual Assistant Erica Surpasses 3 Billion Client Interactions – 2025. – URL: <https://newsroom.bankofamerica.com/content/newsroom/press-releases/2025/08/a-decade-of-ai-innovation--bofa-s-virtual-assistant-erica-surpas.html> (дата обращения: 25.09.2025)
6. Capital One : официальный сайт. – 2025. – URL: <https://www.capitalone.com/digital/tools/eno/> (дата обращения: 25.09.2025)
7. Сатин Д. Банковские чат-боты: текущее состояние, тренды и лучшие практики – 2025. – URL: <https://usabilitylab.ru/blog/bankovskie-chat-boty/> (дата обращения: 25.09.2025)
8. Подольская Т. В. Практика внедрения компьютерного аудита и искусственного интеллекта в банковском секторе / Т. В. Подольская, А. П. Сидельников, Л. Гелашвили // Вопросы инновационной экономики. – 2021. – Т. 11, № 4. – С. 1493–1508. – DOI 10.18334/vinec.11.4.113673 (дата обращения: 25.09.2025)
9. eXpress. Чат-бот банка. Что это и как работает – 2024. – URL: <https://express.ms/blog/obzory/chat-boty-bankov-kak-rabotayut-tsifrovye-pomoshchniki/> (дата обращения: 25.09.2025)
10. Стрункин Р. Как банки используют искусственный интеллект для персонализации услуг и продуктов – 2025. – URL: <https://www.raiffeisen-media.ru/biznes/kak-banki-ispolzuyut-iskusstvennyj-intellekt-dlya-personalizacii-uslug-i-produktov/> (дата обращения: 25.09.2025)
11. Москаль А. Финополис-2025: налоговый бот от Сбера, бизнес-ассистент «Катюша» от ПСБ и акселератор для НКО от Т-Банка – 2025. – URL: <https://rb.ru/news/finopolis-2025-nalogovyj-bot-ot-sbera-biznes-assistent-katyusha-ot-psb-i-akselerator-dlya-nko-ot-t-banka/> (дата обращения: 25.09.2025)
12. Швецова А. Клиенты банков все больше отдают предпочтение чат-ботам – 2025. – URL: https://www.comnews.ru/content/232964/2024-05-03/2024-w18/1008/klienty-bankov-vse-bolshe-otdayut-predpochtenie-chat-botam?utm_source=chatgpt.com (дата обращения: 25.09.2025)

ПРИМЕНЕНИЕ VOXELMORPH В ЗАДАЧЕ СЕГМЕНТАЦИИ МЕДИЦИНСКИХ 3Д ИЗОБРАЖЕНИЙ

А. В. Черемискин¹, И. Л. Каширина^{1,2}

¹Воронежский государственный университет

²МИРЭА – Российский технологический университет

Аннотация. В работе рассматривается применение VoxelMorph для регистрации мультифазных КТ печени в задаче сегментации. Многофазные данные содержат ценную диагностическую информацию, однако смещения анатомических структур между фазами затрудняют автоматическую обработку. Использование VoxelMorph позволяет скорректировать эти различия за счёт моделирования нелинейных деформаций. После регистрации изображения использовались для обучения сегментационной модели ResUNet. Применение VoxelMorph демонстрирует потенциал для интеграции в мультифазные pipelines медицинской визуализации. Дальнейшие исследования могут быть направлены на оптимизацию совмещения регистрации и сегментации в единой архитектуре.

Ключевые слова: VoxelMorph, регистрация изображений, сегментация, ResUNet, поле смещений, анатомическое выравнивание, карцинома, медицинские изображения.

Введение

Сегментация медицинских изображений является одной из ключевых задач в современной компьютерной диагностике, так как точное выделение анатомических структур лежит в основе последующего анализа и лечения. Последние исследования [1] показали эффективность применения мультифазного подхода, при котором используются снимки с разных фаз контрастирования для одного пациента. Но на данный момент при таком подходе остаются сложности, связанные с изменением положения анатомических структур между фазами, что приводит к снижению точности сегментации. Одним из решений этой проблемы может быть регистрация изображений. Развитие нейронных сетей упростило этот процесс, в частности VoxelMorph [2] зарекомендовал себя как эффективный метод регистрации, способный учитывать сложные пространственные взаимосвязи. Цель данной работы — оценить, как влияет применение VoxelMorph регистрации на итоговую точность сегментации.

1. Материалы и методы

В исследовании был использован набор данных WAW-TACE [3], в котором содержатся данные пациентов с гепатоцеллюлярной карциномой. В этом наборе данных представлены КТ-снимки с разных фаз контрастирования, маски сегментации опухолей, маски сегментации внутренних органов, автоматически сгенерированные при помощи «Total Segmentator» [4]. Важно отметить, что у каждого пациента маска опухоли была размечена на одной фазе контрастирования, таким образом, для пространственного соответствия маски КТ-снимкам других фаз, необходима регистрация изображений. Из всех пациентов были выбраны те, у которых присутствуют все фазы контрастирования. Затем все КТ снимки были приведены к единому разрешению по оси Z с толщиной среза 1мм. Кроме того, для фокусирования обучения моделей на области печени все воксели вне маски печени, полученной при помощи «Total Segmentator», были заменены на -1024, что соответствует значению воздуха по шкале Хаунсфилда. Также на этапе предобработки было выполнено смещение по Z оси тех КТ-снимков, для которых впоследствии будет применяться VoxelMorph регистрация. Величина смещения определялась по

максимальному совпадению профилей масок печени между фазами. Данные были разделены на тренировочную и тестовую выборки в пропорции 85 % к 15 % соответственно.

Используемая архитектура VoxelMorph построена на U-Net подобной структуре с последовательными этапами энкодера и декодера, каждый из которых состоит из сверточных слоев с ядром $3 \times 3 \times 3$. После каждого сверточного слоя использовался слой активации LeakyReLU. В энкодере для постепенного понижения пространственного разрешения используются свертки с $co\ stride = 2$. Декодер постепенно восстанавливает пространственное разрешение, чередуя сверточные слои с Upsample слоями. Кроме того, в данной архитектуре используются пропускные соединения (skip-connections), которые передают данные из энкодера в соответствующие слои декодера. На вход сети подается пара изображений — фиксированное (fixed) и подвижное (moving), которые объединяются в единый двухканальный объем. Предсказанием сети является трехмерное поле смещений (flow), описывающее деформацию, необходимую для приведения подвижного изображения к фиксированному. Архитектура сети представлена на рис. 1.

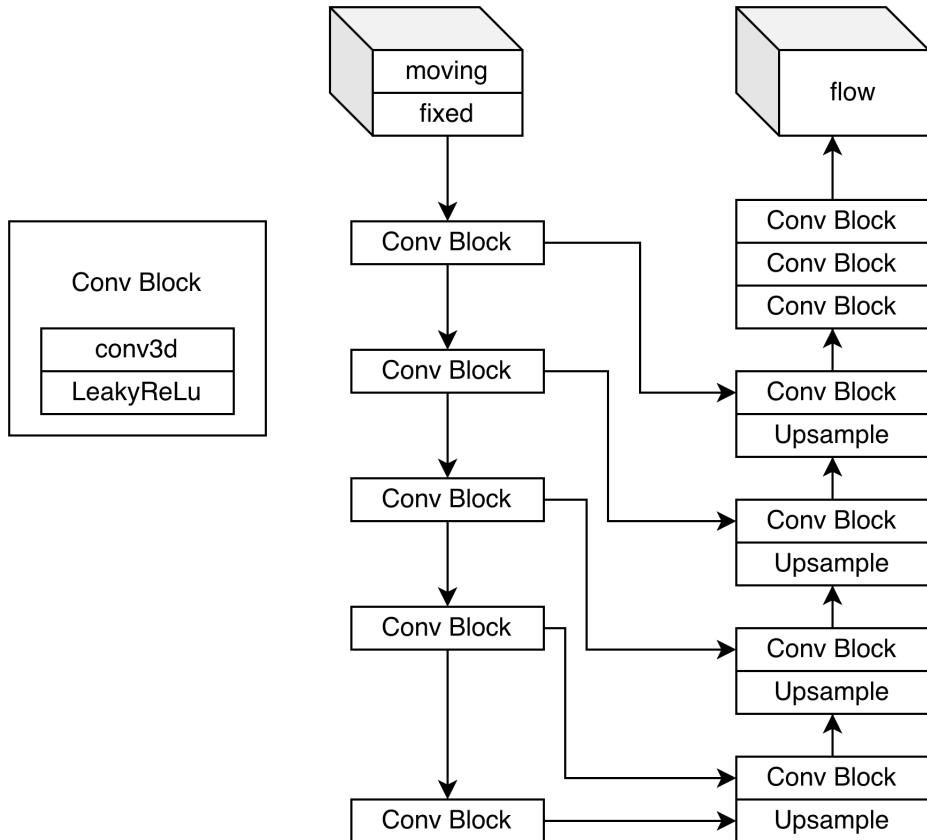


Рис. 1. Архитектура VoxelMorph

В качестве функции ошибки использовалась комбинация метрики сходства между зарегистрированным и фиксированным изображениями и регуляризации поля смещений. В качестве метрики сходства использовалась локальная кросс-корреляция (CC).

$$CC(f, m \circ \phi) = \sum_{p \in \Omega} \frac{\left(\sum_{p_i} (f(p_i) - \hat{f}(p))([m \circ \phi](p_i) - [\hat{m} \circ \phi](p)) \right)^2}{\left(\sum_{p_i} (f(p_i) - \hat{f}(p))^2 \right) \left(\sum_{p_i} ([m \circ \phi](p_i) - [\hat{m} \circ \phi](p))^2 \right)},$$

где f — фиксированное изображение, $m \circ \phi$ — зарегистрированное изображение, $\hat{f}(p_i)$ — среднее значение вокселей фиксированного изображения в окрестности $9 \times 9 \times 9$ вокселя p , $[\hat{m} \circ \phi](p_i)$ — среднее значение вокселей зарегистрированного изображения в окрестности $9 \times 9 \times 9$ вокселя p .

Регуляризация используется для того, чтобы поле деформаций было гладким и физически реалистичным. Ошибка регуляризации выглядит следующим образом

$$L_{smooth}(\phi) = \sum_{p \in \Omega} \left\| \left(\frac{\partial u(p)}{\partial x}, \frac{\partial u(p)}{\partial y}, \frac{\partial u(p)}{\partial z} \right) \right\|,$$

где u — поле смещений, при этом используется аппроксимация $\frac{\partial u(p)}{\partial x} \approx u((p_x + 1, p_y, p_z)) - u((p_x, p_y, p_z))$, аналогично для y, z .

Итоговая функция ошибки выглядит следующим образом

$$L_{us}(f, m, \phi) = -CC(f, m \circ \phi) + \lambda L_{smooth}(\phi).$$

Обучение VoxelMorph проводилось на тренировочной выборке в течении 50 эпох. Для каждого пациента выбиралась целевая фаза — та, на которой были размечены опухоли. Снимки с остальных фаз выступали в качестве подвижного изображения. Для экономии ресурсов КТ снимки обрабатывались патчами размера $48 \times 256 \times 256$. Итоговый прогноз в свою очередь выполнялся методом скользящего окна.

В качестве модели сегментации использовалась нейронная сеть ResUNet. Эта архитектура сочетает классическую U-Net структуру с остаточными блоками (residual blocks), что облегчает обучение глубоких нейронных сетей и предотвращает затухание градиентов. Обучение проводилось в течении 300 эпох, в качестве функции ошибки использовалась TverskyLoss с $a = 0.3$, $b = 0.7$.

$$TverskyLoss = 1 - \frac{TP}{TP + a \cdot FP + b \cdot FN},$$

где TP — количество истинно положительных вокселей, FP — количество ложно положительных вокселей, FN — количество ложно отрицательных вокселей, a — коэффициент, регулирующий важность ложно положительных ошибок, b — коэффициент, регулирующий важность ложно отрицательных ошибок.

Модель обучалась на артериальной фазе контрастирования. Для сравнения модель была обучены как на исходных данных, так и на данных после регистрации.

2. Результаты и обсуждение

Пример результатов работы VoxelMorph представлен на рис. 2.

Можно заметить, что помимо пространственного сдвига, применение VoxelMorph приводит к локальному сглаживанию изображения.

Полученные в результате обучения ResUNet результаты представлены в табл. 1.

Результаты тестирования показали, что применение VoxelMorph в качестве регистрации к набору данных, несмотря на качественное пространственное смещение, приводит к уменьше-

Таблица 1

Результаты обучения ResUNet моделей

Показатель	Исходные данные	Зарегистрированные изображения
Dice	0.64	0.57
Dice Global	0.66	0.67
Процент зарегистрированных изображений в тренировочной выборке	0 %	48 %
Процент зарегистрированных изображений в тестовой выборке	0 %	41 %

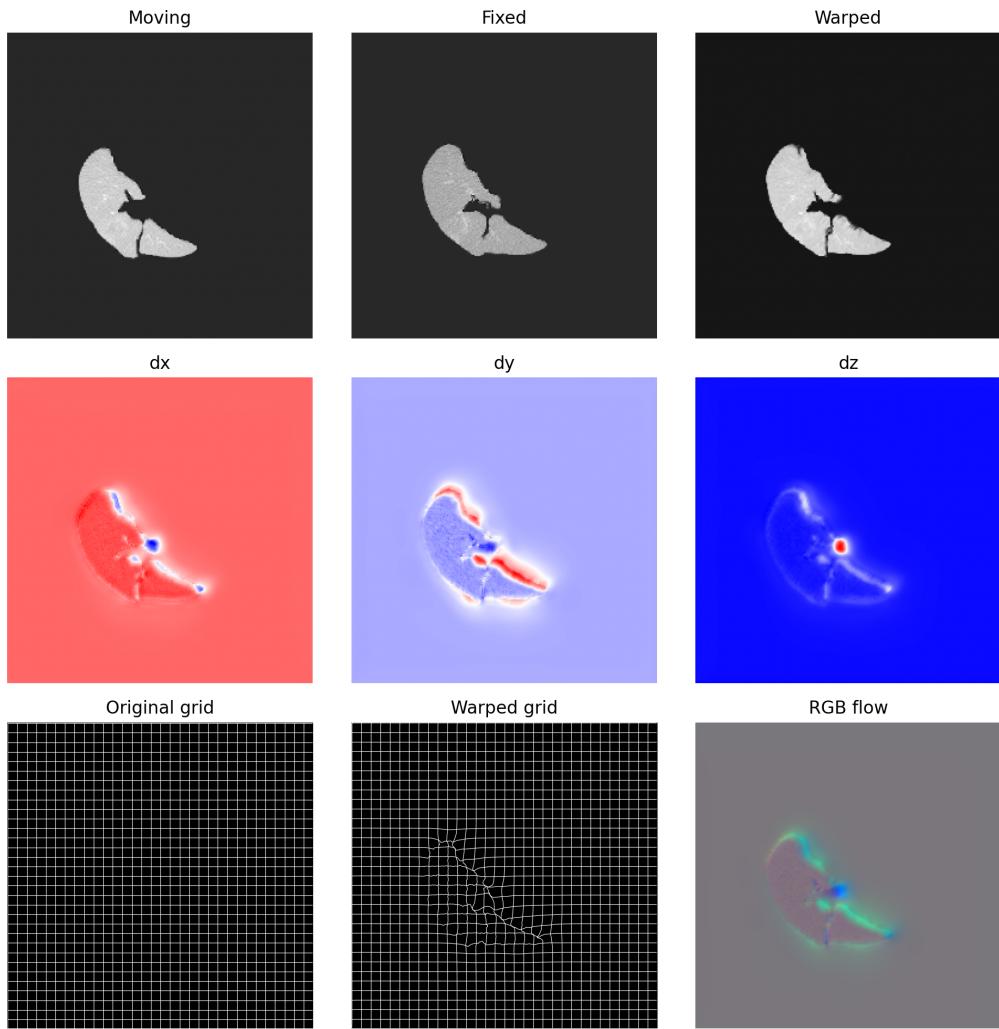


Рис. 2. Результат применения VoxelMorph, где warped — зарегистрированное изображение, dx, dy, dz — смещения по трем осям. Original и Warped grid — система координат до и после деформации. RGB flow — объединенное цветовое представление поля смещений

нию метрики Dice, при одновременном улучшении Dice Global. Такое поведение чаще всего означает, что сеть лучше распознает крупные объекты и хуже небольшие. В задаче сегментации опухолей зачастую важнее различать именно маленькие патологии, таким образом снижение локальной точности сегментации является критичным.

Полученные результаты могут указывать на то, что процесс регистрации сглаживает мелкие структуры, это также можно заметить на рис. 2. При этом улучшение Dice Global говорит о лучшем смещении крупных анатомических областей, что ожидаемо для модели, которая создает гладкое поле смещений.

Заключение

Данное исследование показывает, что применение VoxelMorph может быть полезным в задачах, где ключевую роль играет корректное выравнивание больших анатомических структур, однако для задач сегментации мелких патологий такой подход требует дополнительной настройки. Возможными направлениями улучшения могут быть: снижение вклада регуляризации в функцию ошибки, совместное обучение регистрации и сегментации (joint training).

Литература

1. Multi-target and multi-stage liver lesion segmentation and detection in multi-phase computed tomography scans [Электронный ресурс] / A. F. Al-Battal [и др.]. – 2024. – Режим доступа: <https://arxiv.org/abs/2404.11152>. (Дата обращения: 19.06.2025).
2. VoxelMorph: A Learning Framework for Deformable Medical Image Registration / Balakrishnan G [и др.] // IEEE Transactions on Medical Imaging. — 2019. — Vol. 38. — DOI: 10.1109/TMI.2019.2897538.
3. WAW-TACE: A Hepatocellular Carcinoma Multiphase CT Dataset with Segmentations, Radiomics Features, and Clinical Data / K. Bartnik [и др.] // Radiology: Artificial Intelligence. – 2024. – Т. 6, № 6, ст. e240296. – DOI: 10.1148/ryai.240296.
4. TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images / J. Wasserthal [и др.] // Radiology: Artificial Intelligence. – 2023. – Т. 5, № 5, ст. e230024. – DOI: 10.1148/ryai.230024.

ДИАГНОСТИЧЕСКИЕ ДЕРЕВЬЯ ДЛЯ ЛЕЙКОПЛАКИИ РОТОВОЙ ПОЛОСТИ

М. А. Шашкина, И. Н. Сарычева

Воронежский государственный университет
Стоматологическая клиника ВГМУ им. Н.Н. Бурденко

Аннотация. В статье рассматривается алгоритм построения диагностических правил для автоматизации процесса принятия решений с целью диагностики форм лейкоплакии ротовой полости, которая относится к предраковым заболеваниям. В качестве исходной информации используются результаты, полученные методом оптической спектроскопии в виде совокупности интенсивностей флуоресценции (ИФ) для различных наборов волн. Диагностическое дерево построено с использованием метода деревьев решений.

Ключевые слова: лейкоплакия, интенсивность флуоресценции, деревья решений, диагностические правила.

Введение

Лейкоплакия ротовой полости — поражение слизистой, возникающее вследствие постоянного раздражения и сопровождающееся повышенным ороговением (гиперкератозом) [1]. Опасность лейкоплакии заключается в том, что она может переродиться в рак, поэтому ее ранняя диагностика крайне важна.

Под медицинской диагностикой понимается процесс установления диагноза — заключения о болезни и состоянии пациента, выраженное в принятой медицинской терминологии [2]. Задача медицинской диагностики заключается в том, чтобы на основе совокупности показателей, характеризующих состояние пациента, из некоторой совокупности заболеваний выделить то, которое в максимальной степени соответствует состоянию пациента. Сложность постановки диагноза заключается в том, что некоторые симптомы могут быть характерными сразу для нескольких заболеваний. Кроме того, даже при одном и том же заболевании у двух пациентов симптомы могут проявляться с разной интенсивностью. Опытный врач принимает во внимание весь комплекс проявляющихся симптомов, его опыт позволяет сократить множество возможных заболеваний до минимума, а затем, применяя дополнительные методы диагностики, установить заболевание и назначить пациенту лечение. Процесс рассуждений врача относительно тех показателей, которые выделены, можно формализовать в виде диагностического дерева, которое, по сути, относится к деревьям решений [3].

Статья посвящена описанию алгоритма построения диагностического дерева, который определяет набор показателей и последовательность их рассмотрения при диагностике лейкоплакии ротовой полости, а также его программной реализации.

1. Общий подход к диагностике лейкоплакии ротовой полости

1.1. Основные формы лейкоплакии

Лейкоплакия — поражение слизистой оболочки, вызванное нарушениями на клеточном уровне, которое проявляется очаговым ороговением многослойного плоского эпителия. Пораженные участки представляют собой единичные или множественные белесоватые или бело-серые очаги с четкими контурами различной формы и размеров. Появление лейкоплакии связано с различными травмирующими факторами, а также наследственной предрасположенностью, наличием вируса папилломы человека, сахарным диабетом, болезнями органов желу-

дочно-кишечного тракта. Лейкоплакия относится к предраковым заболеваниям. В процессе развития ЛРП выделяют две основные формы [4], которые представляют собой этапы патологического процесса и требуют специальных подходов к лечению: *плоская лейкоплакия*, протекающая с невыраженной симптоматикой (больной жалуется только на повышенную сухость слизистой оболочки, симптомы натянутости губ, щек); *веррукозная лейкоплакия* (бляшечная и эрозивная формы) — происходит прогрессивное орогование, есть большая вероятность перехода в злокачественный процесс.

Для диагностики типа лейкоплакии предлагается использовать технологию деревьев решений. Диагностическое дерево учитывает взаимосвязи между значениями показателей (атрибутов) причем с учетом динамически формируемых приоритетов. В каждом узле дерева находится предикат, истинность или ложность которого является основой для ветвления. Прогнозирование (предсказание) или классификация на основе дерева решений осуществляется как перемещение по дереву от корня к некоторой висячей вершине.

1.2. Диагностика ЛРП на основе оптической спектроскопии

Основной метод диагностики лейкоплакии основан на биопсии пораженных участков слизистой с последующим гистологическим и цитологическим исследованием полученного материала. Однако, для оценки вероятности перехода повреждений в раковую опухоль возникает необходимость в привлечении дополнительных диагностических методов, в частности, методов оптической спектроскопии: лазерно-индуцированная флуоресценция (ЛИФ) и метод диффузного отражения (МДО).

В проведенном исследовании спектры флуоресценции регистрируются с помощью устройства, созданного на базе волоконно-оптического спектрометра, сопряженного с компьютером, и зонда для измерения отражения в форме волоконного жгута. Дополнительно используется метод спектроскопии отражения в видимом и ближнем ИК-диапазоне длин волн на основе волоконно-оптического спектрофотометра. Экспериментально были получены спектры диффузного отражения от участков ткани слизистой, пораженной простой формой лейкоплакии и веррукозной формой лейкоплакии, а также интактных участков слизистой полости рта. Визуализация результатов позволяет сделать следующие выводы:

- 1) спектры отражения интактной слизистой и слизистой, пораженной лейкоплакией, существенно отличаются по интенсивности, причем имеют разнонаправленную зависимость;
- 2) все спектральные полосы для пораженной слизистой более размыты по сравнению со спектром интактной слизистой.

Таким образом, для раннего выявления поражений ротовой области и диагностики форм лейкоплакии целесообразно использовать спектральный анализ с привлечением методов машинного обучения. Ясно, что при таком подходе требуется сформировать качественное обучающее множество (рис. 1). Исходная информация формируется для двух групп пациентов: первая группа — больные веррукозной формой, вторая группа — больные простой формой. Данные собирают с помощью специального прибора путем облучения пораженных и здоровых участков различными длинами волн (для каждой длины волны делается 14 измерений), тем самым, для каждого пациента формируется вектор измерений ИФ, каждая компонента которого соответствует длине волны и означает интенсивность флуоресценции.

Цель исследования заключалась в том, чтобы выделить характерные диагностические признаки для веррукозной и простой форм лейкоплакии, что позволит уточнить диагноз и назначить соответствующую терапию. Идея алгоритма для обработки выборочных данных заключается в переходе к средним величинам с последующим анализом графика зависимости значений ИФ от длины волны. Для корректности необходимо, чтобы выполнялись гипотезы о

равенстве математических ожиданий значений ИФ при каждой длине волны для всех пациентов, что дает основание для выводов относительно поведения «обобщенного» графика.

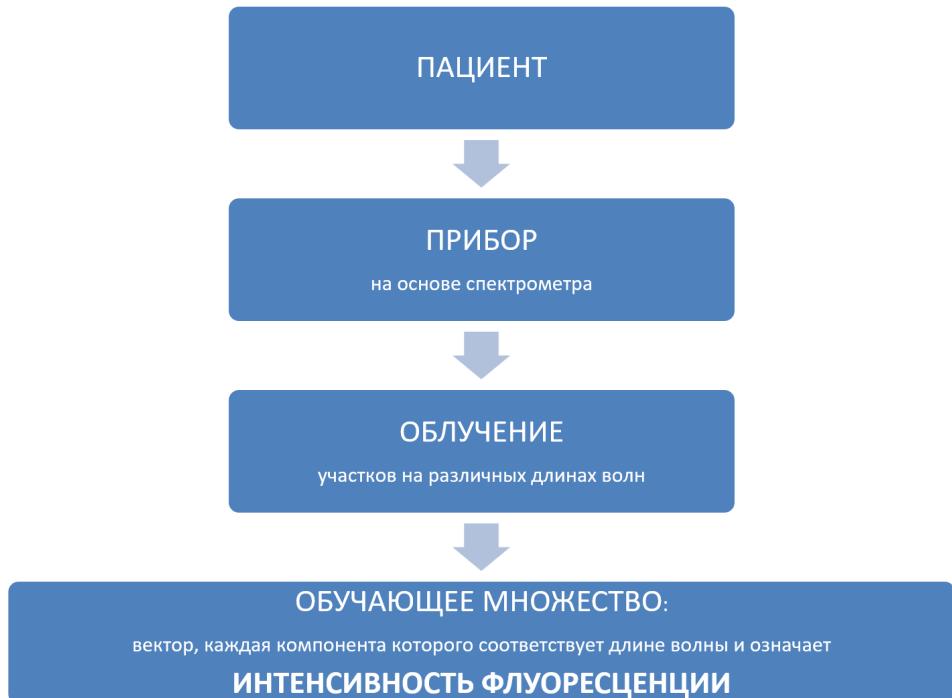


Рис. 1. Схема формирования обучающего множества

Но чтобы проверять гипотезу о равенстве средних необходимо установить, что выборочные данные являются независимыми и имеют нормальное распределение. Таким образом, можно сформулировать следующий алгоритм обработки данных для построения диагностического дерева, основываясь на [3, 5, 6].

Алгоритм построения диагностического дерева

1. *Общий анализ качества данных.* На данном этапе целесообразно проанализировать данные (есть ли пропуски, выбросы и т. п.).
2. *Усреднение данных:* а) каждому пациенту при каждой длине волны ставится в соответствие среднее значение показателя; б) по каждой длине волны вычисляется среднее по всем пациентам.

3. *Проверка гипотезы о независимости и нормальности значений ИФ для каждого пациента.*

4. *Проверка гипотезы о равенстве средних значений ИФ по всем пациентам при каждой длине волны.*

5. *Анализ максимумов и минимумов:*

а) выделить максимумы и минимумы на обобщенных графиках для простой и веррукозной форм, они соответствуют определенной длине волны;

б) отдельно для каждой группы пациентов с веррукозной и простой формами по каждому пациенту в исходной таблице определить максимальные и минимальные значения ИФ для каждой длины волны.

6. *Построение диагностического дерева.*

Рассмотрим некоторые особенности реализации данного алгоритма.

Переход к усредненным значениям предполагает следующие действия: для каждого пациента усредняем измерения при фиксированной длине волны, тем самым, для каждого пациента будет сформирована векторная оценка, каждая компонента которой есть усредненное значение ИФ, соответствующее определенной длине волны. На основе средних строятся графики

средних значений в единой системе координат для каждого пациента (по оси абсцисс — среднее значение измерений, по оси ординат — длины волн) и обобщенный график усредненных значений по всем пациентам (ось абсцисс — длина волны, ось ординат — среднее значение ИФ) для каждой группы.

Проверка гипотезы о независимости и нормальности значений ИФ для каждого пациента может быть проведена на основе гистограмм, значений эксцесса или асимметрии или с помощью таких тестов, как график Колмогорова — Смирнова, Шапиро — Уилка или график Q-Q [7].

Проверка гипотезы о равенстве средних значений ИФ по всем пациентам при каждой длине волны осуществляется классическим способом и реализована на основе ANOVA-тест в Python с помощью библиотеки SciPy. В качестве группирующей переменной берется номер пациента, в качестве зависимой переменной — среднее значение ИФ. Результатом данного теста является F -статистика, которая определить изменчивость между выборками и внутри выборок. Если расчетное значение F меньше 0,05, то нулевая гипотеза отвергается, а альтернативная — принимается. Таким образом, ANOVA-тест используется для определения статистически значимого различия между группами. Если оно существует, т. е. верна альтернативная гипотеза, то необходимо определить, в каких группах различие наблюдается.

2. Реализация алгоритма построения диагностического дерева

В группу было включено $N = 68$ пациентов. У каждого пациента с исследуемой зоны при каждой длине волны снималось 14 значений ИФ. Таким образом, исходная информация представляет собой таблицу, в которой каждому пациенту при каждой длине волны соответствует 14 значений ИФ.

В соответствии с изложенным в предыдущем разделе алгоритмом было построено диагностическое дерево с использованием библиотек Phyton. Перечислим основные шаги и особенности их реализации:

1. Импорт необходимых библиотек и модулей.

2. Подключение Google диска и загрузка данных.

3. Подготовка данных — проверка пропущенных значений, определение типов данных для каждого столбца, формирование отчета.

4. Выполнение одностороннего ANOVA-теста:

а) преобразование данных в длинный формат: данные преобразуются из широкого формата в длинный с помощью функции `pd.melt()`, при этом Patient_ID и Form остаются как идентификаторы, а новые столбцы получают названия Wavelength и Fluorescence (длины волн переводятся в числовой формат);

б) определение функции для выполнения ANOVA для каждой длины волны и формы: функция `perform_anova` выбирает подмножество данных для указанной формы, затем для каждой длины волны строит модель обыкновенных наименьших квадратов (OLS) и вычисляет таблицу ANOVA;

в) выполнение ANOVA и преобразование результатов в DataFrame;

г) определение функции для формирования результатов: функция `apply_conditional_formatting` обеспечивает визуальное выделение значений красным цветом, где гипотеза отвергается на основе p-value.

д) сохранение результатов: задаются пути для сохранения файлов, далее используется `pd.ExcelWriter` для создания и записи данных в файл Excel.

Результаты дисперсионного анализа показывают, что гипотеза о равенстве средних значений по всем пациентам с простой формой отвергается при уровне значимости $p = 0,05$ для следующей длины волны: 379,25.

Сравнение графиков средних значений для пациентов с различными формами лейкоплакии показывает, что каждый график имеет характерные точки минимума и максимума, причем у всех пациентов с простой (рис. 2) и verrukозной (рис. 3) формами точки минимума и максимума совпадают, а, следовательно, графики средних значений имеют характерные максимумы и минимумы при одинаковых значениях волн.

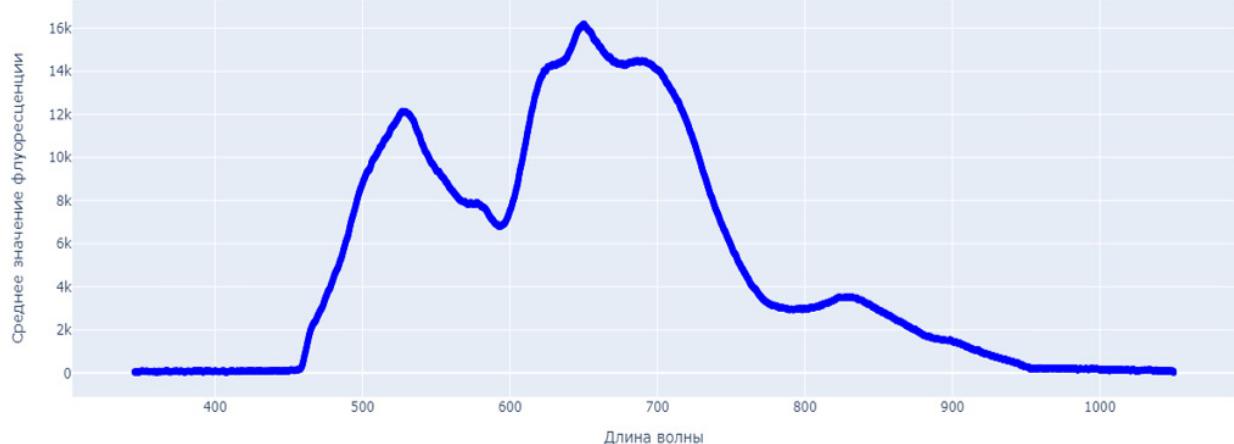


Рис. 2. График средних значений при каждой длине волны по всем пациентам с простой формой

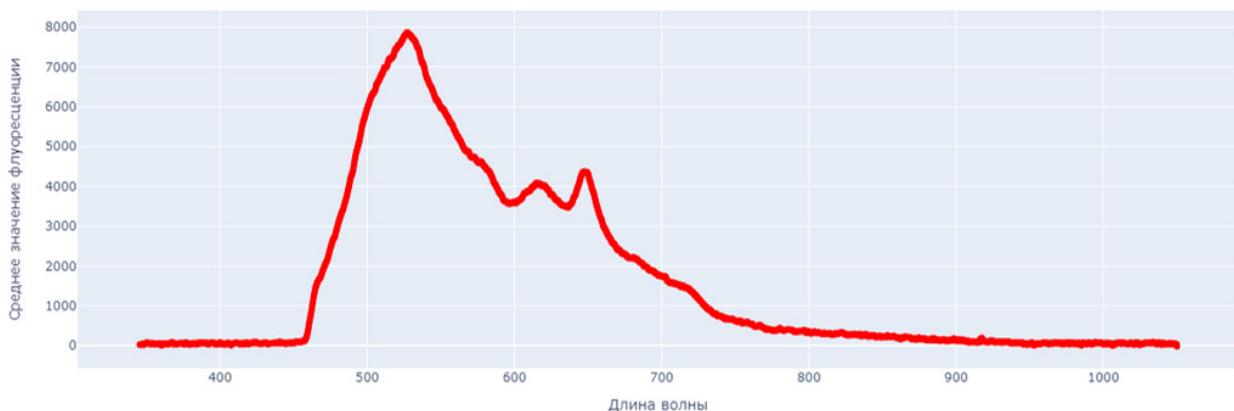


Рис. 3. График средних значений при каждой длине волны по всем пациентам с verrukозной формой

Установлено, что для всех длин волн, на которых достигаются минимумы и максимумы, выполнены предположения о том, что выборка получена из нормальной генеральной совокупности. Затем для простой и verrukозной форм были получены следующие описательные статистики для значений флуоресценции в точках максимума и минимума:

- Mean — среднее выборочное значение флуоресценции по всем пациентам;
- [confidence –0,95; confidence –0,95] — это интервал, который с вероятностью 0,95 накрывает истинное среднее значение по всей генеральной совокупности;
- Median — характеризует такое значение флуоресценции, что вероятность попасть левее данного значения равна вероятности попасть правее и равна $p = 0,5$;
- Minimum/maximum — минимальное/максимальное значение флуоресценции;
- Lower/Upper Quartile — нижняя и верхняя квартили, Lower Quartile — такое значение флуоресценции, что 25 % пациентов имеют значение ниже данного, Upper Quartile означает, что 25 % пациентов имеют оценку флуоресценции выше данного значения.
- Range — это разброс между минимальным и максимальным значением;
- Std. Dev. — стандартное отклонение характеризует средний разброс вокруг среднего.

Чтобы определить, можно ли по поведению графиков в точках минимума и максимума построить биомаркеры, применялся метод дерева решений, который позволяет строить диагно-

стистические правила. Данные включают 68 строк (40 пациентов с verrukозной формой и 28 пациентов с простой формой) и 9 столбцов, содержащих значения флуоресценции в различных точках максимума и минимума. Десятый столбец содержит метку формы лейкоплакии (0 — простая форма, 1 — verrukозная форма). Данные были разделены на обучающую и тестовую выборки в пропорции 80:20 (%). Обучающая выборка использовалась для построения модели, а тестовая — для оценки её качества. Для классификации использовался алгоритм Decision Tree Classifier из библиотеки Scikit-learn. Модель была обучена на обучающей выборке.

Визуализация фрагмента дерева решений для значения волны 650.19 представлена на рис. 4.

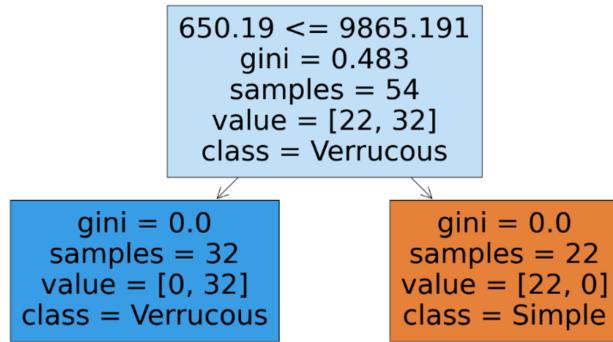


Рис. 4. Дерево решений для длины волны 650.19

Для оценки качества дерева использовалось несколько характеристик (коэффициент Gini, кросс-валидация, *Confusion Matrix* и *Classification Report*). Затем рассмотренная длина волны убирается из рассмотрения и берется следующее значение.

В результате итерационного процесса, когда на каждой итерации рассматривается определенная длина волны, были построены диагностические правила:

1. Если значение интенсивности меньше или равно 9865, 191, то форма verrukозная, в противном случае, простая.
2. Если значение интенсивности меньше или равно 9651,158, то форма verrukозная, в противном случае, простая.
3. Если значение интенсивности меньше или равно 1830,707, то форма verrukозная, в противном случае, простая.
4. Если значение интенсивности меньше или равно 1555,327, то форма verrukозная, в противном случае, простая.
5. Если значение интенсивности меньше или равно 1677,871, то форма verrukозная, в противном случае, простая.
6. Если значение интенсивности меньше или равно 5126,96, то форма verrukозная, в противном случае, простая.
7. Если значение интенсивности меньше или равно 9805,6, то форма verrukозная, в противном случае, простая.
8. Если значение интенсивности меньше или равно 5052,256, то форма verrukозная, в противном случае, простая.
9. Если значение интенсивности меньше или равно 1631, 552, то форма verrukозная, в противном случае, простая.

Разработанный программный модуль прошел тестирование на реальных данных и показал высокую точность в распознавании форм лейкоплакии. Полученные результаты демонстрируют потенциал использования машинного обучения в медицинской практике и открывают возможности для дальнейших исследований и улучшений.

Программный код был написан в среде google.colab, синхронизированной с google.drive.

Заключение

Автоматизация процессов принятия диагностических решений повышает качество и эффективность оказания медицинской помощи. Важнейшей составляющей для формализации диагностических решений является диагностическое дерево, которое, с одной стороны, может быть построено опытным специалистом-врачом, а, с другой, — с привлечением методов машинного обучения при условии, если имеется подходящая выборка. В представленной статье рассматривается задача построения диагностического дерева для лейкоплакии ротовой полости на основе интенсивности флуоресценции. Основным результатом исследования является совокупность правил принятия решений на основе показателей ИФ.

Литература

1. Григорьев С. С. Гиперкератозы слизистой оболочки рта (красный плоский лишай, лейкоплакия): учебно-методические рекомендации / С. С. Григорьев, Г. И. Ронь, А. А. Епишова. – Екатеринбург : Издательский дом «Тираж», 2019. – 72 с.
2. Медицинская диагностика. – https://ru.wikipedia.org/wiki/Медицинская_диагностика (дата обращения 6.06.2024).
3. Груздев А. В. Прогнозное моделирование в IBM SPSS Statistics, R и Phyton: Метод деревьев решений и случайный лес / А. В. Груздев. – Москва : ДМК Пресс, 2017. – 642 с.
4. Гончарик П. В. Лейкоплакия слизистой оболочки полости рта / П. В. Гончарик, Р. Н. Супруновский, Г. Д. Панасюк. – Гомель : ГУ «РНПЦ РМиЭЧ», 2019. – 27 с.
5. Дюк В. А. Информационные технологии в медико-биологических исследованиях / В. А. Дюк, В. Эммануэль. – Санкт-Петербург : Питер, 2003. – 528 с.
6. Реброва О. Ю. Статистический анализ медицинских данных. Применение пакета прикладных программ STATISTICA / О.Ю. Реброва. – Москва : МедиаСфера, 2002. – 125 с.
7. Бююль А. SPSS: искусство обработки информации. Анализ статистических данных и восстановление скрытых закономерностей. – Санкт-Петербург : ДиаСофтЮП, 2002. – 608 с.

СОВРЕМЕННЫЕ ПОДХОДЫ К ПРОБЛЕМЕ РАЗОБУЧЕНИЯ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

Н. А. Шелоумов, В. А. Пастухова

МИРЭА – Российский технологический университет

Аннотация. В статье представлен обзор методов машинного разобучения, позволяющих удалять влияние данных из обученных моделей без переобучения. Рассматриваются точные и приближённые подходы, описаны ключевые методы (SISA, ARCANE, PGU, GAU, Influence Functions) и специализированные архитектуры с акцентом на безопасность и приватность. Обсуждаются современные вызовы: квантование моделей, нормативные требования и метрики оценки качества. Выводы подчеркивают необходимость разработки гибридных и сертифицированных алгоритмов, повышения масштабируемости, стандартизации и формальной верификации для обеспечения приватности, надёжности и доверия в применениях машинного разобучения.

Ключевые слова: разобучение нейронных моделей (machine unlearning), метрики эффективности, аттестация моделей, приватность данных, безопасность искусственного интеллекта, устойчивость к атакам.

Введение

Масштабное применение нейронных сетей в различных прикладных задачах сопровождается непреднамеренным накоплением в них чувствительных данных пользователей и компаний. В условиях традиционных подходов удаление частей обучающего набора данных требует полного переобучения модели, что в случае с глубокими архитектурами зачастую приводит к значительным временным и вычислительным затратам. Формирование концепции машинного разобучения (machine unlearning) обусловлено необходимостью обеспечить возможность выборочного «забывания» данных моделью без возврата к начальному этапу обучения.

Выборочное удаление информации из уже обученной модели является критически важным элементом не только в контексте ускорения процессов поддержки и обновления различных систем искусственного интеллекта, но и с точки зрения выполнения законодательных требований. В частности, Общий регламент по защите данных ЕС (GDPR) устанавливает право субъекта персональных данных на удаление информации о нём из всех хранилищ и аналитических систем, включая модели машинного обучения. Такие же положения присутствуют в законодательной базе крупных мировых государств, таких как США и Франция, что подчеркивает значимость исследований в области машинного разобучения [1].

Настоящая работа направлена на систематизацию существующих подходов к машинному разобучению, анализ их преимуществ и ограничений, а также выявление основных вызовов и перспективных направлений исследований. Погружение в тему позволит сформулировать практические рекомендации по выбору и применению методов разобучения в различных прикладных областях, где вопросы приватности и соответствия правовым нормам играют ключевую роль.

1. Классификация подходов к машинному разобучению

Механизмы машинного разобучения можно разделить на две ключевые категории в зависимости от того, какая точность требуется при исключении влияния данных на итоговую версию модели [4].

Точное разобучение (или exact unlearning). Целью точного разобучения является достижение такого поведения модели, которое будет эквивалентно сценарию, где удалённые данные изначально не участвовали в процессе обучения модели.

Приблизительное разобучение (или approximate unlearning). Приблизительное разобучение направлено на снижение влияния удалённых данных до уровня, на котором их значимость статистически неотличима от нуля, при этом не ставя целью полное их удаление. При этом вычислительные затраты ниже, чем при точном разобучении, ведь происходят локальные корректировки, а не дополнительное переобучение частей модели. В основе большинства таких методов лежит идея градиентной коррекции, при которой веса модели обновляются в направлении, минимизирующем влияние удалённых данных. Такой процесс выражается следующим уравнением:

$$\theta_{t+1} = \theta_t - \eta P_{\perp G}(\nabla_{\theta} L(D_f; \theta_t)), \quad (1)$$

где θ_t — параметры модели на итерации t , η — шаг обучения, $L(D_f; \theta_t)$ — функция потерь на данных, подлежащих удалению, $P_{\perp G}$ — оператор проекции на подпространство, ортогональное к сохранённым градиентам G , с целью минимизации воздействия на сохраненные параметры модели.

2. Алгоритмические методы машинного разобучения

Алгоритмические методы машинного разобучения направлены на реализацию как точного, так и приблизительного удаления влияния данных с учётом требований к вычислительным ресурсам и гарантиям приватности. Рассмотрим ключевые методы и их главные особенности в табл. 1.

Таблица 1

Основные методы машинного разобучения

Класс метода	Название метода	Краткое описание
Точное разобучение	SISA (Sharded, Isolated, Sliced, Aggregated)	Разделение обучающего набора на независимые сегменты и переобучение лишь затронутых частей модели [2]
	ARCANE	Архитектура с декомпозицией модели на подмодели, позволяющая локально обновлять параметры без полного переобучения [9]
Приближенное разобучение	Projected Gradient Unlearning (PGU)	Проекция градиента обновления на ортогональное подпространство, минимизация влияния удаляемых данных [3]
	Gradient Ascent Unlearning (GAU)	Целенаправленное увеличение функции потерь на удаляемых примерах для подавления их влияния
	Influence Functions	Аппроксимация влияния каждого примера с помощью второй производной для локальной коррекции
	Newton Unlearning	Метод Ньютона с кубической регуляризацией для устойчивой и быстрой оптимизации
	SURE (Saliency-Based Unlearning)	Обновление только наиболее значимых параметров, определенных через карты значимости (saliency maps)

Каждый из перечисленных методов обладает уникальным сочетанием преимуществ и ограничений. Выбор конкретного алгоритма зависит от требований к точности удаления, до-

ступных вычислительных ресурсов, требуемого уровня формальных гарантий удаления информации и специфики прикладной задачи.

3. Специализированные архитектуры и применения

3.1. Forgetting Neural Networks (FNN)

Forgetting Neural Networks (FNN) — нейронные сети, включающие специальные слои или нейроны с заданными параметрами «скорости забывания». После подачи сигнала о необходимости удаления данных соответствующие параметры встраиваемых слоёв адаптивно уменьшаются или совсем обнуляются, эмулируя тем самым биологические механизмы забывания внутри нейронной сети. Для количественного описания процесса забывания в FNN часто используют функцию затухания весов нейронов по мере обучения:

$$\omega_i^{(t+1)} = \omega_i^{(t)} \cdot (1 - \lambda_t), \quad (2)$$

где $\omega_i^{(t)}$ — значение веса на итерации t , λ_t — параметр, регулирующий скорость «забывания».

3.2. Разобучение LLM (Large Language Models)

В связи с увеличением масштаба и архитектурной сложности больших языковых моделей, возникла необходимость в специально адаптированных алгоритмах разобучения, решающих задачи удаления токсичного и недостоверного контента, соответствия требованиям авторских прав и снижения вероятности генерации «галлюцинаций» и ошибочных ответов. Методы разобучения LLM включают техники soft prompting (обучение небольших дополнительных векторов и наборов параметров), gradient ascent (целенаправленное повышение значения функции потерь по удаляемым примерам и параметрам для удаления информации) и методы fine-tuning (дообучение на более специализированных наборах данных для корректировки параметров) с учётом приватности [11].

3.3. Graph Neural Networks (GNN)

Graph Neural Networks (GNN, графовые нейронные сети). В контексте работы с GNN разработаны специализированные стратегии удаления нужных узлов, рёбер и связанной информации без полного переобучения всей структуры графов. Примером такой стратегии может служить метод GNNDelete, который обеспечивает формальные гарантии удаления влияния заданных элементов, что особенно важно для распределённых и конфиденциальных графов данных. Обновление представления узлов в графовой нейронной сети происходит за счёт агрегации информации от соседних узлов и применения нелинейной функции активации [5]. При удалении узла v обновление состояния соседнего узла u на следующем слое $(l+1)$ можно формализовать выражением:

$$h_u^{(l+1)} = \sigma \cdot \left(\sum_{\omega \in N(u) \setminus \{v\}} W^{(l)} h_{\omega}^{(l)} \right), \quad (3)$$

где $h_u^{(l+1)}$ — представление узла u на слое $l+1$, $N(u)$ — множество соседей узла u , $W^{(l)}$ — параметр слоя, а σ — сигмоидальная или другая функция активации.

3.4. Spiking Neural Networks (SNN)

Spiking Neural Networks (SNN, спайковые нейронные сети). Опираясь на принципы биологии, спайковые нейронные сети реализуют машинное разобучение через механизмы си-

наптического обрезания (целенаправленное уменьшение/удаление весов синапсов, которые связывают нейроны) и адаптивную регуляцию порогов возбуждения нейронов. Это позволяет ограничивать влияние определённых сигналов на всю нейронную сеть и обеспечивать управляемое забывание на уровне связей между нейронами.

4. Критические вызовы и уязвимости

Машинное разобучение, несмотря на значительное развитие, сталкивается с целым рядом сложных вызовов и уязвимостей, которые требуют особого внимания на стадии проектирования и внедрения соответствующих решений. Одной из наиболее существенных проблем оказывается уязвимость разобучения к процедуре квантования моделей: оптимизация разрядности весов, используемая для повышения эффективности на ресурсно-ограниченных устройствах, способна частично или полностью восстановить влияние данных, ранее подверженных процедуре удаления. Экспериментальные данные свидетельствуют о том, что квантование может существенно ослаблять эффект разобучения, что обосновывает необходимость создания защищённых алгоритмов, устойчивых к подобным преобразованиям структуры модели.

Не менее актуальной задачей остаётся верификация корректности реализованного разобучения. Существующие методы проверки, как правило, оказываются недостаточно надёжными и не способны гарантировать полное устранение следов удалённых данных в параметрах модели и её выходах. Проверочные процедуры включают, в частности, использование закодированных триггеров для бэктур-верификации или сравнение результатов работы модели после удаления данных с результатами повторного обучения без них. Тем не менее такой подход не позволяет добиться абсолютной уверенности в корректности процесса разобучения [10].

Дополнительной угрозой выступают атаки на восстановление удалённой информации. Злоумышленники, обладая доступом к модели, могут реализовать Membership Inference Attack или Reconstruction Attack, а также инверсию, используя поведение модели либо её параметры для поэтапного восстановления исходных данных. Важно, что аналогичные риски наблюдаются даже для простых моделей, например линейной регрессии, подчёркивая необходимость комплексного анализа уязвимостей и разработки надёжных средств защиты конфиденциальных данных [6–8]. Существенным ограничением масштабирования методов точного машинного разобучения выступают требования к вычислительным ресурсам. Высокие затраты на вычисления затрудняют интеграцию эффективных алгоритмов разобучения в большие промышленные системы, вынуждая исследователей искать компромисс между точностью удаления, скоростью работы и доступными ресурсами вычислительных платформ.

Устранение перечисленных уязвимостей и проблем является необходимым условием для широкого и безопасного применения машинного разобучения, особенно в областях с высокими требованиями к защите данных и соответствуя нормативно-правовым актам различных государств и сообществ.

5. Инструменты оценки эффективности

Для объективной оценки качества методов машинного разобучения требуется комплексная система метрик и специализированных эталонных тестовых наборов (бенчмарков), позволяющих всесторонне проанализировать как эффективность удаления данных, так и сохранение полезности и адекватности модели после корректировки обучающих наборов. Основной набор показателей, используемых для оценки этих аспектов, приведён в табл. 2.

Комплексное использование перечисленных выше метрик и бенчмарков позволяет получить объективную оценку качества методов машинного разобучения и выбрать наиболее под-

Таблица 2

Основные инструменты оценки эффективности методов машинного разобучения

Метрика	Описание	Назначение и применение
Accuracy	Доля корректных предсказаний на тестовом наборе, исключающем удаленные данные	Оценка сохранения общей функциональности модели после разобучения
Precision и Recall	Precision — точность положительных предсказаний; Recall — полнота исключения данных	Характеризуют способность модели корректно «забывать» целевые примеры
ROC AUC	Площадь под ROC-кривой, отражающей соотношение чувствительности и специфичности	Анализ изменения дискриминационных свойств модели
MIA Accuracy	Метрика измеряет точность атак по определению принадлежности примеров в обучающем наборе (MIA)	Оценка степени утечки информации об удаленных примерах
Reconstruction Attack Resistance	Устойчивость к атакам восстановления данных по параметрам модели	Проверка защищенности модели после процедуры разобучения
Quantization Robustness	Изменение свойств модели после квантования параметров	Анализ устойчивости разобучения к сжатию модели
Time efficiency	Время выполнения процедуры разобучения	Оценка применимости метода в практических условиях
Memory Usage	Объём задействованной памяти	Анализ ресурсоемкости алгоритма
Scalability Index	Характеризует изменение вычислительных затрат при увеличении размера данных или модели	Оценка масштабируемости подхода

ходящие решения для конкретных прикладных задач с учётом требований к безопасности, эффективности и вычислительных ресурсов.

6. Рекомендации для внедрения и дальнейших исследований

На основании проведённого анализа современных методов машинного разобучения и актуальных вызовов, можно выделить несколько ключевых направлений как для практического внедрения этих технологий, так и для дальнейшего научного развития области.

Одним из важнейших трендов становится комбинирование точных и приближённых алгоритмов, что позволяет достичь оптимального баланса между гарантированной защитой приватности и необходимым уровнем вычислительной эффективности. Применение гибридных подходов способствует снижению риска восстановления удалённых данных и укрепляет устойчивость моделей к различным типам атак.

Особое значение приобретает формализация стандартов и совершенствование процедур верификации успешного разобучения. Принятие воспроизводимых и общедоступных методик проверки — таких как *reproducibility verification* или публичные бенчмарки — обеспечивает доверие и прозрачность результатов для пользователей, компаний и регуляторов. Отмечается тенденция к стандартизации процессов оценки моделей и внедрению открытых тестовых наборов.

Важным направлением становится повышение устойчивости реализуемых методов к различным видам оптимизации, включая квантование и компрессию, что особенно актуально для внедрения на периферийных устройствах и в массовых корпоративных продуктах. Создание устойчивых к потере разобучающего эффекта архитектур, способных противостоять изменению структуры параметров модели, — необходимое условие безопасности и сохранения приватности.

Дополнительным направлением для современных исследователей и инженеров является защита от атак. На этапе проектирования новых алгоритмов рекомендуется заранее интегрировать механизмы мониторинга и аудита устойчивости к потенциальным угрозам, что позволяет своевременно обнаруживать и реагировать на попытки восстановления удалённых данных.

Неотъемлемой задачей становится обеспечение интерпретируемости и прозрачности технологий машинного разобучения. Включение в рабочий процесс методов объяснения и явной фиксации удалённых компонентов модели помогает повысить доверие бизнеса и конечных пользователей, а также облегчит последующую интеграцию технологий в регулируемые отрасли.

На перспективу открывается путь к развитию расширенных профессиональных инструментов и фреймворков, поддерживающих гибкую интеграцию различных алгоритмов разобучения под архитектурные и предметные задачи — в том числе для работы с крупными языковыми моделями, графовыми и спайковыми нейросетями. Последовательное соблюдение этих принципов позволит увеличить качество, безопасность и прозрачность машинного разобучения, а также ускорит международную интеграцию и дальнейшее развитие исследований в этой области.

Заключение

Современные методы и архитектуры машинного разобучения позволяют удалять влияние отдельных обучающих данных из нейронных моделей без полного переобучения, что приобретает особую значимость в современных условиях. Внедрение гибридных и сертифицированных алгоритмов, развитие формальных процедур верификации, обеспечение устойчивости к оптимизации и атакам, а также контроль интегративности — всё это способствует эффективному и безопасному применению техники разобучения в практических задачах. Перспективы дальнейших исследований связаны с усилением приватности, расширением масштабируемости для самых крупных моделей и созданием открытых инструментов для стандартизации.

Литература

1. Cao Y., Yang J., Hsieh C. J. [et al.] Machine Unlearning // Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS'19). – Vancouver, Canada, 2019. – P. 1234–1245.
2. Bouroule L., Chandrasekaran V., Choquette-Choo C. A. [et al.] Machine Unlearning // Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21). – Montreal, Canada, 2021. – P. 1234–1240.
3. Hoang T. N., Rana S., Gupta S., Venkatesh S. Learn to Unlearn for Deep Neural Networks: Minimizing Unlearning Interference with Gradient Projection // Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV'24). – Waikoloa, Hawaii, USA, 2024. – P. 432–441.
4. Sekhari A., Acharya J., Kamath G., Suresh A. T. Remember What You Want to Forget: Algorithms for Machine Unlearning // Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS'22). – New Orleans, Louisiana, USA, 2022. – P. 18075–18088.

5. *Cheng J., Dasoulas G., He H., Agarwal C., Zitnik M.* GNNDelete: A General Strategy for Unlearning in Graph Neural Networks // Proceedings of the 11th International Conference on Learning Representations (ICLR'23). – Kigali, Rwanda, 2023. – P. 12126–12145.
6. *Shokri R., Stronati M., Song C., Shmatikov V.* Membership Inference Attacks Against Machine Learning Models // IEEE Symposium on Security and Privacy (S&P'17). – San Jose, California, USA, 2017. – P. 3–18.
7. *Salem A., Carlini N., Choo K. [et al.]* Data Set Inference and Reconstruction Attacks in Online Learning // USENIX Security Symposium. – Santa Clara, California, USA, 2020. – P. 1589–1606.
8. *Fredrikson M., Jha S., Ristenpart T.* Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures // Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15). – Denver, Colorado, USA, 2015. – P. 1322–1333.
9. *Agarwal S., Kumar A. [et al.]* ARCANE: An Efficient Architecture for Exact Machine Unlearning // Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI'22). – Vienna, Austria, 2022. – P. 2956–2962.
10. *Pawelczyk M., Casper S., So K., Hadwiger M., Ong C.* Machine Unlearning Fails to Remove Data from Trained Models // Proceedings of the 13th International Conference on Learning Representations (ICLR'25). – Singapore, 2025. – P. 1890–1905.
11. *Liu X., Cheng P., Fang B. [et al.]* Large Language Model Unlearning via Embedding-Corrupted Prompts // NeurIPS Workshop. – New Orleans, Louisiana, USA, 2024. – P. 1–15.

ОБУЧЕНИЕ МАЛЫХ ЯЗЫКОВЫХ МОДЕЛЕЙ С ЦЕЛЬЮ АНАЛИЗА ПОЛЬЗОВАТЕЛЬСКОЙ ОБРАТНОЙ СВЯЗИ

Н. А. Экерт, И. Е. Воронина

Воронежский государственный университет

Аннотация. Рассматривается актуальная проблема извлечения смыслов из пользовательской обратной связи с целью улучшения существующего программного продукта. В качестве предмета исследования выбраны малые языковые модели (Small Language Models), используемые для задач краткого изложения (суммаризации) и извлечения смыслов из текста, а также метод их обучения Fine-tuning. Приведены результаты обучения, анализ точности, сравнительные таблицы и графическая интерпретация различий до и после обучения. Для повышения качества моделей применён метод Supervised Fine-Tuning с использованием QLoRA INT4. Новизна результата заключается в получении адаптированных к решению задач в рамках выбранной предметной области моделей и их сравнении.

Ключевые слова: малые языковые модели, Small Language Models, SLM, оценка отзывов, Llama, DeepSeek, Qwen, сравнение языковых моделей, извлечение смыслов.

Введение

Современные программные продукты развиваются в условиях высокой конкуренции, где способность своевременно выявлять и устранять недостатки является ключевым фактором успеха. Пользовательские отзывы выступают наиболее ценным источником информации о реальных сценариях использования ПО, проблемах и пожеланиях [1].

Ручная обработка отзывов становится затруднительной при росте аудитории, что обуславливает необходимость автоматизации. Языковые модели могут решить задачу формирования задач для бэклога разработки на основе отзывов, однако качество результатов существенно зависит от адаптации модели под конкретную предметную область [2].

Цель работы — исследовать эффективность применения Fine-tuning малых моделей в условиях ограниченных вычислительных ресурсов и сравнить качество их работы до и после обучения.

1. Формирование обучающего датасета

Корпус данных представляет собой набор из 3000 реальных пользовательских отзывов на программный продукт. Каждый отзыв содержит до 1000 символов, включает субъективные оценки, упоминания функциональных возможностей, описания ошибок, пожелания по развитию и общий пользовательский опыт. Данные были предварительно анонимизированы: удалены персональные данные, сетевые идентификаторы, имена пользователей и любые сведения, относящиеся к конфиденциальной информации [3].

Структура отзывов характеризуется высокой разнородностью: встречаются фрагменты разговорной речи, англоязычные термины, эмоциональные формулировки, описания сценариев использования и сравнительные оценки между версиями продукта. Это накладывает ограничения на возможности малых языковых моделей, требуя тщательной подготовки обучающего материала.

Для корректной работы алгоритмов обучения все отзывы были приведены к унифицированному формату: лишние символы удалены, текст нормализован, выполнена токенизация. Дополнительная нормализация (лемматизация, исправление опечаток) для этапа fine-tuning

не проводилась, поскольку обучение малых моделей предполагает работу с естественным языком без глубоких преобразований исходного текста [4].

Корпус был разделён на три выборки:

- обучающая — 2700 отзывов (90 %);
- валидационная — 150 отзывов (5 %);
- тестовая — 150 отзывов (5 %).

Такое распределение позволило одновременно обеспечить достаточный объём данных для обучения и сохранить объективность итоговой оценки.

В исследовании используется подход SFT — Supervised Fine-Tuning [5-6], при котором модель обучается на паре «инструкция – корректный ответ». Это один из наиболее распространённых и эффективных подходов тонкой настройки SLM для решения конкретной задачи. В отличие от RAG (Retrieval-Augmented Generation) [7], SFT формирует новое поведение модели за счёт обновления её параметров (в нашем случае — через адаптеры LoRA/QLoRA) [8-9].

Для каждой записи создавался единый формат данных, включающий инструкцию, входные данные и целевой ответ. Формат целевого ответа представлен в листинге 1. Выбранный формат обеспечивает совместимость с большинством открытых библиотек fine-tuning, таких как HuggingFace, LlamaFactory, unsloth, и полностью соответствует SFT-подходу.

Ручная разметка выполнялась с помощью привлечения эксперта, который действовал по следующему алгоритму. Он считывал текст отзыва и определял ключевую проблему или пожелание пользователя. Формулировал задачу кратко, не более чем одной строкой, согласно формату результата user story-like. Это краткий однозначный формат в виде сформулированной задачи для разработчиков по улучшению ПО, например, «улучшить скорость запуска приложения на устройствах с низкой производительностью» или «исправить ошибку авторизации при переходе между экранами». Далее эксперт исключал дублирование и содержание, не относящееся к функционалу, проверял соответствие результата заранее выбранному формату.

В результате было размечено 100 примеров отзывов, которые использовались:

- как основа для SFT-датасета;
- как образцы для авторазметки оставшейся части корпуса (псевдо-labeling);
- как эталон для оценки качества моделей.

При этом 3000 отзывов не были синтетически расширены. В будущем планируется исследовать влияние расширения обучающего датасета с помощью генерации синтетических отзывов для улучшения качества обучения.

2. Методика обучения Fine-tuning

В исследовании рассматриваются три компактные языковые модели: Llama 3.2 1B, DeepSeek-R1 1.5B и Qwen 3 1.7B. Все они относятся к классу малых моделей (до 2 млрд параметров), что делает возможным локальный Fine-tuning на пользовательском оборудовании. Малые модели отличаются высокой скоростью работы и низкими требованиями к памяти, однако без обучения их качество в решении прикладных задач ограничено, в том числе и в задаче анализа пользовательских отзывов. Малые модели хуже работают с задачами смыслового обобщения и извлечения скрытых контекстов — именно поэтому Fine-tuning становится обязательным условием.

До обучения все приведенные модели демонстрируют следующие проблемы: недостаточная точность в определении основной проблемы; подмена запроса *overly-general* (черезчур базовыми) ответами; склонность к галлюцинациям; неустойчивая формулировка задач. Предполагается, что Fine-tuning должен компенсировать эти недостатки за счёт специализации моделей на узкоспециализированной задаче.

В исследовании рассматриваются 3 целочисленных типа квантования со снижением разрядности: INT8, INT4, NF4. Для сравнения также приводится FP16. INT8 позволяет минимизировать потерю в качестве модели, при этом сильнее нагружает VRAM. INT4 позволяет найти компромисс между скоростью и качеством. NF4 (Normalized Float 4) — более гибкая форма квантования, обеспечивающая высокую точность, но требующая поддержки инфраструктуры QLoRA.

QLoRA — это метод обучения модели, где веса основной модели заморожены и хранятся в INT4, а вычисления умножения выполняются в NF4. Обучаемыми при этом становятся только низкоранговые матрицы LoRA. Это дает преимущество в виде отсутствия значимых потерь качества при SFT, ускоренной загрузки и инференса модели.

В табл. 1 и на рис. 1 приведено распределение и визуализация нагрузки на VRAM при различных уровнях квантования.

Таблица 1

Нагрузка на VRAM при различных уровнях квантования

Семейство	Модель	Объем VRAM, Gb			
		FP16	INT8	INT4	NF4
Llama	3.2 (1b)	2.0	1.0	0.6	0.7
DeepSeek	R1 (1.5b)	3.0	1.5	0.9	1.0
Qwen	3 (1.7b)	3.5	1.7	1.1	1.2

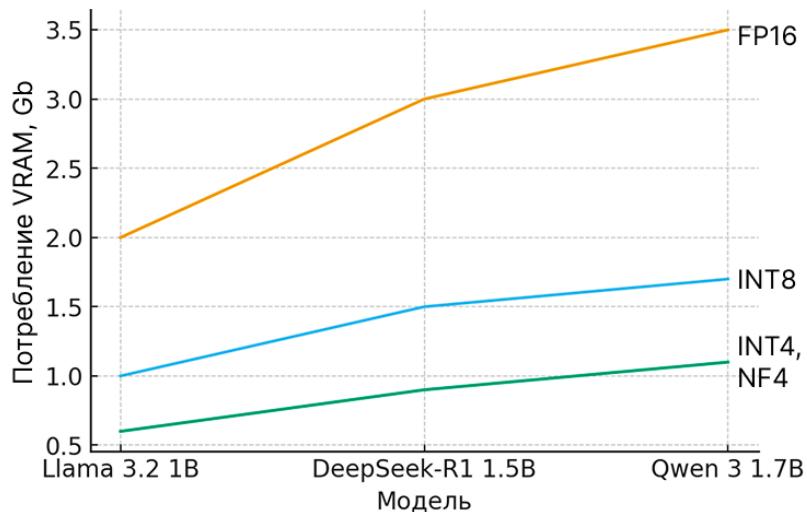


Рис. 1. Визуализация нагрузки на VRAM при различных уровнях квантования

Процесс дообучения малых языковых моделей проводился по методике Supervised Fine-Tuning (SFT) с использованием квантования QLoRA INT4, что позволило существенно снизить требования к видеопамяти без значимой потери качества.

В отличие от Reinforcement Learning из человеческой обратной связи (RLHF), SFT не требует сложной цепочки обратной связи, обеспечивает предсказуемое поведение модели, подходит для решения узкоспециализированных задач и позволяет относительно малыми усилиями адаптировать модель к новой предметной области. Данный подход особенно эффективен, когда необходимо фильтровать шум и выделять основную мысль из текста.

Fine-tuning выполнялся на пользовательском ПК следующей конфигурации:

- CPU: AMD Ryzen 7 5700X
- RAM: 64 GB
- GPU: Nvidia RTX 4070 ti Super (16 GB VRAM)
- OS: Ubuntu 24.04

Для всех моделей использовались одинаковые параметры, чтобы обеспечить корректное сравнение результатов. Параметры приведены в табл. 2.

Таблица 2

Параметры обучения моделей

№	Параметр	Значение
1	Число эпох	3
2	Batch size	128 (эффективный, с градиентным накоплением)
3	Learning rate	2e-4
4	LoRA rank	16
5	LoRA alpha	32
6	LoRA dropout	0,05
7	Warmup steps	100
8	Max seq length	1024 токенов
9	Тип квантования	QLoRA INT4
10	Тип обучения	SFT

3. Оценка полученных результатов

Итоговое время обучения составило около 3 часов, детализация по времени обучения для каждой модели представлена в табл. 3.

Таблица 3

Детализация времени обучения для каждой модели

№	Название модели	Время на одну эпоху, мин	Общее время обучения, мин
1	Llama 3.2 (1b)	15	45
2	DeepSeek R1 (1.5b)	19	58
3	Qwen 3 (1.7b)	24	73

По окончании обучения было проведено сравнение качества решения задачи оценки отзывов пользователей моделями до применения Fine-tuning и после применения Fine-tuning. Для тестирования использовались 150 отзывов из тестового набора, 100 вручную размеченных эталонных примера. Метрика — точность совпадения формулировки задачи с экспертной разметкой в %. Совпадение учитывает смысл, формулировку и корректность выделенной проблемы. Результаты метрики точности при сравнении моделей приведены в табл. 4.

Анализируя полученные результаты, можно сделать вывод о том, что модели существенно улучшили свою точность после непродолжительного этапа обучения. При этом, первоначальные моделиправлялись с задачами крайне плохо. Это связано с тем, что у них значительно меньшее количество параметров, по сравнению с большими моделями. Прирост в точности составил от 33 до 35 %, при том, что датасет искусственно не расширялся и применялись стандартные 3 эпохи обучения.

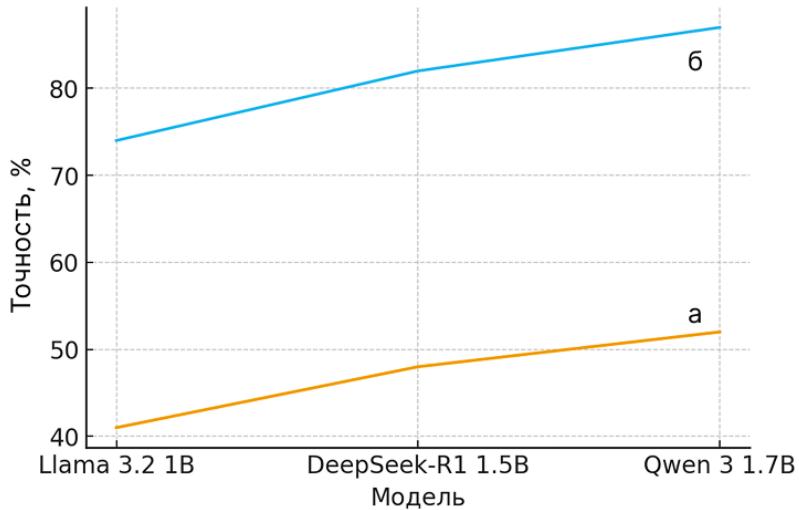
Таблица 4

Результаты сравнения моделей

№	Название модели	Точность модели до Fine-tuning, %	Точность модели после Fine-tuning, %	Прирост, %
1	Llama 3.2 (1b)	41	74	33
2	DeepSeek R1 (1.5b)	48	82	34
3	Qwen 3 (1.7b)	52	87	35

После обучения модели перестали галлюцинировать, начали использовать краткую формулировку задач и намного точнее ориентироваться в предметной области. Однако, модели все еще испытывают трудности при анализе сложных отзывов с множественными проблемами.

Разный прирост точности моделей можно списать на погрешность, однако очевидным является улучшение показателей точности при большем количестве параметров обучаемой модели. Так, модель с большим изначальным количеством параметров во всех тестах превзошла другие модели и показала больший прирост при обучении (см. рис. 2).



а) до обучения модели; б) после обучения модели;
Рис. 2. Результаты метрики точности при сравнении моделей

Заключение

Обозначена необходимость использования малых языковых моделей для решения задачи изложения (суммаризации) и извлечения смыслов из текста. В качестве инструмента решения выбраны малые языковые модели (Small Language Model). Проведено обучение трех представленных на рынке SLM с применением метода Supervised Fine-Tuning с использованием QLoRA INT4. Приведены результаты обучения, анализ точности, сравнительные таблицы и графическая интерпретация различий до и после обучения.

Литература

1. Экерт Н. А. Алгоритм именования кластеров при автоматизированном формировании критериев оценивания программного продукта / Н. А. Экерт, И. Е. Воронина // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2025. – № 2. – С. 78–88.
2. Экерт Н. А. Использование больших языковых моделей для оценки отзывов на программные продукты / Н. А. Экерт, И. Е. Воронина // Сборник трудов Международной научно-технической конференции «Актуальные проблемы прикладной математики, информатики и механики». Секция: Искусственный интеллект, робототехнические системы, машинное обучение. – 2024 – С. 365–372.
3. Ингерсолл Г. С. Обработка неструктурированных текстов / Г. С. Ингерсолл, Т. С. Мортон, Э. Л. Фэррис ; Москва: ДМК Пресс, 2015. – 414 с.
4. Хобсон Л. Обработка естественного языка в действии / Л. Хобсон, Х. Ханнес, Х. Коул ; СПб. : Питер, 2020. – 576 с.

5. *Xiaoxuan Liao*. A Fine-Tuning Approach for T5 Using Knowledge Graphs to Address Complex Tasks. / Xiaoxuan Liao, Binrong Zhu, Jacky He, Guiran Liu, Hongye Zheng, Jia Gao. – 2025. – DOI 10.48550/arXiv.2502.16484.

6. *Zorik Gekhman*. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? / Zorik Gekhman, Gal Yona, Roee Aharoni, Matan Eyal, Amir Feder, Roi Reichart, Jonathan Herzig. – 2024. – DOI 10.48550/arXiv.2405.05904.

7. *Angels Balaguer*. RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture. / Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O. Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, Ranveer Chandra. – 2024. – DOI 10.48550/arXiv.2401.08406.

8. *Jiacheng Hu*. Optimizing Large Language Models with an Enhanced LoRA Fine-Tuning Algorithm for Efficiency and Robustness in NLP Tasks. / Jiacheng Hu, Xiaoxuan Liao, Jia Gao, Zhen Qi, Hongye Zheng, Chihang Wang. – 2024. – DOI 10.48550/arXiv.2412.18729.

9. *Dipankar Sarkar*. Viz: A QLoRA-based Copyright Marketplace for Legally Compliant Generative AI. – 2023. – DOI 10.48550/arXiv.2401.00503.

О КОМПЬЮТЕРНОЙ МОДЕЛИ СИСТЕМЫ УПРАВЛЕНИЯ ПЕРЕВЕРНУТЫМ МАЯТНИКОМ ДЛЯ МИКРОКОНТРОЛЛЕРА STM32F103 НА ОСНОВЕ МЕТОДА PILCO

А. Ю. Яковлев, А. Е. Жуков, И. О. Мягкова

Воронежский государственный университет

Аннотация. Работа посвящена компьютерной реализации алгоритма управления перевернутым маятником, основанном на методе машинного обучения PILCO [4]. Разработано программное обеспечение, реализующее модель и алгоритм управления, выполнены компьютерные эксперименты. Результаты демонстрируют высокую эффективность примененного подхода для стабилизации перевернутого маятника в вертикальном положении. Исследование ориентировано на процесс внедрения подобных систем управления в робототехнические комплексы и беспилотные летательные аппараты.

Ключевые слова: регрессионное моделирование, гауссовские процессы [1, 3], перевернутый маятник, робототехнический комплекс, алгоритм PILCO [2, 4], байесовский вывод, программное обеспечение, управление, угловая скорость, мехатроника, машинное обучение, компьютерное моделирование.

Введение

Разработка и внедрение систем управления на основе методов машинного обучения для мехатронных и робототехнических устройств является актуальной задачей современной мехатроники и робототехники. Одной из таких задач является построение системы управления для балансирующего робота (перевернутого маятника). Балансирующий робот — автономное устройство, оснащенное микроконтроллером (например, STM32F103) и набором инерциальных датчиков, гироскоп и акселерометр (например, в модуле MPU6000). Программная часть системы управления балансирующим роботом обрабатывает данные, полученные от инерциальных датчиков, и вычисляет управляющие сигналы для приводов с целью стабилизации его корпуса в вертикальном положении. Для снижения нагрузки на реальное оборудование целесообразно создать компьютерную модель системы на базе микроконтроллера, которые будет управлять реальным устройством (в данном случае — STM32F103). В качестве метода построения системы управления был выбран метод машинного обучения, основанный на алгоритме PILCO [2, 4]. Полученная компьютерная модель позволит отрабатывать алгоритмы, реализуемые на бортовой части робототехнического комплекса. Компьютерная модель включает в себя также программный комплекс для персонального компьютера, предназначенный для проведения вычислений, визуализации процессов, приёма и передачи данных. Бортовая часть осуществляет вычисление управляющего сигнала по данным компьютерной модели и передает необходимый набор параметров на персональный компьютер.

1. Описание математической модели перевернутого маятника.

Рассмотрим систему, состоящую из подвижного основания тележки и шарнирно прикрепленного к ней стержня-маятника [2]. Тележка может двигаться вдоль горизонтальной плоскости вправо и влево. Трение о поверхность при движении тележки учитывается. Движение тележки возникает в результате прилагаемого управляющего усилия *u* и направленного вдоль горизонтальной прямой. Задача для системы управления заключается в непрерывном приложении такого усилия, которое позволит системе удержать стержень в вертикальном положении, а тележку в начале горизонтальной координатной оси.

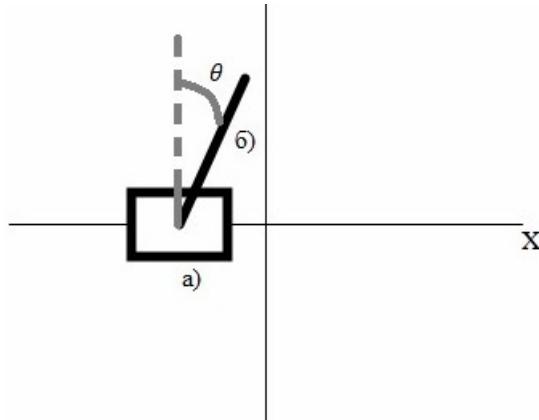


Рис. 1. Модель перевернутого маятника: а) тележка, б) стержень

Перевернутый маятник, изображенный на рис. 1, состоит из тележки массой m_1 и прикрепленного к ней маятника массой m_2 и длиной l , который свободно качается в плоскости. Тележка может двигаться горизонтально под действием приложенной внешней силы u и параметра b , который описывает трение между тележкой и поверхностью.

Положение тележки вдоль направляющей обозначается как x_1 . Координаты x_2 и y_2 обозначают положение центра тяжести маятника

$$x_2 = x_1 + \frac{1}{2}l \sin \theta, \quad (1)$$

$$y_2 = -\frac{1}{2}l \cos \theta. \quad (2)$$

Для вывода уравнения движения запишем функцию Лагранжа системы L , которая представляет собой разность между кинетической энергией T и потенциальной энергией Π системы и в данном случае имеет вид

$$L = T - \Pi = \frac{1}{2}(m_1 + m_2)\dot{x}_1^2 + \frac{1}{6}m_2l^2\dot{\theta}^2 + \frac{1}{2}m_2l(\dot{x}_1\dot{\theta} + g)\cos \theta, \quad (3)$$

где g — ускорение свободного падения.

Уравнения движения в общем случае могут быть получены из набора уравнений

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i, \quad (4)$$

где $i = 1, 2$, Q_i — обобщенные силы, а q_i и \dot{q}_i — переменные состояния системы. В нашем случае, система уравнений представлена соотношениями (5)–(6)

$$(m_1 + m_2)\ddot{x}_1 + \frac{1}{2}m_2l(\ddot{\theta} \cos \theta - \dot{\theta} \sin \theta) = u - b\dot{x}_1, \quad (5)$$

$$2l\ddot{\theta} + 3\ddot{x}_1 \cos \theta + 3g \sin \theta = 0. \quad (6)$$

Таким образом состояние системы в каждый момент времени описывается следующими параметрами

$$\mathbf{x} = [x_1, \dot{x}_1, \dot{\theta}, \theta]^T. \quad (7)$$

Математическую модель перевернутого маятника представляют выражения (5)–(7), которые будут применены для численного моделирования.

2. Описание алгоритма управления перевернутым маятником

Для удержания маятника в вертикальном положении применяется алгоритм PILCO [2, 4]. Цель работы алгоритма — определить параметры функции π , которая минимизирует ожидаемые долгосрочные затраты на последовательности движений системы в течение конечного числа T временных шагов [2]. Функцию $\pi(\mathbf{x})$ называют политикой, она определяет величину управляющего сигнала в зависимости от вектора \mathbf{x} текущего положения системы. Вычисление ожидаемых долгосрочных затрат соответствует вычислению функции значений V

$$V(x_0) = \sum_{t=0}^T E_{x_t} [c(\mathbf{x}_t)], \quad (8)$$

где функция $V(x_0)$ вычисляет затраты для состояния системы x_0 с учетом текущей политики π , E — математическое ожидание функции ценности $c(\mathbf{x})$. Функция ценности для данного случая, записанная как

$$c(\mathbf{x}) = 1 - \exp\left(-\frac{1}{2a^2} d(\mathbf{x}, \mathbf{x}_n)^2\right) \quad (9)$$

характеризует геометрическое отклонение состояния системы \mathbf{x} от целевого состояния \mathbf{x}_n [2]. Под целевым состоянием системы будем понимать вертикальное положение маятника и положение тележки в начале отсчета

$$\mathbf{x} = [0, 0, 0, 0]^T.$$

Рассмотрим алгоритм PILCO более подробно [2, 4].

На начальном этапе выполняется подача случайных управляющих сигналов на тележку с фиксацией данных о меняющемся положении системы. По сформированному набору данных в виде кортежей $\mathbf{x} = [x_1, \dot{x}_1, \dot{\theta}, \theta]^T$ вычисляем коэффициенты функции управления — политики, которая в данном случае имеет вид

$$\pi(\mathbf{x}) = \Psi \mathbf{x} + \nu, \quad (10)$$

где Ψ — матрица размером 4×4 элементов, ν — вектор размером 4 элемента.

Далее циклическое повторение действий.

Начало цикла 1.

Применяем политику (10) маятнику и фиксируем новый набор данных в виде кортежей $\mathbf{x} = [x_1, \dot{x}_1, \dot{\theta}, \theta]^T$.

Рассчитываем параметры регрессионной модели динамики системы, основанной на обновленном наборе данных [2].

Начало цикла 2.

Используя обновленную регрессионную модель динамики системы [2], моделируем ее движение с учетом обновленной политики управления на T шагов по времени вперед. В процессе моделирования один шаг соответствует 0,05 с. В процессе моделирования вычисляем ожидаемые затраты (8).

Используя полученные ожидаемые значения затрат, улучшаем политику [2, 4].

Конец цикла 2.

Конец цикла 1.

3. Компьютерная модель перевернутого маятника и алгоритм ее работы

Компьютерная модель представляет собой программный комплекс, который включает в себя программу для ПК и программу для микроконтроллера STM32F103.

Программа для ПК имеет интерфейс взаимодействия с пользователем, окно для визуализации процесса движения модели и обеспечивает необходимые расчеты по алгоритму PILCO,

передачу и сохранение данных. Программа для ПК написана на языке JAVA в среде Processing [5]. В состав программы для ПК входит девять модулей, при этом ее общий размер составляет 309 Кб.

Программа для МК STM32F103 написана на языке С в среде Keil v5. В ее функционал входит прием команд на выполнение расчета от ПК, проведение численного расчета по соотношения динамики перевернутого маятника в режиме реального времени, передачу полученных данных о состоянии системы после расчета на ПК по интерфейсу UART.

Рассмотрим основные модули программы для ПК.

Definitions.pde — реализует объявление классов для описания регрессионной модели динамики [1], оптимизации гиперпараметров регрессионной модели, процесса прогнозирования и другое;

Matrixs.pde — содержит обширный набор функций по работе с векторами, матрицами и многомерными массивами;

Base.pde — реализует процесс выполнения эпизода обучения с фиксацией данных для текущей политики (10). Содержит функцию вычисления величины ценности (9) для данного состояния системы.

Control.pde — содержит функции реализующие политику управления (10).

Gp.pde — реализует регрессионную модель динамики системы на основе гауссовских процессов.

Util — содержит функции поиска максимума методом BFGS [2].

Программа для МК STM32F103 содержит следующие файлы.

Main.h/c — файл с основным циклом программы, который обеспечивает расчеты компьютерной модели с частотой 20 Гц.

System.h/c — реализует набор необходимых системных функций, например функций по работе с временными отрезками.

Usart.h/c — содержит набор функций по работе с интерфейсом UART для приема и передачи данных расчетов и различных команд при работе с ПК.

Model.h/c — реализует компьютерную модель перевернутого маятника и численное интегрирование методом Рунге — Кутты 2 порядка.

Matrix.h/c — объединяет набор функций для математической обработки векторов и матриц.

PILCO.h/c — содержит набор функций для вычисления прогноза управляющего сигнала для текущего положения системы с учетом переданных от ПК параметров политики управления.

Рассмотрим более подробно алгоритм работы программного комплекса.

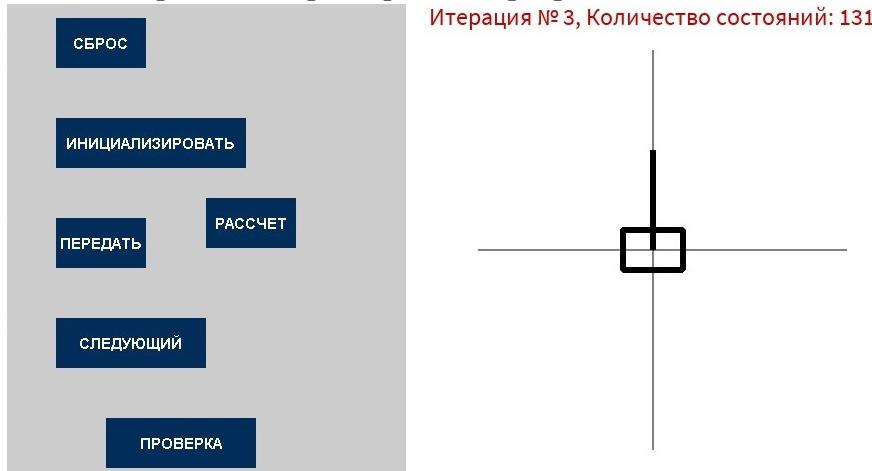


Рис. 2. Основное окно программы для ПК

На рис. 2 представлено рабочее окно программы для ПК. С правой стороны отображается модель перевернутого маятника с демонстрацией его движения в соответствующих режимах работы программы.

После запуска программы для ПК пользователь выбирает режим ИНИЦИАЛИЗИРОВАТЬ. Соответствующая команда передается по интерфейсу UART на МК STM32F103, где запускается процесс подачи случайных управляющих сигналов на компьютерную модель. При этом фиксируются состояния системы с частотой 20 Гц. Движение ограничивается заданным временным интервалом T или выходом стержня за предельные углы положения, которые исходно задаются. Далее происходит автоматическая передача сформированного массива данных на ПК.

Теперь пользователь может выбрать режим РАСЧЕТ. В рамках этого режима вычисляются параметры политики по текущему массиву данных согласно алгоритму выше.

После появления сообщения о завершении расчета можно произвести передачу параметров на микроконтроллер. Для этого необходимо выбрать режим ПЕРЕДАТЬ.

Далее пользователь выбирает режим СЛЕДУЮЩИЙ — производится выполнение очередного эпизода обучения с дополнением массива данных под управлением политики с обновленными параметрами. При этом движение ограничивается заданным временным интервалом T или выходом стержня за предельные углы положения.

Для запуска процесса снова необходимо выбрать режим СБРОС. Можно запустить систему управления в режиме ПРОВЕРКА. В этом случае, движение стержня под действием текущей политики ограничиваются только предельными углами положения.

4. Результаты работы компьютерной модели

На рис. 3 приведены фазы движения системы после обучения политики. Массив обучающих данных составил 121 кортеж вида $\mathbf{x} = [x_1, \dot{x}_1, \dot{\theta}, \theta]^T$. Количество обучающих эпизодов — три.

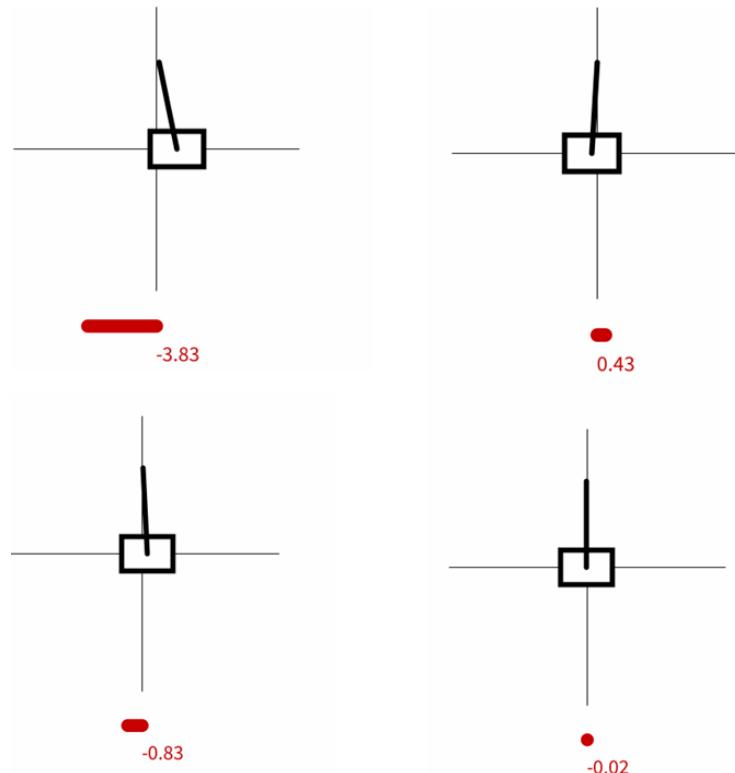


Рис. 3. Демонстрация процесса стабилизации перевернутого маятника в вертикальном положении

Заключение

В работе рассмотрена компьютерная модель для микроконтроллера STM32F103 перевернутого маятника и его системы управления. Система управления построена на основе алгоритма машинного обучения PILCO. Программная реализация компьютерной модели включает в себя программу для персонального компьютера и программу для микроконтроллера STM32F103. Результаты, полученные в работе, можно использовать для создания системы управления на основе алгоритма PILCO для широкого класса малоразмерных робототехнических комплексов и беспилотных летательных аппаратов.

Литература

1. *Бишоп Кристофер М.* Распознавание образов и машинное обучение. : Пер. с англ. – СПб. : ООО «Диалектика», 2020. — 960 с.
2. *Deisenroth M. P.* Efficient Reinforcement Learning using Gaussian Processes. KITScintificPublishing, 2011.
3. *Измаилов П. А.* Алгоритмы обучения гауссовских процессов для больших объемов данных. – Москва, 2017.
4. PILCO: A Model-Based and Data-Efficient Approach to Policy Search, 2011. – С. 1–4.
5. Processing : официальный сайт. – URL: <https://processing.org> (дата обращения: 20.11.2025).