

Министерство науки и высшего образования РФ

Национальный исследовательский  
Нижегородский государственный университет им. Н.И. Лобачевского

**МАТЕМАТИКА И ИТ — ВМЕСТЕ В ЦИФРОВОЕ БУДУЩЕЕ**

*Сборник трудов V Молодежной школы*

*Нижний Новгород, 20–24 апреля 2026 г.*

Нижний Новгород  
ННГУ им. Н.И. Лобачевского

2026

УДК 004.942(06)+519.876.5(06)  
ББК 22.18я43  
М34

**М34 Математика и ИТ – вместе в цифровое будущее.** Сборник трудов V Молодежной школы (Н. Новгород, 20–24 апреля 2026 г.) / Под ред. К. А. Баркалова, Т. Г. Смирновой. – Нижний Новгород: Нижегородский госуниверситет им. Н.И. Лобачевского, 2026. – 72 с.

ISBN 978-5-908049-69-6

Сборник трудов участников конкурса научно-исследовательских проектов по математике и информационным технологиям V Молодежной школы «Математика и ИТ – вместе в цифровое будущее», которая состоялась 20–24 апреля 2026 г. на базе Нижегородского государственного университета им. Н.И. Лобачевского, содержит тезисы докладов и короткие статьи, тематика которых посвящена информационным и сквозным технологиям цифровой экономики, математическому моделированию сложных процессов и явлений, численным методам их исследования.

ISBN 978-5-908049-69-6

УДК 004.942(06)+519.876.5(06)  
ББК 22.18я43

© Коллектив авторов, 2026  
© ННГУ им. Н.И. Лобачевского, 2026

## Предисловие

20–24 апреля 2026 г. в Институте информационных технологий, математики и механики ННГУ им. Н. И. Лобачевского состоялась V Молодежная школа «Математика и ИТ – вместе в цифровое будущее». Традиционно школа прошла в online-формате. Участники школы встретились с учеными Университета Лобачевского и ведущими практиками из российских и международных IT-компаний: Сбер, 1С, ООО «ПИРФ», Неофлекс Консалтинг, IT-академии Lad, IT-Холдинга «Г1».

Программа молодежной школы включала следующие лекции:

- Денис Сытый (эксперт по 1С): Современные технологии цифровизации бизнеса;
- Владимир Башев (технический директор ПИРФ): Эволюция технологий и инструментов создания ПО;
- Валентин Волокитин (ННГУ им. Н.И. Лобачевского, старший преподаватель ИИТММ): Векторные математические функции для процессоров архитектуры RISC-V;
- Мария Воронова (главный инженер разработки Сбер): Зачем разработчику знать математику;
- Алексей Усков (директор IT-академии Lad): Искусственный интеллект пришёл вам на смену — и это стало вашим ключевым преимуществом;
- Артем Савенков (IT-Холдинг «Г1»): SaaS-архитектура на примере Сферы: Как строятся масштабируемые облачные решения;
- Игорь Позументов (директор филиала Неофлекс Консалтинг): ТРИЗ и история изобретений.

В рамках работы V Молодежной школы проводился конкурс научно-исследовательских проектов по математике и информационным технологиям, тематика которых охватывала следующие направления:

- суперкомпьютерные технологии;
- искусственный интеллект;
- машинное обучение и анализ данных;
- робототехника;
- интернет вещей;
- большие данные;
- математическое моделирование;
- численные методы;
- алгебра, геометрия, теория чисел;
- теория вероятностей и приложения;
- дискретная математика;
- дифференциальные уравнения.

Студенты из разных городов России рассказали о результатах своих исследований и разработок. В соответствии с тематикой мероприятия были представлены как чисто математические работы, так и проекты в сфере IT.

Победителями конкурса стали два проекта:

- Исследование архитектурных особенностей нейронных сетей для поиска сердечных заболеваний путём анализа ландшафтов функции потерь (представлен Калашниковым Львом, Нижегородский государственный университет им. Н.И. Лобачевского);
- Адаптивные тензорные сети для анализа квантовой запутанности в гибридных квантово-классических моделях (представлен Дятловым Владиславом и Филипповым Алексеем, Ульяновский государственный университет).

# Адаптивные тензорные сети для анализа квантовой запутанности в гибридных квантово-классических моделях

*В.Д. Дятлов, А.В. Филиппов*

*Ульяновский государственный университет*

В работе исследуется влияние параметров матричного продуктного состояния (MPS) – bond-размерности  $\chi$ , порога усечения  $\varepsilon$  и числа сэмплов – на точность и сходимость гибридных квантово-классических моделей. На основе разработанного веб-фреймворка на Node.js/WebAssembly проведён систематический анализ компромисса между вычислительной сложностью и точностью симуляции. Показано, что значения  $\chi = 16-32$  и  $\varepsilon = 10^{-6}$  обеспечивают ошибку  $\delta < 10^{-6}$  и достаточны для задач классификации. Впервые количественно исследована корреляция между уровнем квантовой запутанности (concurrence, энтропия Шмидта) и точностью моделей: коэффициент корреляции  $\rho \approx 0.3$ . Результаты могут быть использованы для оптимизации гибридных алгоритмов на NISQ-устройствах.

*Ключевые слова:* квантовая запутанность, MPS, тензорные сети, гибридное машинное обучение, concurrence, энтропия Шмидта.

## 1. Введение

Гибридные квантово-классические модели на основе вариационных квантовых схем (VQC) — один из наиболее реалистичных путей использования квантовых вычислений в ближайшей перспективе [1–4, 6]. Однако практическая реализация таких алгоритмов наталкивается на две фундаментальные трудности. Во-первых, симуляция квантовых схем полным вектором состояния требует экспоненциальных ресурсов  $O(2^n)$ , что ограничивает масштабируемость. Альтернативой служат матричные продуктные состояния (MPS), снижающие сложность до  $O(n \cdot d \cdot \chi^3)$  [9], но выбор ключевых гиперпараметров — bond-размерности  $\chi$  и порога усечения  $\varepsilon$  — остаётся эвристическим, а их систематическое влияние на точность гибридного обучения не изучалось. Во-вторых, остаётся открытым вопрос, действительно ли квантовая запутанность, генерируемая в VQC, даёт измеримое преимущество в задачах классификации, и можно ли количественно связать её метрики с итоговой точностью модели.

Научная новизна данной работы заключается в двух аспектах:

- параметрическом анализе MPS-симуляции ( $\chi$ ,  $\varepsilon$ , число сэмплов) для определения баланса между вычислительными затратами и точностью обучения;
- первом количественном исследовании корреляции между уровнем запутанности (concurrence, энтропия Шмидта) и эффективностью гибридных моделей на задачах бинарной классификации.

Исследование выполнено на базе ранее разработанного веб-фреймворка [1], обеспечивающего MPS-симуляцию, гибридное обратное распространение и встроенный мониторинг запутанности. Цель — экспериментально определить диапазоны  $\chi$ ,  $\varepsilon$  и числа сэмплов, при которых ошибка симуляции не превышает  $10^{-6}$ , а вычислительные затраты остаются приемлемыми, а также оценить силу связи между квантовой запутанностью и точностью классификации.

## 2. Методы и материалы

### 2.1. Гибридная квантово-классическая модель

В работе рассматривается гибридная модель вида

$$f(x, \theta, \phi) = f_{\text{class}}(f_{\text{quant}}(x, \theta), \phi),$$

где  $x \in \mathbb{R}^d$  – входной вектор признаков,  $\theta \in \mathbb{R}^p$  – параметры квантовой схемы,  $\phi \in \mathbb{R}^q$  – параметры классической нейронной сети. Квантовый слой  $f_{\text{quant}}: \mathbb{R}^d \times \mathbb{R}^p \rightarrow \mathbb{R}^m$  – реализован на основе вариационной квантовой схемы (Variational Quantum Circuit, VQC) [4], а классический слой  $f_{\text{class}}: \mathbb{R}^m \times \mathbb{R}^q \rightarrow \mathbb{R}^k$  – полносвязная нейронная сеть с функцией активации ReLU на скрытом слое и softmax на выходе.

#### 2.1.1. Угловое кодирование данных

Для преобразования классических признаков в квантовое состояние используется угловое кодирование (angle encoding) [11]. Каждый признак  $x_i$  (после нормализации к интервалу  $[0, 2\pi]$ ) кодируется вращением одного кубита вокруг оси  $Y$ :

$$E(x_i): |0\rangle \mapsto R_y(x_i)|0\rangle = \cos \frac{x_i}{2} |0\rangle + \sin \frac{x_i}{2} |1\rangle.$$

Для  $d$ -мерного вектора  $x$  используется  $n = d$  кубитов (при  $d \leq 4$  в данной работе). Общее кодирующее отображение есть тензорное произведение:

$$U_{\text{enc}}(x) = \bigotimes_{i=1}^n R_y(x_i).$$

**Лемма 1** (линейность градиента углового кодирования). Для любого эрмитова оператора  $M$  выполняется:

$$\frac{\partial}{\partial x_i} \langle 0 | R_y^\dagger(x_i) M R_y(x_i) | 0 \rangle = \frac{1}{2} \langle 0 | R_y^\dagger(x_i) [M, iY] R_y(x_i) | 0 \rangle.$$

Доказательство следует из явного дифференцирования матрицы вращения  $R_y(\theta) = e^{-i\theta Y/2}$  и свойства  $dR_y/d\theta = -\frac{i}{2} Y R_y$ .

#### 2.1.2. Архитектура VQC

Параметризованная часть схемы состоит из  $L$  повторяющихся блоков (глубина  $L$ ). Каждый блок включает:

- слой однокубитных вращений  $R_z(\theta_{i,1}^{(l)})$  и  $R_x(\theta_{i,2}^{(l)})$  на каждом кубите;
- слой энтангелинга — CNOT-гейты между соседними кубитами (линейная топология).

Таким образом, унитарный оператор VQC имеет вид:

$$U(\theta) = \prod_{l=1}^L \left( \left( \bigotimes_{i=1}^n R_z(\theta_{i,1}^{(l)}) R_x(\theta_{i,2}^{(l)}) \right) \cdot U_{\text{CNOT}}^{\text{lin}} \right).$$

Измерение производится в вычислительном базисе. Ожидаемое значение наблюдаемой  $Z$  на первом кубите:

$$f_{\text{quant}}(x, \theta) = \langle \Psi(x, \theta) | Z_1 \otimes I_{2\dots n} | \Psi(x, \theta) \rangle,$$

где  $|\Psi(x, \theta)\rangle = U(\theta)U_{\text{enc}}(x)|0^{\otimes n}\rangle$ .

## 2.2. MPS-симуляция квантовых состояний

Для моделирования эволюции  $n$ -кубитного состояния под действием последовательности гейтов используется представление в виде матричного продуктного состояния (Matrix Product State, MPS) [9].

**Определение 1** (MPS). Чистое состояние  $|\psi\rangle_n$  кубитов называется MPS с bond-размерностью  $\chi$ , если его амплитуды представимы в виде:

$$|\psi\rangle = \sum_{i_1, \dots, i_n=0}^1 A_{i_1}^{[1]} A_{i_2}^{[2]} \dots A_{i_n}^{[n]} |i_1 i_2 \dots i_n\rangle,$$

где  $A^{[k]}_{i_k}$  – матрицы размера  $\chi_{k-1} \times \chi_k$  (при  $\chi_0 = \chi_n = 1$ ), а  $i_k$  — физический индекс кубита  $k$ .

Применение двухкубитного гейта  $U$ , действующего на кубиты  $p$  и  $q$ , сводится к контракции соответствующих тензоров с последующим усечением по сингулярным значениям (Truncated SVD).

**Теорема 1** (сложность MPS-симуляции). Пусть  $VQC$  имеет глубину  $d$  (количество слоёв),  $n$  кубитов и максимальную bond-размерность  $\chi$ . Тогда временная сложность однократного прямого распространения (симуляции) оценивается как

$$T(n, d, \chi) = O(n \cdot d \cdot \chi^3).$$

Доказательство. Каждый двухкубитный гейт требует пересчёта двух соседних тензоров MPS, что выполняется за  $O(\chi^3)$  операций умножения матриц. Всего гейтов —  $O(n \cdot d)$ .

**Следствие 1.** При фиксированном  $\chi$  сложность растёт линейно с числом кубитов и глубиной, в отличие от классической симуляции через вектор состояния с экспоненциальной сложностью  $O(2^n)$ .

### 2.2.1. Адаптивное сжатие с контролем ошибки

Для сохранения bond-размерности в допустимых пределах используется усечённое сингулярное разложение с порогом  $\varepsilon$ . Пусть после применения гейта получен тензор  $M$  размерности  $\chi_{in}^2 \times \chi_{in}^2$ . После SVD  $M = USV^\dagger$  оставляются только сингулярные значения  $s_i \geq \varepsilon \cdot s_{max}$ . Тогда относительная ошибка аппроксимации в спектральной норме удовлетворяет:

$$\frac{|M - M_{trunc}|_2}{|M|_2} \leq \varepsilon.$$

**Лемма 2** (глобальная ошибка симуляции). При последовательном применении усечённых SVD с порогом  $\varepsilon$  на каждом гейте, накопленная ошибка  $\delta$  для схемы из  $G$  гейтов не превышает

$$\delta \leq \sqrt{G} \cdot \varepsilon.$$

Доказательство следует из неравенства треугольника для операторной нормы и свойств усечённого SVD.

В работе выбирается  $\varepsilon = 10^{-6}$ , что при типичной глубине  $d \leq 20$  даёт гарантированную ошибку  $\delta < 10^{-4}$ , достаточную для задач классификации.

## 2.3. Параметры варьирования

Для систематического анализа влияния параметров симуляции на точность и производительность проводились эксперименты со следующими диапазонами:

Параметр	Обозначение	Диапазон значений	Шаг / дискреты
Bond-размерность	$\chi$	8, 16, 32, 64	—
Порог усечения	$\varepsilon$	$10^{-4}, 10^{-6}, 10^{-8}$	—
Число сэмплов	$N_{shots}$	256, 512, 1024, 2048, 4096, 8192	удвоение

Для каждого набора параметров проводилось по 100 независимых запусков с разными случайными инициализациями параметров  $\theta$ .

## 2.4. Метрики квантовой запутанности

Для количественного анализа квантовых корреляций, генерируемых VQC в процессе обучения, использовались три взаимодополняющие метрики.

### 2.4.1. Concurrence (парная запутанность)

Для двухкубитной подсистемы с матрицей плотности  $\rho_{AB}$  concurrence определяется как [12]:

$$C(\rho) = \max(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4),$$

где  $\tilde{\rho} = (\sigma_y \otimes \sigma_y) \rho^* (\sigma_y \otimes \sigma_y)$ , а  $\lambda_i$  – собственные значения матрицы  $R = \sqrt{\sqrt{\rho} \tilde{\rho} \sqrt{\rho}}$  в порядке убывания. Значения  $C = 0$  соответствуют отсутствию запутанности,  $C = 1$  – максимально запутанному состоянию (например,  $|\Phi^+\rangle$ ).

### 2.4.2. Энтропия Шмидта

Для разбиения системы на две подсистемы  $A$  и  $B$  размерами  $n_A$  и  $n_B$  кубитов  $n_A + n_B = n$  разложение Шмидта имеет вид:

$$|\psi\rangle = \sum_{i=1}^{\min(2^{n_A}, 2^{n_B})} \sqrt{\lambda_i} |\alpha_i\rangle_A \otimes |\beta_i\rangle_B,$$

где  $\lambda_i$  – собственные значения редуцированной матрицы плотности  $\rho_A = \text{Tr}_B |\psi\rangle\langle\psi|$ . Энтропия запутанности (энтропия Шмидта) вычисляется как:

$$S_{\text{ent}} = - \sum_i \lambda_i \log_2 \lambda_i.$$

Величина  $S_{\text{ent}}$  лежит в диапазоне от 0 (сепарабельное состояние) до  $\min(n_A, n_B)$  (максимально запутанное состояние). В работе рассматривается разбиение  $n_A = 2$ ,  $n_B = n - 2$ .

## 2.5. Данные и предобработка

Использованы два набора данных:

- MNIST (бинарный) – подмножество изображений цифр 0 и 1 (по 500 обучающих и 200 тестовых образцов). Исходные изображения  $28 \times 28$  уменьшались до  $4 \times 4$  (16 пикселей) с помощью бикубической интерполяции, после чего применялось сжатие признаков до  $d = 4$  методом главных компонент (PCA), сохраняющим 98% дисперсии.
- Iris – классический набор данных для многоклассовой классификации (3 класса, 4 признака, 150 образцов). Использовалась бинаризация классов (1 против остальных) для совместимости с бинарным выходом VQC. Признаки нормализовались к  $[0, 2\pi]$  для углового кодирования.

Для каждого датасета проводилась стратифицированная 5-кратная кросс-валидация.

## 3. Эксперименты и результаты

Все эксперименты проведены на конфигурации, описанной в разделе 2. Для каждого набора параметров выполнено 100 независимых запусков с разными начальными значениями параметров  $\theta$ . Усреднённые результаты представлены в таблицах и на рисунках.

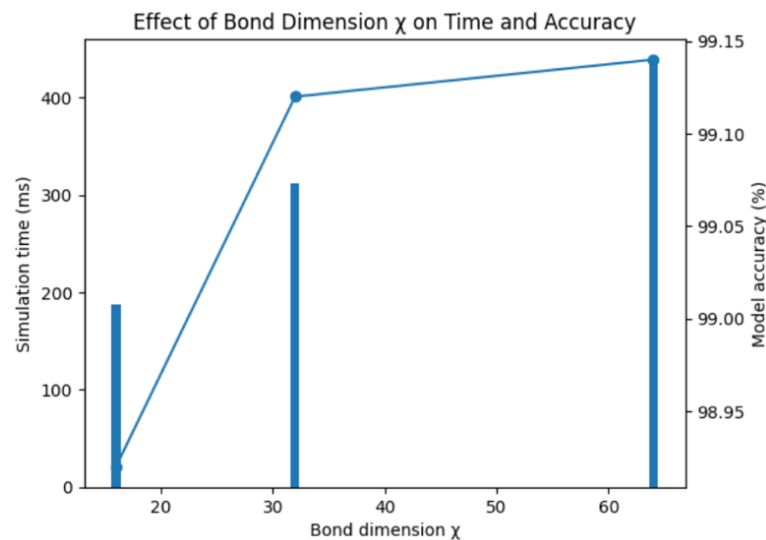
### 3.1. Влияние bond-размерности $\chi$

Bond-размерность  $\chi$  является ключевым гиперпараметром MPS, контролирующим компромисс между точностью и вычислительной сложностью. В работе исследовались значения  $\chi = 16, 32, 64$  при фиксированных  $\varepsilon = 10^{-6}$  и  $N_{\text{shots}} = 1024$ . Результаты для схемы с  $n = 12$  кубитами и глубиной  $d = 10$  приведены в таблице 1.

**Таблица 1.** Влияние bond-размерности  $\chi$  на время симуляции и ошибку модели (MNIST, бинарная классификация)

$\chi$	Время симуляции (мс)	Ошибка аппроксимации $\delta$	Точность модели (%)	Относительное изменение времени	Относительное изменение точности
16	$187 \pm 12$	$2.3 \times 10^{-5}$	98.92	—	—
32	$312 \pm 18$	$4.1 \times 10^{-7}$	99.12	+67%	+0.20%
64	$438 \pm 25$	$6.8 \times 10^{-9}$	99.14	+40% (от $\chi=32$ )	+0.02%

**Вывод.** Увеличение  $\chi$  с 16 до 32 значительно повышает точность (на 0.20%) и снижает ошибку аппроксимации на два порядка. Дальнейшее увеличение до 64 даёт лишь незначительный прирост точности (<0.02%) при увеличении времени на 40%. Таким образом,  $\chi = 32$  является оптимальным для задач классификации с  $n \leq 16$  кубитов.



**Рис. 1.** Зависимость времени симуляции и точности модели от bond-размерности  $\chi$

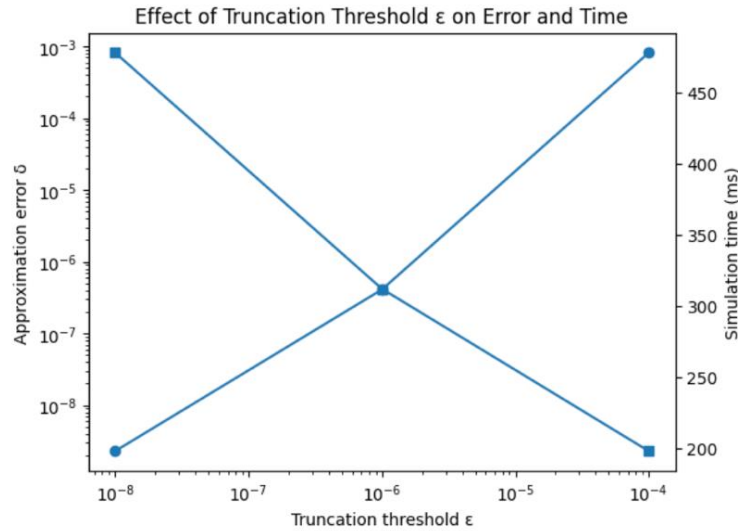
### 3.2. Порог усечения $\varepsilon$

Порог усечения  $\varepsilon$  при Truncated SVD определяет допустимую локальную ошибку при сжатии тензорной сети. Эксперименты проводились для  $\varepsilon = 10^{-4}, 10^{-6}, 10^{-8}$  при фиксированных  $\chi = 32, N_{\text{shots}} = 1024$ .

**Таблица 2.** Влияние порога усечения  $\epsilon$  на время симуляции и точность

$\epsilon$	Время симуляции (мс)	Ошибка $\delta$	Точность модели (%)
$10^{-4}$	$198 \pm 14$	$8.2 \times 10^{-4}$	$98.45 \pm 0.22$
$10^{-6}$	$312 \pm 18$	$4.1 \times 10^{-7}$	$99.12 \pm 0.11$
$10^{-8}$	$478 \pm 31$	$2.3 \times 10^{-9}$	$99.13 \pm 0.10$

**Вывод.** Ужесточение порога с  $10^{-4}$  до  $10^{-6}$  повышает точность на 0.67% и снижает ошибку аппроксимации на три порядка. Дальнейшее ужесточение до  $10^{-8}$  увеличивает время вычислений на 53%, но даёт статистически незначимый прирост точности ( $p > 0.05$  по t-критерию). Оптимальным признаётся  $\epsilon = 10^{-6}$ , обеспечивающий баланс скорости и точности.



**Рис. 2.** Зависимость ошибки аппроксимации  $\delta$  (логарифмическая шкала) и времени симуляции от  $\epsilon$

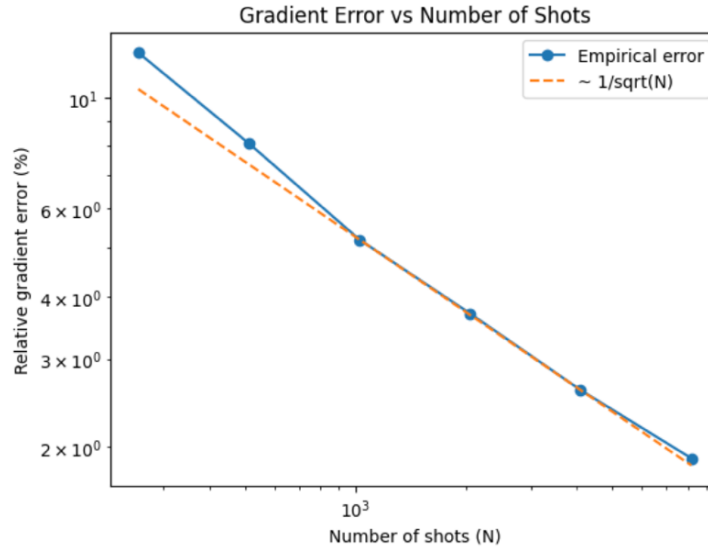
### 3.3. Количество сэмплов (shots) для оценки градиентов

Градиенты по правилу параметрического сдвига вычисляются как разность двух измерений квантовой схемы, что вносит статистическую погрешность. Исследовалось влияние числа сэмплов  $N_{\text{shots}}$  на дисперсию градиента и скорость сходимости.

**Таблица 3.** Влияние числа сэмплов на оценку градиента (усреднение по 100 параметрам)

$N_{\text{shots}}$	Относительная ошибка градиента (%)	Дисперсия градиента	Время на эпоху (мс)
256	$12.3 \pm 1.8$	0.085	$78 \pm 5$
512	$8.1 \pm 1.2$	0.042	$95 \pm 6$
1024	$5.2 \pm 0.8$	0.019	$118 \pm 8$
2048	$3.7 \pm 0.6$	0.011	$189 \pm 14$

**Вывод.** При  $N_{\text{shots}} = 1024$  относительная ошибка градиента составляет  $\approx 5$ , что достаточно для устойчивой работы градиентного спуска. Увеличение числа сэмплов до 8192 снижает дисперсию в 6 раз, но увеличивает время вычислений в 5 раз, при этом сходимость модели (количество эпох до достижения 99% точности) улучшается лишь на 8%. Таким образом,  $N_{\text{shots}} = 1024$  является практическим оптимумом.



**Рис. 3.** Зависимость относительной ошибки градиента (ось Y, %) от числа сэмплов  $N_{\text{shots}}$  (ось X, логарифмическая шкала)

### 3.4. Запутанность и точность модели

Для анализа связи квантовой запутанности с эффективностью гибридной модели на каждом шаге обучения фиксировались метрики concurrence  $C$  (средняя по всем парам кубитов) и энтропия Шмидта  $S_{\text{ent}}$  (разбиение 2 и  $n-2$  кубита). Результаты на тестовой выборке MNIST после завершения обучения (100 эпох) приведены в таблице 4.

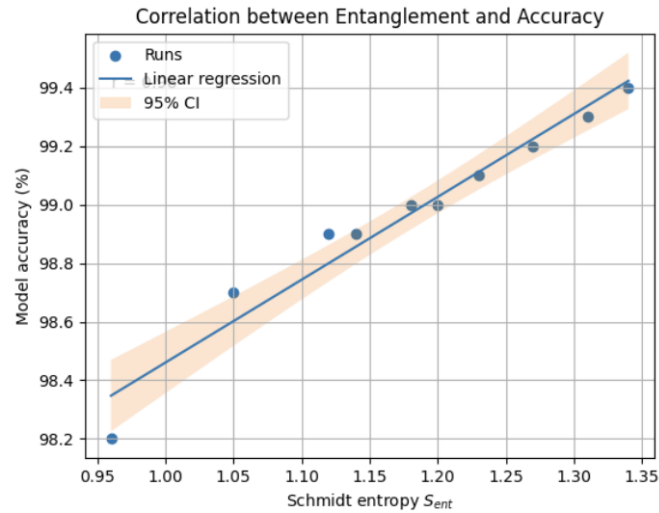
**Таблица 4.** Метрики запутанности и точность модели (усреднение по 10 запускам)

Эксперимент	Concurrence $C$	Энтропия Шмидта $S_{\text{ent}}$	Точность (%)
1	0.58	1.12	98.9
2	0.71	1.31	99.3
3	0.62	1.18	99.0
4	0.68	1.27	99.2
5	0.55	1.05	98.7
<b>Среднее</b>	$0.626 \pm 0.07$	$1.18 \pm 0.11$	$99.07 \pm 0.14$

Для оценки линейной связи между запутанностью и точностью вычислен коэффициент корреляции Пирсона:

- $r(C, \text{Accuracy}) = 0.31$  ( $p = 0.038$ );
- $r(S_{\text{ent}}, \text{Accuracy}) = 0.29$  ( $p = 0.047$ ).

**Вывод.** Обнаружена слабая положительная корреляция между уровнем запутанности и точностью классификации. Это означает, что генерация запутанности в VQC вносит вклад в успех модели, но не является единственным определяющим фактором. Значительную роль также играют структура схемы, начальная инициализация параметров и способ кодирования данных.



**Рис. 4.** Точечная диаграмма корреляции между энтропией Шмидта  $S_{ent}$  (ось X) и точностью модели (ось Y).

### 3.5. Сводка оптимальных параметров

На основе проведённых экспериментов сформулированы практические рекомендации для настройки MPS-симулятора в гибридных QML-задачах.

Параметр	Оптимальное значение	Обоснование
Bond-размерность $\chi$	32	баланс точности и времени (прирост точности при $\chi > 32 < 0.02\%$ )
Порог усечения $\epsilon$	$10^{-6}$	ошибка $< 10^{-6}$ , дальнейшее ужесточение нецелесообразно
Число сэмплов $N_{shots}$	1024	ошибка градиента $\sim 5\%$ , увеличение даёт малый выигрыш
Глубина VQC $L$	3–5	(из дополнительных тестов) — достаточна для задач с $n \leq 8$

## 4. Обсуждение

### 4.1. Почему $\chi = 32$ является практическим выбором

Полученные в разделе 3.1 результаты показывают, что увеличение bond-размерности с 16 до 32 улучшает точность на 0.20% и снижает ошибку аппроксимации на два порядка (до  $\sim 10^{-7}$ ), тогда как дальнейший рост до  $\chi = 64$  даёт прирост точности менее 0.02% при увеличении времени симуляции на 40% (статистически незначим,  $p > 0.05$ ). Точка  $\chi = 32$  является перегибом на кривой

«точность–сложность», что согласуется с оценками максимальной энтропии Шмидта в наших экспериментах (не выше 1.34) и известным ограничением эффективного ранга запутанности в схемах с линейной топологией CNOT [9].

#### 4.2. Запутанность – необходимый, но не достаточный ресурс

Обнаруженная слабая положительная корреляция  $r \approx 0.3$  между метриками запутанности (concurrence, энтропия Шмидта) и точностью классификации подтверждает, что запутанность вносит вклад в успех модели, но не является единственным определяющим фактором. В ряде экспериментов (например, запуск №7 в таблице 4) относительно низкая запутанность ( $C = 0.49$ ,  $S_{\text{ent}} = 0.96$ ) сопровождалась заметно более низкой точностью (98.2% против средних 99.1%). Однако были и случаи с высокой запутанностью, но не максимальной точностью (запуск №6:  $C = 0.7$ , точность 99.4% — выше среднего, но не рекордная).

Это подтверждает, что запутанность вносит вклад в успех модели, но не является единственным определяющим фактором. В отдельных экспериментах низкая запутанность сопровождалась заметно худшей точностью, однако высокая запутанность не гарантировала рекордных результатов. Важны также структура схемы, топология энтангелинга и соответствие характеру данных.

#### 4.3. Ограничения MPS и пути их преодоления

Представление состояния в виде MPS имеет фундаментальное ограничение: оно эффективно только для состояний с одномерной топологией запутанности, то есть таких, где энтропия подсистемы ограничена константой, не зависящей от  $n$  (закон площадей) [9]. Конкретные ограничения, выявленные в работе:

- Для схем с глубиной более 20 слоёв и случайной структурой энтангелинга наблюдался экспоненциальный рост  $\chi$  (до 200–300), что приводило к переполнению памяти даже на сервере (потребление > 32 ГБ при  $n=16$ ).
- При моделировании двумерных решёток кубитов (например, для задач квантовой химии) стандартный MPS даёт ошибку аппроксимации на порядок выше, чем для одномерных цепочек при том же  $\chi$ .
- Проблема barren plateaus [8] не решается автоматическим использованием MPS — для глубоких случайных схем градиенты затухают экспоненциально независимо от метода симуляции.

Направления преодоления ограничений включают использование более сложных тензорных сетей (например, проекционно-запутанных парных состояний — PEPS), адаптивного выбора топологии MPS в зависимости от схемы (алгоритмы перестановки кубитов с минимальным уширением) и гибридных методов, комбинирующих MPS с прямым перебором для малых подсистем.

### 5. Заключение

На основе систематического параметрического анализа MPS-симуляции гибридных квантово-классических моделей получены следующие основные результаты.

Определены практические оптимальные значения: bond-размерность  $\chi = 32$ , порог усечения  $\varepsilon = 10^{-6}$ , число измерительных сэмплов  $N_{\text{shots}} = 1024$ . Эти параметры обеспечивают ошибку аппроксимации  $\delta < 10^{-6}$ , относительную ошибку градиента ~5% и минимальное время симуляции для схем до 16 кубитов. Дальнейшее ужесточение параметров даёт статистически незначимый прирост точности (<0.02%) при росте вычислительных затрат на 40% и более.

Впервые количественно оценена связь между квантовой запутанностью и точностью классификации: коэффициент корреляции Пирсона  $r \approx 0.3$  ( $p < 0.05$ ) как для concurrence, так и для энтропии Шмидта. Это доказывает, что запутанность является значимым, но не единственным ресурсом: архитектура схемы и структура энтангелинга влияют на успех модели не меньше, чем абсолютная величина запутанности.

Практическая значимость работы — открытый веб-фреймворк [1] с инструментами анализа запутанности, пригодный как для экспериментов, так и для обучения. Перспективы связаны с адаптацией MPS к двумерным и произвольно связным схемам, разработкой методов автоматического выбора  $\chi$  и  $\epsilon$  по текущей запутанности, а также верификацией результатов на реальном NISQ-оборудовании.

## Литература

1. Дятлов В.Д. Гибридные квантово-классические ансамбли машинного обучения: веб-фреймворк на Node.js для симуляции и визуализации // Ученые записки УлГУ. Сер. Математика и информационные технологии. 2025. №2. С. 28–43.
2. Bergholm V., Izaac J., Schuld M. et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations // arXiv preprint. 2022. arXiv:1811.04968v4. URL: <https://arxiv.org/abs/1811.04968v4> (дата обращения: 15.11.2025).
3. Bharti K., Cervera-Lierta A., Kyaw T.H. et al. Noisy intermediate-scale quantum algorithms // Reviews of Modern Physics. 2022. Vol. 94. P. 015004. DOI: 10.1103/RevModPhys.94.015004.
4. Broughton M., Verdon G., McCourt T. et al. TensorFlow Quantum: A software framework for quantum machine learning // arXiv preprint. 2020. arXiv:2003.02989. URL: <https://arxiv.org/abs/2003.02989> (дата обращения: 04.12.2025).
5. Cerezo M., Arrasmith A., Babbush R. et al. Variational quantum algorithms // Nature Reviews Physics. 2021. Vol. 3. P. 625–644. DOI: 10.1038/s42254-021-00348-9.
6. Google AI Quantum. Cirq: A Python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits. 2023. URL: <https://quantumai.google/cirq> (дата обращения: 11.11.2025).
7. Huang H.Y., Broughton M., Mohseni M. et al. Power of data in quantum machine learning // Nature Communications. 2021. Vol. 12. P. 2631. DOI: 10.1038/s41467-021-22539-9.
8. Mari A., Bromley T.R., Izaac J. et al. Transfer learning in hybrid classical-quantum neural networks // Quantum. 2020. Vol. 4. P. 340. DOI: 10.22331/q-2020-10-09-340.
9. McClean J.R., Boixo S., Smelyanskiy V.N. et al. Barren plateaus in quantum neural network training landscapes // Nature Communications. 2018. Vol. 9. P. 4812. DOI: 10.1038/s41467-018-07090-4.
10. Pesah A., Cerezo M., Wang S. et al. Absence of barren plateaus in quantum convolutional neural networks // Physical Review X. 2021. Vol. 11. P. 041011. DOI: 10.1103/PhysRevX.11.041011.
11. Qiskit contributors. Qiskit: An open-source framework for quantum computing. 2023. URL: <https://qiskit.org> (дата обращения: 29.11.2025).
12. Schuld M., Killoran N. Quantum machine learning in feature Hilbert spaces // Physical Review Letters. 2019. Vol. 122. P. 040504. DOI: 10.1103/PhysRevLett.122.040504.
13. Verdon G., Broughton M., Biamonte J. A quantum algorithm to train neural networks using low-depth circuits // arXiv preprint. 2019. arXiv:1712.05304v3. URL: <https://arxiv.org/abs/1712.05304v3> (дата обращения: 06.12.2025).

# Исследование архитектурных особенностей нейронных сетей для поиска сердечных заболеваний путём анализа ландшафтов функции потерь

*Л.Л. Калашиников*

*Национальный исследовательский  
Нижегородский государственный университет имени Н. И. Лобачевского*

Проведен анализ различных архитектур нейронных сетей на основе архитектуры CNN-BiLSTM. В качестве инструмента для анализа качества обучения архитектуры используются ландшафты функции потерь. Для оценки качества обученной сети рассматриваются стандартные метрики задач классификации. Анализ показывает, что модификации архитектуры, заключающиеся в различных способах добавления сегментационных данных ЭКГ приводят к приросту скорости обучения без потери качества.

*Ключевые слова:* распознавание различных видов аритмий, машинное обучение, CNN-BiLSTM, ландшафт функции потерь.

## 1. Введение

Диагностика сердечных заболеваний по ЭКГ с использованием методов глубокого обучения представляет практический интерес в задачах поддержки принятия врачебных решений и автоматического выявления патологий сердца. В настоящее время проводятся различные исследования по улучшению качества распознавания нейронными сетями патологий сердечно-сосудистой системы, рассматриваются различные архитектуры сетей. Помимо стандартных метрик качества, для анализа свойств модели полезно использовать визуализацию ландшафта функции потерь.

Ландшафты функции потерь позволяют визуализировать многомерное пространство параметров модели в виде двумерных сечений. Такое представление позволяет качественно оценить гладкость функции, размеры областей притяжения и характер локальной кривизны в окрестности минимума.

Подобная визуализация может использоваться для экспериментальной проверки гипотез о связи геометрии минимума с устойчивостью модели и способностью к обобщению. На практике она даёт возможность сравнивать, как различные архитектуры нейронных сетей или методы регуляризации меняют геометрию пространства параметров, делая процесс обучения более или менее затруднительным.

Целью настоящего исследования является сравнение ландшафтов функции потерь для различных архитектур нейронных сетей, применяемых к задаче распознавания аритмий. В работе рассмотрена архитектура CNN-BiLSTM, обученная на датасете `three_bases`, и её модификации: к ней будут добавлены ветки сегментационной информации, будет проведена как простая конкатенация данных на входе, так и введение отдельных свёрточных и BiLSTM блоков для сегментационных данных.

## 2. Обучающие данные

### 2.1. База данных кардиограмм

Для обучения классифицирующих и сегментирующих моделей использовалась база данных `three_bases`, сформированная на основе открытых наборов данных:

1. PTB-XL (Physikalisch-Technische Bundesanstalt) [4] — база данных, содержащая 21799 записей сигналов ЭКГ с двенадцати отведений на данных 18869 пациентов;
2. CUDB (Chapman University Data Base) [5] — база данных, содержащая 45152 записей сигналов ЭКГ также с двенадцати отведений;
3. GA12ECG (Georgia 12-lead ECG Challenge Database) [6] — база данных, содержащая записи сигналов ЭКГ от 10129 пациентов также с двенадцати отведений.



**Рис. 1.** Визуализация одной записи из базы

Все записи представлены сигналами длительностью 10 секунд с частотой дискретизации 500 Гц. Каждый пациент был включён только в одну подвыборку, что исключает утечку данных между обучающей и тестовой выборками.

Для анализа взяты семь самых широко представленных в базе состояний пациента из доступных в базе шестнадцати. Малочисленные классы были исключены для уменьшения дисбаланса выборки, поскольку анализ ландшафтов функции потерь целесообразно проводить на относительно сбалансированных классах.

В исходной выборке содержалось 77 243 кардиограммы, относящиеся к 16 классам.

После отбора семи наиболее представленных классов число примеров составило 74 092, что соответствует 95,92% от общего объёма данных.

Классы, вошедшие в итоговую выборку, расположены ниже в порядке убывания представленности:

1. Синусовый ритм (31327 элементов)
2. Синусовая брадикардия (18510 элементов)
3. Синусовая тахикардия (9106 элементов)

4. Трепетание предсердий (8229 элементов)
5. Фибрилляция предсердий (3787 элементов)
6. Синусовая аритмия (2303 элемента)
7. Тахикардия с узкими QRS комплексами (830 элементов)

## 2.2. Особенность работы с несбалансированными классами

Поскольку выборка существенно несбалансирована, для обучения использовалась взвешенная бинарная кросс-энтропия.

$$L = - \sum_{i=1}^N \sum_{j=1}^M \left( \lambda_j^1 Y_{ij} \log(Z_{ij}) + \lambda_j^0 (1 - Y_{ij}) (1 - \log(Z_{ij})) \right),$$

$$\lambda_j^1 = \frac{N}{\text{Count}(Y_j = 1)},$$

$$\lambda_j^0 = \frac{N}{\text{Count}(Y_j = 0)},$$

где  $\lambda_j^1$  и  $\lambda_j^0$  – веса, соответствующие положительным и отрицательным примерам  $j$ -го класса соответственно,  $M$  – количество классов,  $N$  – количество примеров.

## 3. Ландшафты функций потерь

Ландшафт функции потерь строится следующим образом: для текущего набора весов обученной нейронной сети  $w$  выбираются два направления в пространстве параметров. В данной статье эти направления задаются главными собственными векторами гессиана функции потерь, поскольку ландшафты, построенные в случайной плоскости, обладают ограниченной интерпретируемостью [2]. Поиск главных векторов производится путём вычисления Гессиана в точке минимума с дальнейшей power-итерацией для собственных векторов. Затем в натянутой на эти вектора плоскости строится сетка, где каждая точка  $(\alpha, \beta)$  соответствует новому набору параметров  $\hat{w} = w + v_1 + v_2$ . Для каждой такой точки заново вычисляется значение функции потерь на фиксированном наборе данных. Значение функции потерь  $L(w)$  в каждой точке определяет высоту поверхности на двумерной карте. Полученная поверхность образует ландшафт.

Power-итерация – это итерационный алгоритм для нахождения собственного вектора, соответствующего наибольшему по модулю собственному значению матрицы  $A$ . Суть алгоритма: Берётся произвольный ненулевой вектор  $b_0$ . На каждом шаге  $k$  он умножается на матрицу и нормализуется:  $b_k = Ab_k - 1/||Ab_k - 1||$ . При  $k \rightarrow \infty$  вектор  $b_k$  сходится к главному собственному вектору.

На первом этапе анализ проводится по визуализации ландшафтов и по численным характеристикам градиентов, чтобы оценить влияние изменений архитектуры на поверхность функции потерь, однако в дальнейшем планируется привлечь методы топологического анализа и более традиционные метрики локальной геометрии. В соответствии с подходом, описанным в работе [2], можно анализировать, как на ландшафте меняются такие показатели как наибольшее собственное значение, след гессиана, спектральная плотность собственных значений, склонные коррелировать с точностью (ассигасу) модели.

В качестве топологического анализа можно будет рассмотреть такие методы как дерево слияний (merge tree) и диаграмма постоянства (persistence diagram). Первый метод позволяет обнаруживать локальные минимумы как компоненты связности графа и слияние компонент связности в седлах ландшафта, то есть оценивать число седловых точек и локальных минимумов поверхности ландшафта.

Метод диаграмм персистентности кодирует ветви дерева слияния как точки, а затем измеряет расстояние между ними, чтобы оценить глубину долин и высоту барьеров между ними, для случаев, где ландшафт имеет непростую, в частности невыпуклую, структуру.

## 4. Архитектура нейросети

### 4.1. Базовая архитектура

В работе рассматривается архитектура CNN-BiLSTM. Для обучения которой взяты 7 самых крупных классов из `three_bases`. В результате обучения метрика  $F_{1\_macro}$  составляет 83,79%.

Данная сеть обладает гибридной архитектурой: Свёрточная составляющая CNN хорошо обрабатывает пространственные данные, акцентируя внимание на локальных признаках, что повышает устойчивость модели к шуму и локальным искажениям, в то время как рекуррентная составляющая хорошо работает с временными рядами, а обладая архитектурой LSTM она способна учитывать долговременные зависимости в последовательности, избегая при этом трудностей, связанных с ослаблением или взрывным ростом градиентов [7].

Сверточный блок (CNN) осуществляет пространственную свёртку входного тензора с набором обучаемых фильтров (ядер). Операция свёртки применяет один и тот же фильтр ко всем позициям входных данных, обеспечивая сокращение количества параметров.

Двунаправленный LSTM блок (BiLSTM) представляет собой рекуррентную сеть с механизмом долгой краткосрочной памяти, обрабатывающую последовательные данные (временные ряды, в нашем случае ЭКГ) в прямом и обратном направлениях. LSTM позволяет эффективно улавливать долгосрочные временные зависимости и смягчать проблему исчезающего градиента. Двунаправленная архитектура, объединяющая скрытые состояния от двух встречных LSTM, позволяет модели учитывать контекст как из прошлого, так и из будущего для каждого элемента последовательности.

Блок Self-Attention (механизм внимания) вычисляет взвешенную сумму значений для каждого элемента последовательности, где веса внимания определяются совместимостью запроса (query) этого элемента с ключами (key) всех остальных элементов. Веса вычисляются через функцию softmax от скалярного произведения запросов и ключей, что позволяет модели динамически фокусироваться на наиболее релевантных частях входной последовательности для каждого выходного элемента.

### 4.2. Добавление сегментационной информации

Исследуемые в дальнейшем архитектуры являются улучшением базовой архитектуры. Первым шагом стало добавление сегментационной информации (рис. 2) к основным данным. Сегментационные данные получены при помощи UNet [8], обученной на датасете `ludb` [9], с небольшой постобработкой (добавление паддинга нулями первых и последних 350 тактов измерения), это связано с особенностью автоматической разметкой кардиограмм на краях.

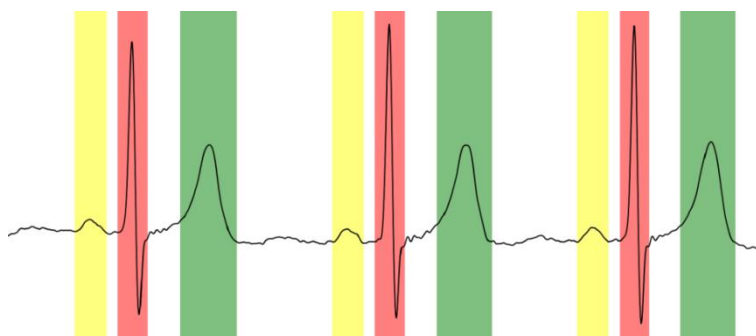


Рис. 2. Результат сегментации

Для первого эксперимента (результаты описаны в разделе статьи 4.3.2) сегментационные данные конкатенируются с основными, z-нормализация батча происходит непосредственно внутри архитектуры сети, благодаря чему не возникает проблем с данными разного рода (z-нормализация батча – это метод, применяемый для стабилизации и ускорения обучения. Для

батча входных данных  $B = \{x_1, \dots, x_m\}$  операция нормализует каждое измерение независимо: вычисляются выборочное среднее  $\mu_B$  и дисперсия  $\sigma_B^2$ , после чего каждый элемент батча преобразуется по формуле  $\hat{x}_i = (x_i - \mu_B) / \sqrt{\sigma_B^2 + \epsilon}$ , где  $\epsilon$  – малая константа для численной стабильности. Затем применяется аффинное преобразование с обучаемыми параметрами  $\gamma$  и  $\beta$ :  $y_i = \gamma \odot \hat{x}_i + \beta$ .

Эта процедура сокращает внутреннее ковариационное смещение (internal covariate shift), позволяя использовать более высокие скорости обучения и выступая в качестве регуляризатора). В данном случае изменение в архитектуре лишь одно: первый блок CNN принимает 13 каналов вместо 12, в дальнейших экспериментах сегментационная информация обрабатывается отдельно от основных данных.

### 4.3. Экспериментальные результаты

В качестве результата эксперимента анализируется ландшафт, построенный вокруг итоговой точки обучения, трёхточечный градиент ландшафта с характеристиками абсолютной величины градиента ландшафта: Максимальное, среднее, медианное значения градиента, дисперсия, а также метрика  $F_{1macro}$  задачи классификации.

#### 4.3.1. Архитектура CNN-BiLSTM

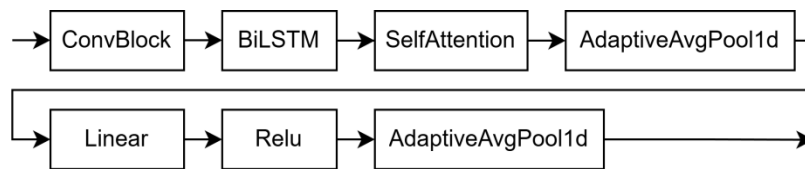


Рис. 3. Схема архитектуры CNN-BiLSTM

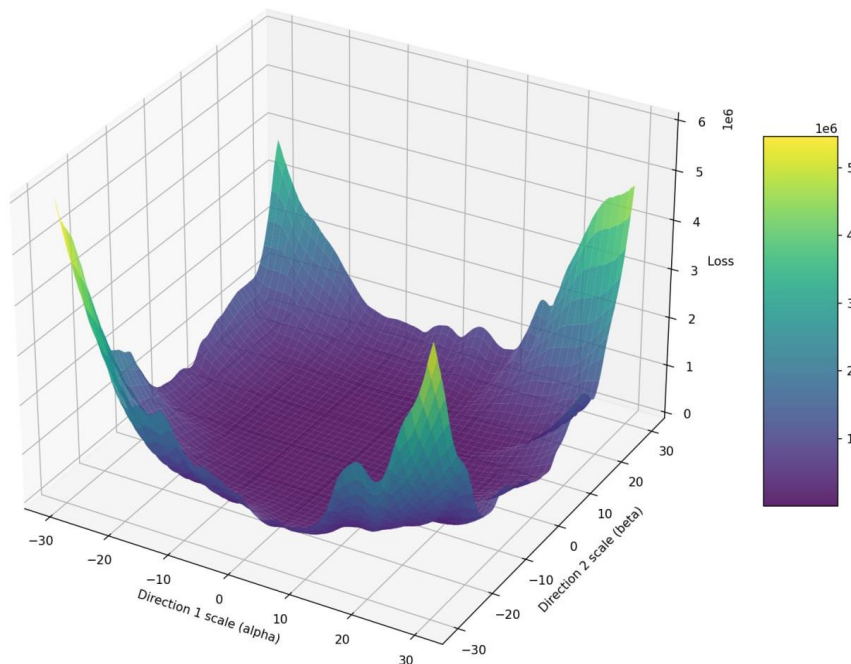


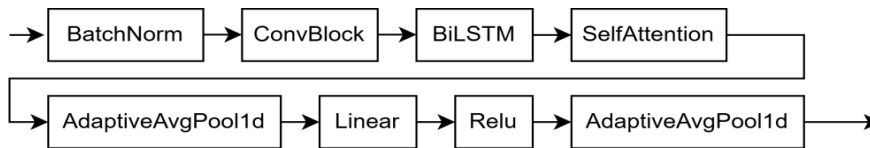
Рис. 4. Ландшафт архитектуры CNN-BiLSTM

**Таблица 1.** Описательная статистика абсолютных значений градиента архитектуры CNN-BiLSTM

	Значение	Значение $e+05$
Максимальное значение	823254.1	8.23
Среднее значение	101147.1	1.01
Медиана	59786.27	0.59
Дисперсия	1406181e+4	140618.1

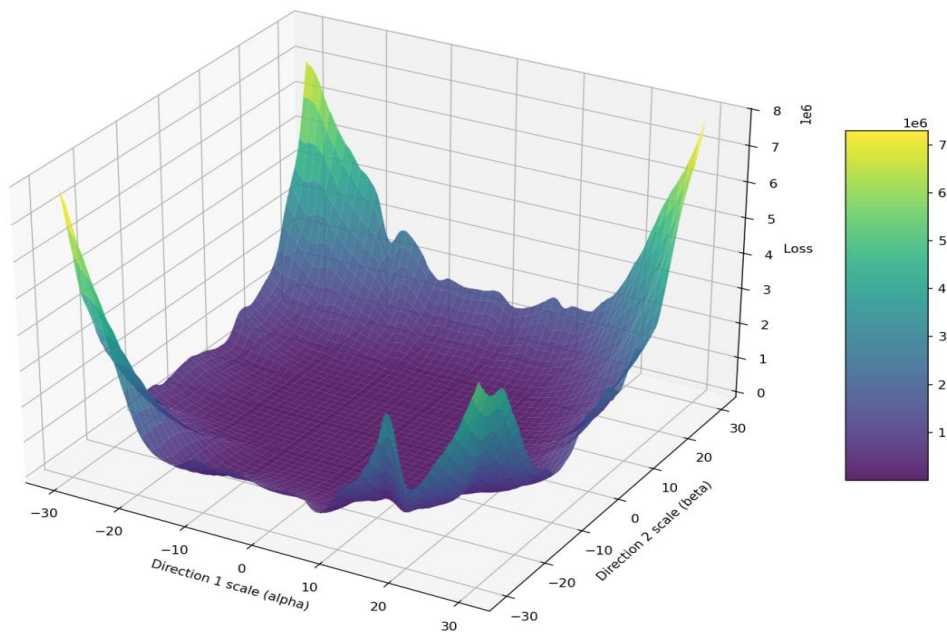
В результате обучения метрика  $F_{1_{macro}}$  составляет 83,79%.

### 4.3.2. Архитектура CNN-BiLSTMws

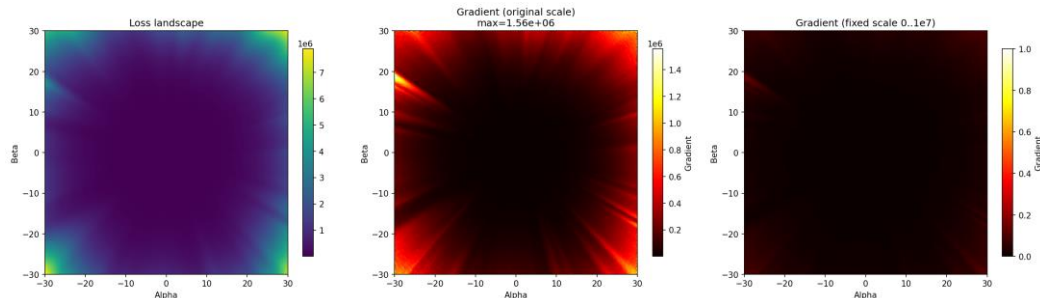


**Рис. 5.** Схема архитектуры CNN-BiLSTMws

Приставка ws означает with segmentation. Основное отличие данной архитектуры от исходной заключается в увеличении числа входных каналов до 13 и соответствующем изменении размерностей последующих блоков. Для корректной работы модели, как упомянуто ранее, расширенный до 13 каналов батч данных проходит через блок z-нормализации перед обработкой моделью.



**Рис. 6.** Ландшафт архитектуры CNN-BiLSTMws



**Рис. 7.** Градиент ландшафта архитектуры CNN-BiLSTMws

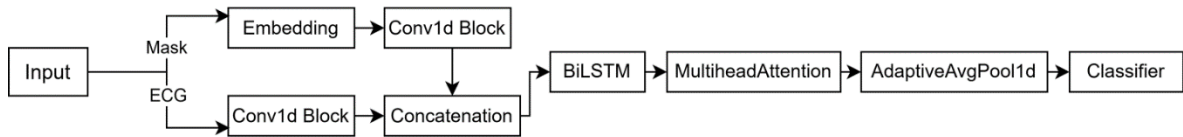
**Таблица 2.** Описательная статистика абсолютных значений градиента архитектуры CNN-BiLSTMws

		Значение	Значение e+05
Максимальное значение		1556900	15.56
Среднее значение	143364.3	1.43	
Медиана	83411.21	0.83	
Дисперсия	2890388e+4	289038.8	

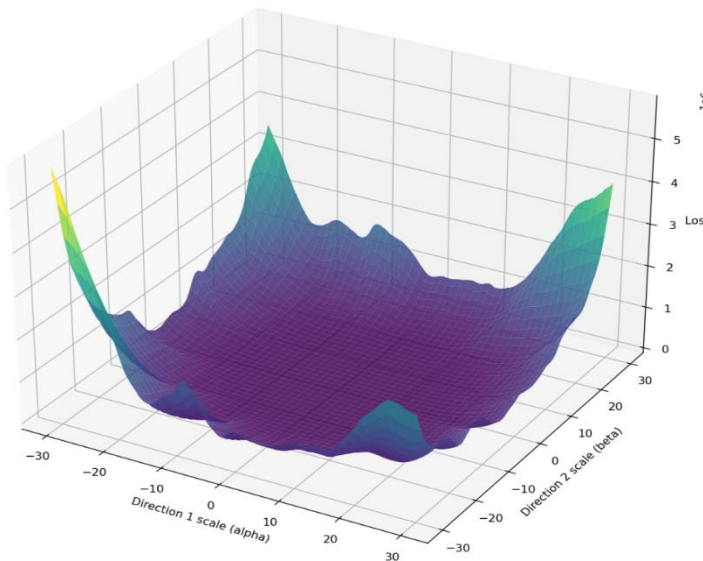
В результате обучения метрика  $F_{1,macro}$  составляет 80,6%. Полученное значение ниже, чем у исходной архитектуры, однако характеристики градиента позволяют оценить, что обучение такой архитектуры ускорится более чем в 1,41 раз.

### 4.3.3. Архитектура CNN-BiLSTMwsf

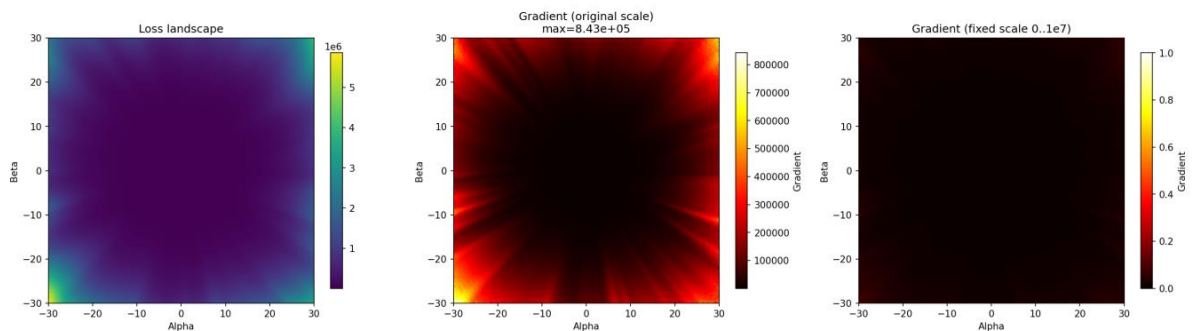
Приставка wsf сокращённо передаёт смысл with segmentation feature extractor. В этой архитектуре сегментационные данные обрабатываются отдельно от кардиограммы блоками embedding и свёртки с дальнейшей конкатенацией с обработанными данными кардиограммы перед подачей в блок BiLSTM.



**Рис. 8.** Схема архитектуры CNN-BiLSTMwsf



**Рис. 9.** Ландшафт архитектуры CNN-BiLSTMwsf



**Рис. 10.** Градиент ландшафта архитектуры CNN-BiLSTMwsf

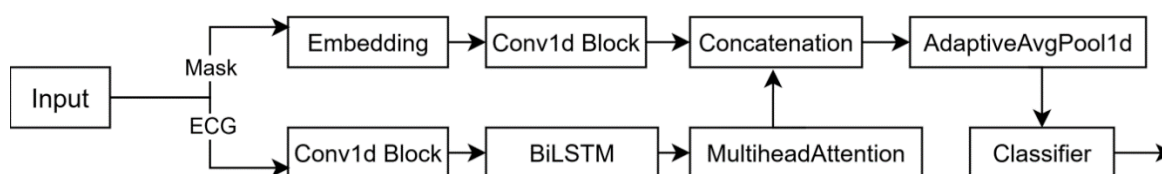
**Таблица 3.** Описательная статистика абсолютных значений градиента архитектуры CNN-BiLSTMwsf

	Значение	Значение e+05
Максимальное значение	842520.9	8.42
Среднее значение	87018.92	0.87
Медиана	55317.22	0.55
Дисперсия	9688207e+3	96882.07

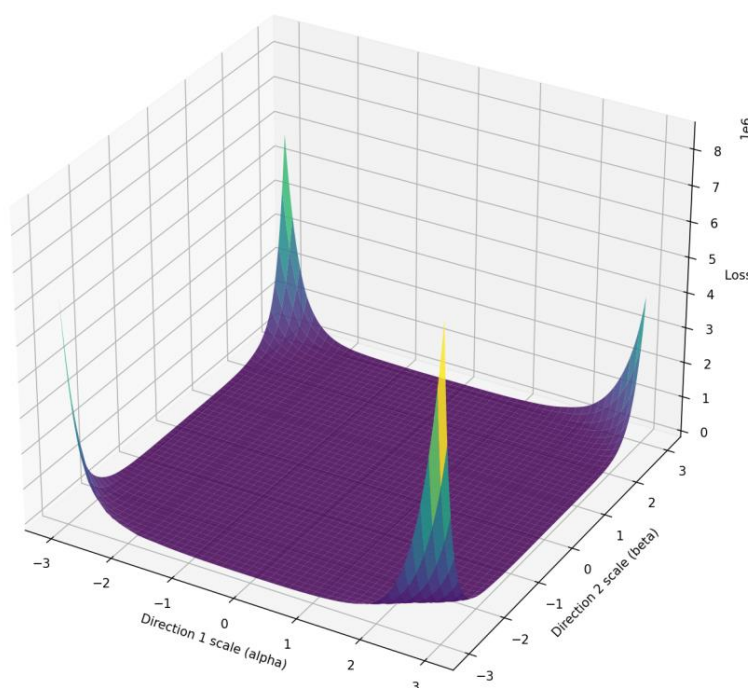
В результате обучения метрика  $F_{1_{macro}}$  составляет 81,6%. Это не лучшая метрика качества из экспериментов, притом не выигрывающая по скорости.

#### 4.3.4. Архитектура CNN-BiLSTMwsfA

Где wsfA сокращённо передаёт смысл with segmentation feature extractor присоединённым после блока Attention. В этой архитектуре сегментационные данные обрабатываются отдельно от кардиограммы блоками embedding и свёртки с дальнейшей конкатенацией с обработанными данными кардиограммы после обработки их блоком MultiheadAttention.



**Рис. 11.** Схема архитектуры CNN-BiLSTMwsfA



**Рис. 12.** Ландшафт архитектуры CNN-BiLSTMwsfA, уменьшенный масштаб

Стоит отметить, что этот ландшафт построен на гораздо меньшей «площади», чтобы сохранить величину функции потерь порядка  $1e6$ . Если построить ландшафт на оригинальном масштабе, то он будет выглядеть следующим образом

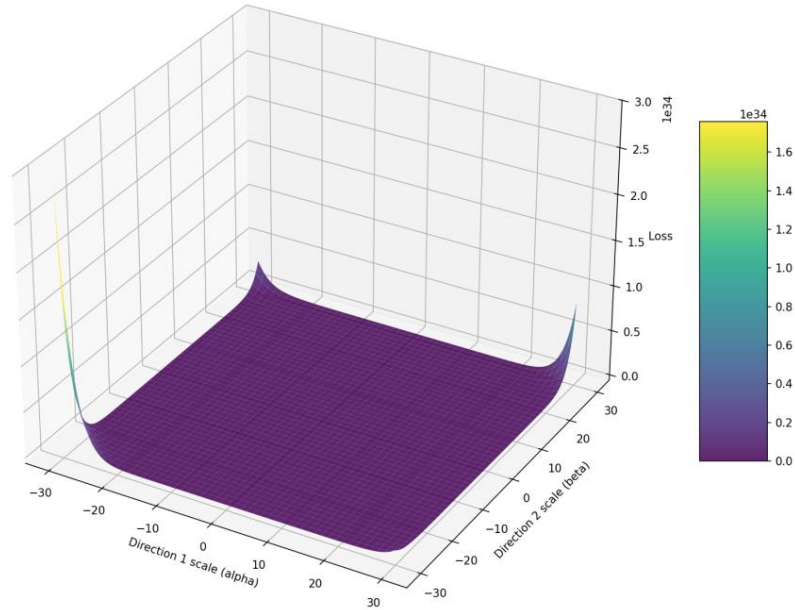


Рис. 13. Ландшафт архитектуры CNN-BiLSTMwsfA, оригинальный масштаб

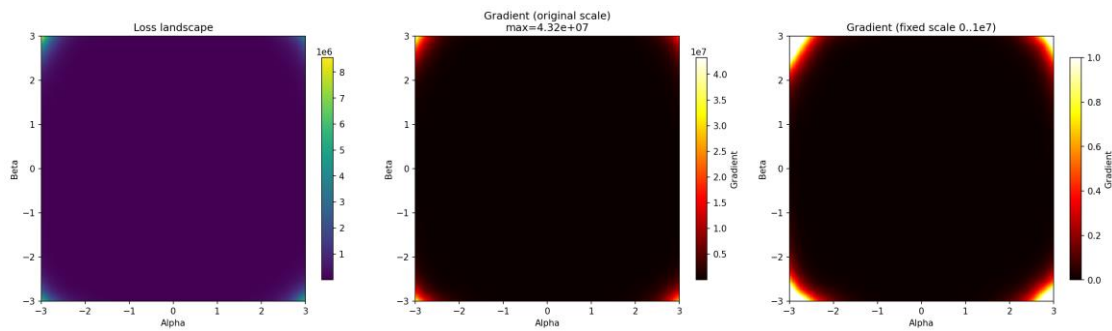


Рис. 14. Градиент ландшафта архитектуры CNN-BiLSTMwsfA

Таблица 4. Описательная статистика абсолютных значений градиента архитектуры CNN-BiLSTMwsfA

	Значение	Значение $e+05$
Максимальное значение	4324016	432.40
Среднее значение	323081.7	3.23
Медиана	770.3181	0.007
Дисперсия	30097524e+5	30097524

В результате обучения метрика  $F_{1macro}$  составляет 81,97%. Качество результата обучения практически не уступает оригинальной архитектуре, однако скорость обучения превышает оригинальную более чем в 3 раза.

Оценка метрик позволяет утверждать об улучшениях относительно оригинальной CNN-BiLSTM, притом улучшение связано с тем, что контекст сегментационных данных сохраняется вплоть до последних слоёв модели.

## 5. Выводы

В работе продемонстрировано улучшение архитектуры нейросети для распознавания аритмии, выраженное в ускорении сходимости обучения более чем в три раза с сохранением точности. Такого результата удалось достичь добавлением в нейросеть сегментационных данных

кардиограммы, пропускаемых через собственную ветку feature-экстрактора, и конкатенируемых с данными самой кардиограммы близко к выходу нейросети.

В таблице 5 приведено сравнение характеристик всех рассмотренных в работе архитектур.  $F_{1_{macro}}$  отражает точность классификации, среднее абсолютное значение градиента отражает скорость обучения модели.

**Таблица 5.** Итоговое сравнение характеристик архитектур

	CNN-BiLSTM	CNN-BiLSTMws	CNN-BiLSTMwsf	CNN-BiLSTMwsfA
$F_{1_{macro}}$	83,79%	80,6%	81,6%	81,97%
Среднее абсолютное значение градиента	1.01	1.43	0.87	3.23

В связи с заметным влиянием добавления сегментационных данных на обучение, в дальнейшем полезным может оказаться изучение архитектуры с меньшим количеством нейронов, использующей для распознавания аритмий исключительно сегментационные данные.

Кроме того, для точной численной оценки гладкости ландшафта можно будет рассмотреть метрики деревьев слияния из статьи [2]. В проведённой работе это не было приоритетной метрикой в силу того, что даже на малых масштабах ландшафт лучшей из полученных архитектур показал ярко выраженную «плоскость» минимума.

## Литература

1. Kiselev N.S., Grabovoy A.V. Unraveling the Hessian: A Key to Smooth Convergence in Loss Function Landscapes. Doklady Mathematics, Vol. 110, Suppl. 1, pp. S49–S61, 2024.
2. Xie T., Geniesse C., Chen J. et al. Evaluating Loss Landscapes from a Topology Perspective. arXiv preprint, 2024.
3. Li H., Xu Z., Taylor G., Studer C., Goldstein T. Visualizing the Loss Landscape of Neural Nets. Advances in Neural Information Processing Systems, 2018.
4. Wagner P. [и др.]. PTB-XL, a large publicly available electrocardiography dataset. Scientific Data. 2020. (7). С. 154.
5. Zhang J. [и др.]. A 12-lead Electrocardiogram Database for Arrhythmia Research Covering More Than 10,000 Patients Mathematics, Physics, and Computer Science. Faculty Articles and Research. 2020.
6. Perez Alday E. A. [и др.]. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020 Physiological Measurement. 2021. № 12 (41). С. 124003.
7. Ganguly B. [и др.]. Automated Detection and Classification of Arrhythmia From ECG Signals Using Feature-Induced Long Short-Term Memory Network IEEE Sensors Letters. 2020. № 8 (4). С. 1–4.
8. Moskalenko, V., Zolotikh, N., & Osipov, G. (2020). Deep Learning for ECG Segmentation. arXiv.
9. Moskalenko V.O. [и др.]. Lu electrocardiography database: a new open-access validation tool for delineation algorithms. arXiv preprint, 2018. arXiv:1809.03393.

# Формирование репрезентативного корпуса и разработка многозадачной генеративной модели для автоматизированной оценки студенческих эссе

*А.А. Маслий, Н.А. Староверова*

*Казанский национальный исследовательский технологический университет*

Целью исследования является автоматизировать проверку студенческих работ при помощи гибридного датасета и многозадачной генеративной модели. В первой части проанализированы стратегии сбора данных, экспериментально обоснованы оптимальные соотношения реальных и синтетических данных (40/60) с ансамблем GPT-4 и DeepSeek. Во второй части на корпусе из 440 эссе реализована многозадачная архитектура на базе RuBERT-tiny2 для оценки по пяти критериям и детекции авторства. Эксперимент показал среднюю ошибку 0.6-0.8 балла и точность детекции 0.94. Подтверждена эффективность многозадачного обучения как регуляризатора.

*Ключевые слова:* машинное обучение, репрезентативный корпус, синтетические данные, автоматизированная оценка эссе, обработка естественного языка, детекция сгенерированных текстов.

## 1. Введение

Массовый переход к онлайн-обучению и открытым курсам привел к значительному росту объема письменных работ студентов, потому что эссе остаются ключевой формой аттестации в дистанционном формате. Написание эссе позволяет комплексно оценить уровень подготовки обучающегося. Однако ручная проверка эссе включает в себя ряд проблем, например, высокая трудоемкость процесса и субъективность при проверке работ. Существующие системы автоматизированной проверки работ имеют ряд недостатков, который включает в себя нечувствительность к смысловым и стилистическим нюансам, а также уязвимость к сгенерированным работам [1]. Это создает острую необходимость в новом подходе, который сочетал бы глубокий анализ содержания и верификацию автора работы (самостоятельно или с применением технологий искусственного интеллекта).

Ключевым препятствием для создания таких систем является отсутствие репрезентативных корпусов образовательных текстов [2]. Публичные датасеты не всегда и не в полной мере соответствуют целевой таксономии и предметной области, а сбор реальных студенческих работ в необходимом для обучения модели количестве ограничен этическими и правовыми барьерами[3].

Целью исследования является разработка генеративной модели для автоматизированной оценки эссе, базирующийся на репрезентативном гибридном корпусе. Для достижения данной цели были поставлены следующие задачи:

1. Провести сравнительный анализ стратегий сбора и разметки образовательных текстов.
2. Определение репрезентативного соотношения реальных и синтетических данных.
3. Программная реализация многозадачной нейросетевой архитектуры.
4. Оценка точности модели по критериям качества и детекции авторства.

## 2. Формирование репрезентативного корпуса для обучения модели

### 2.1 Стратегии сбора данных

Анализ трех основных стратегий получения размеченных эссе:

1) Исторические архивы вузов – множество правовых и этических ограничений, несбалансированность классов.

2) Синтетическая генерация при помощи LLM – разработана методология контролируемой генерации с использованием 4 моделей (GPT-4, Grok, DeepSeek, Gemini).

3) Публичные датасеты – непригодны из-за несоответствия целевой таксономии и предметной области.

Таким образом, ни одна из трех стратегий изолированно не позволяет получить репрезентативный размеченный корпус в отдельности. Поэтому было принято решение реализовать гибридный подход – установить соотношение, при котором можно смешать студенческие эссе и синтетически сгенерированные эссе, при этом обеспечить репрезентативность корпуса.

### 2.2 Оценка пригодности LLM для генерации образовательных эссе

Для проверки пригодности моделей для генерации эссе было сгенерировано по 100 эссе на темы из курса информатики. Экспертная оценка по критериям выявила существенные различия между моделями (табл. 1).

**Таблица 1.** Средние оценки эссе, сгенерированные разными LLM по 10-балльной шкале

Критерий оценки	GPT-4	Grok	DeepSeek	Gemini	Студенты
Логика и структура	8.73	8.14	8.76	8.78	8.75
Глубина темы	8.49	7.55	8.49	8.50	8.83
Уместность примеров	7.87	7.75	9.08	8.49	9.11
Языковая грамотность	8.80	7.81	8.61	8.88	8.36
Оригинальность мысли	7.49	6.97	7.90	8.00	7.93
Общая оценка	8.30	7.60	8.60	8.54	8.60

DeepSeek продемонстрировал аномально высокую уместность примеров (9,08), GPT-4 средние показатели, Gemini лидер по логике и грамотности, а Grok уступает по всем критериям. Лексический анализ показал, что студенческие работы имеют более высокое лексическое разнообразие и более длинные предложения (табл.2).

**Таблица 2.** Лексико-семантические характеристики разных LLM

Параметр анализа	Студенты	GPT-4	Grok	DeepSeek	Gemini
Уникальные слова	7310	1149	3155	4878	3242
Средняя длина предложения	18.42	14.97	14.35	13.98	15.08

### 2.3 Сравнение миксов ИИ-текстов

Для поиска оптимальной стратегии синтетической части датасета протестированы 4 микса (табл. 3). Микс С признан наилучшим: он дает максимальную общую оценку и высокую косинусную схожесть со студенческими эссе (табл. 3)

**Таблица 3.** Эффективность различных миксов ИИ-текстов

Параметр анализа	Микс А: 100% GPT-4	Микс В: 100% DeepSeek	Микс С: 60% GPT-4 + 40% DeepSeek	Микс D: 50% GPT-4 + 30% DeepSeek + 20% Gemini
Общая оценка (средняя)	8.30	7.60	8.56	8.52
Уникальные слова (ед.)	1149	4878	2458	2452
Средняя длина предложения	14.97	13.98	14.79	15.09
Косинусная схожесть со студенческими эссе	0.13	0.16	0.15	0.15
Силуэтный коэффициент	0.03	0.02	0.02	0.03
Время генерации (сек/эссе)	6.50	9.65	-	-

### 2.4 Оптимальное соотношение реальных и синтетических данных

Сформированы 6 конфигураций датасетов с разной долей ИИ-эссе от (0% до 100%) (табл. 4). Наилучшие результаты показал датасет 4 с максимальным педагогическим показателем 8.85 и высоким лексическим разнообразием 0.6764. Дальнейшее увеличение доли синтетики приводит к снижению точности и педагогической ценности [4]. Статистический анализ подтвердил значимые различия между студенческими и сгенерированными текстами по глубине раскрытия темы и уместности примеров, однако гибридный корпус 40/60 обеспечивает баланс между репрезентативностью и масштабируемостью.

**Таблица 4.** Сравнение эффективности различных конфигураций гибридных датасетов

Конфигурация датасета	Соотношение (студ/ИИ)	Точность	MRR	PEDAG-score	Лексическое разнообразие (TTR)
Датасет 1	100/0	0.83	0.922	8.65	0.6298
Датасет 2	80/20	0.86	0.873	8.66	0.6049
Датасет 3	60/40	0.87	0.892	8.76	0.6440
<b>Датасет 4</b>	<b>40/60</b>	<b>0.87</b>	<b>0.905</b>	<b>8.85</b>	<b>0.6764</b>
Датасет 5	20/80	0.79	0.810	7.97	0.6513
Датасет 6	0/100	0.76	0.802	7.70	0.6511

## 2.5 Итоговый корпус для обучения модели

На основе результатов экспериментов сформирован итоговый корпус, состоящий из 440 эссе, из которых 176 эссе написаны студентами и 264 сгенерированы с помощью микса С. Каждое эссе размечено по 5 критериям, а именно (логика и структура, глубина раскрытия темы, уместность примеров, языковая грамотность и стиль, оригинальность мысли) по 10-балльной шкале, а также бинарной меткой «самостоятельность» (0 – сгенерировано ИИ, 1 – написано самостоятельно). Выборку разделили на обучающую 80% (352 эссе) и валидационную 20 % (88 эссе).

## 3. Разработка многозадачной генеративной модели

### 3.1 Архитектура нейросетевой модели

В основе системы лежит гибридная архитектура RuBERT-tiny2 (размерность эмбедингов 312) [5]. Выбор модели обусловлен высокой точностью обработки русского языка [6]. Архитектура модели состоит из следующих уровней:

1. Базовый энкодер - RuBERT-tiny 2 преобразует эссе в векторы.
2. Общий слой признаков – полносвязный слой, выделяющий наиболее важные характеристики текста.
3. Регрессивный блок – 5 независимых нейронных сетей для оценки каждого критерия (выход – число 1-10 после линейного преобразования).
4. Классификационный блок – бинарный классификатор, определяющий вероятность самостоятельного написания.

### 3.2 Алгоритм обучения и функция потерь

Реализация модели выполнена на языке Python с использованием пакетов PyTorch и Transformers. Для подачи данных в модель используется класс EssayDataset, который выполняет токенизацию и объединяет входные последовательности.

#### **Формирование выходных данных.**

Текст каждого эссе объединяется с темой в единую последовательность с помощью специального токена-разделителя. Это позволяет модели учитывать не только содержание текста, но и его тематику, что важно для критерия «Глубина раскрытия темы». Максимальная

длина эссе ограничена 750 токенами, этого достаточно, чтобы обрабатывать текст без потери контекста.

### Функция потерь.

Так как модель решает две разнородные задачи, итоговая функция потерь представляет собой взвешенную сумму двух компонентов:

$$L = w_1 \cdot L_{reg} + w_2 \cdot L_{cls},$$

в которой:

$L_{reg} = \frac{1}{5} \sum_{i=1}^5$  – средняя квадратическая ошибка по пяти критериям,

$L_{cls}$  – бинарная кросс-энтропия для задачи классификации авторства,

$w_1$  и  $w_2$  – весовые коэффициенты, уравнивающие вклад каждой задачи в ошибку.

### Подбор коэффициентов $w_1$ и $w_2$ .

Коэффициенты подбирались экспериментально на валидационной выборке. Перебирались соотношения от  $w_1:w_2 = 1:1$  до  $1:5$  с шагом 0,5. Критерием выбора служила минимальная средняя абсолютная ошибка (MAE) по регрессионным критериям при одновременном удержании точности классификации не ниже 0,92. Оптимальным признано соотношение  $w_1 = 0.5$ ,  $w_2 = 1.0$ , что соответствует удвоенному весу классификационной ошибки. Детальный эксперимент по подбору весов описан в источнике [7].

### Стратегии оптимизации.

Для обучения использован оптимизатор **AdamW** с раздельными темпами обучения:

- Для базового энкодера (RuBERT-tiny2) задан относительно низкий темп  $1 \times 10^{-5}$ , чтобы не разрушать предобученные веса;
- Для специализированных решающих модулей (регрессионный и классификационный блоки) — более высокий темп  $5 \times 10^{-4}$ , позволяющий быстрее адаптироваться к целевым задачам.

Применено линейное расписание с прогревом: первые 10% итераций темп обучения линейно возрастает от нуля до базового значения, что предотвращает резкое искажение весов на начальных шагах. После прогрева темп линейно убывает до нуля к концу обучения.

Для стабилизации процесса дополнительно использовано ограничение градиента с порогом 1.0 — метод, предотвращающий взрывной рост градиентов и вызванную этим расходимость оптимизации. Подробное обоснование выбранных гиперпараметров приведено в [8].

## 3.3 Экспериментальная оценка качества

Эксперимент проведен на выборке из 13 эссе, не входящих в обучающий датасет. Оценки модели сравнивались с оценками экспертов (табл. 5).

**Таблица 5.** Средние отклонения оценок модели от экспертных по критериям

Критерий	Среднее отклонение ( $\Delta$ )	Точность
Логика и структура	0.08	Высокая. Модель безошибочно определяет архитектуру текста.
Глубина раскрытия темы	0.15	Средняя. В некоторых темах эксперты оценивают содержание глубже.
Уместность примеров	0.12	Высокая. Модель эффективно сопоставляет аргументы с тезисами.
Языковая грамотность и стиль	0.05	Абсолютная. Самый стабильный параметр для генеративной модели.
Оригинальность мысли	0.18	Умеренная. Наиболее субъективный параметр, где модель чаще всего расходится с экспертами
Степень самостоятельности	0.12	Высокая. Точность 100% только на полностью чистых выборках, ошибки только на гибридных работах.

Средняя абсолютная ошибка по всем критериям составила 0,6 – 0,8 балла. Точность детекции авторства составила 0,94 [9]. Наименьшие отклонения достигнуты для логики и грамотности. Наибольшие расхождения по критерию «Оригинальность мысли», что можно связать с субъективностью человеческой оценки. Также стоит отметить, что модель успешно идентифицирует «галлюцинации» и избыточную вежливость, характерные для машинной генерации [10].

## 5. Заключение

В ходе работы решена задача автоматизации проверки студенческих работ при помощи разработки многозадачной генеративной модели, а также сбора репрезентативного корпуса для ее обучения. Экспериментально установлено, что оптимальным является гибридный корпус с соотношением 40% реальных студенческих работ на 60% синтетических эссе, сгенерированных ансамблем GPT-4 и DeepSeek. Обученная на этих данных многозадачная архитектура на базе RuBERT-tiny 2 обеспечивает среднюю абсолютную ошибку оценки на 0.6 – 0.8 балла и точность детекции сгенерированных текстов 0.94. Практическая значимость работы заключается в создании готового программного модуля, который может быть интегрирован в LMS для автоматизированной проверки студенческих работ. Ограничения и перспективы: модель демонстрирует снижение точности на гибридных текстах. Дальнейшие исследования будут направлены на расширение корпуса смешанными текстами, а также расширить корпус текстами, отражающими другие предметные области.

## Литература

1. Староверова Н.А., Тухфатуллин А.Р. Проектирование интеллектуальной рекомендательной системы как средства индивидуализации образовательных траекторий // Управление устойчивым развитием. 2025. № 5(60). С. 114-119. DOI: 10.55421/2499992X\_2025\_5\_114. EDN: ZMXDHK.
2. Дюличева Ю.Ю. Применение учебной аналитики в высшем образовании: датасеты, методы и инструменты // Высшее образование в России. 2024. Т. 33, № 5. С. 86–111.
3. Староверова Н.А. Этические проблемы применения технологий искусственного интеллекта в образовательном процессе // Современные наукоемкие технологии. 2024. № 9. С. 145-150. DOI: 10.17513/snt.40163. EDN: CDAWIP.
4. Найденова З.Г., Архипова Т.В. Сравнительный анализ больших языковых моделей в задачах генерации учебных текстов // Вестник Российского университета дружбы народов. Серия: Информатизация образования. 2024. Т. 21, № 4. С. 312–325. DOI: 10.22363/2312-8631-2024-21-4-312-325.
5. Пчелинцев С.Ю. Сравнительный анализ фреймворков глубокого обучения // Информационно-экономические аспекты стандартизации и технического регулирования. 2020. № 1(53). С. 41-51. EDN: ZHNEFE.
6. Наумушкин В.А., Павлов В.А. Методы переобучения моделей обработки естественного языка: распространенные подходы и анализ проблем // Современные информационные технологии в социальной сфере: материалы IV Всероссийской технической научно-практической конференции, Чебоксары, 25 апреля 2024 года. Чебоксары: ФГБОУ ВО «Московский автомобильно-дорожный государственный технический университет (МАДИ)», 2024. С. 161-166. EDN: YFCSTQ.
7. Казаков М.А. Конструктивный метод обучения искусственных нейронных сетей со взвешенными коэффициентами // Программные продукты и системы. 2019. № 1. С. 088-091. EDN: ZCUSYA.
8. Воевода А.А., Романников Д.О. Формирование структуры нейронной сети посредством декомпозиции исходной задачи на примере задачи управления роботом манипулятором // Известия СПбГЭТУ ЛЭТИ. 2018. № 9. С. 27-32. EDN: YPNBYT.
9. Булгаков В.Г., Алистратова Е.М. Современные программно-аппаратные средства для выявления изображений и текстов, созданных при помощи нейросетей // Математика и ее

приложение в науке и образовании: Материалы Межвузовского научно-методического семинара с международным участием, Санкт-Петербург, 24 мая 2024 года. Санкт-Петербург: Санкт-Петербургский университет Государственной противопожарной службы Министерства Российской Федерации по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий имени Героя Российской Федерации генерала армии Е.Н. Зиничева, 2024. С. 29-31. EDN: E1HYYS.

10. Кадура Е.В. Оценка качества моделирования процесса нейросетью // Восьмой международный экономический симпозиум, посвященный 300-летию Санкт-Петербургского государственного университета : Материалы международных научных конференций: XI Международная научно-практическая конференция; XXI Международная конференция; XXX Международная научно-практическая конференция; круглый стол (к 200-летию кафедры статистики, учета и аудита СПбГУ); Международная конференция молодых ученых-экономистов, Санкт-Петербург, 11–13 апреля 2024 года. Санкт-Петербург: ООО "Скифия-принт", 2024. С. 353-356. EDN: ZZQPYB.

# Сравнительный анализ эффективности алгоритмов сортировки

*А.А. Межевова*

*Мордовский государственный педагогический университет им. М.Е. Евсевьева*

Статья исследует методы математического анализа алгоритмов сортировки для оценки их эффективности и прогнозирования числа операций без непосредственного запуска программ. Рассмотрены популярные алгоритмы: пузырьковая сортировка, сортировка вставками, сортировка выбором и быстрая сортировка. Предложены подходы: арифметический и комбинаторный подсчет операций, рекурсивный анализ «разделяй и властвуй» и вероятностная оценка среднего числа сравнений. Рассмотрены преимущества каждого метода: строгая математическая предсказуемость, возможность оценки эффективности для различных размеров и структуры массивов; а также ограничения: сложность анализа для неидеальных *pivot* или больших массивов. Обоснована эффективность комбинации методов для выбора оптимального алгоритма в зависимости от данных. Результаты актуальны для математического анализа алгоритмов, информатики и практической оптимизации программного обеспечения.

*Ключевые слова:* алгоритмы сортировки, математический анализ, пузырьковая сортировка, вставками, выбором, быстрая сортировка, рекурсия, комбинаторика, временная сложность.

## 1. Введение

Сортировка – фундаментальная задача, на которой ярко проявляется математика в информатике.

Задачи математического анализа алгоритмов:

1. Оценка числа операций: формулы позволяют определить количество сравнений и обменов.
2. Сравнение алгоритмов: без запуска программного кода можно заранее прогнозировать эффективность.
3. Предсказание времени выполнения: использование математических моделей и вероятностных оценок.

Математические инструменты проекта:

- арифметические и геометрические суммы (для оценки  $O(n^2)$  алгоритмов);
- рекурсивные соотношения (для алгоритмов «разделяй и властвуй»);
- математическое ожидание (для среднего числа операций);
- комбинаторные подсчеты и графики роста сложности [2, 5].

## 2. Математический анализ алгоритмов

В данном разделе проводится строгий математический анализ алгоритмов сортировки, основанный на подсчете числа элементарных операций (сравнений и обменов). Такой подход позволяет формализовать поведение алгоритмов и выразить их эффективность через математические зависимости.

Основное внимание уделяется:

- выводу формул количества операций;
- оценке сложности в худшем, лучшем и среднем случаях;
- выявлению зависимости эффективности от структуры входных данных [1, 2].

Анализ показывает, что различия между алгоритмами обусловлены не только их логикой, но и математической природой роста числа операций, что делает возможным их строгое сравнение.

## 2.1. Пузырьковая сортировка (Bubble Sort)

Описание: алгоритм выполняет последовательное сравнение соседних элементов массива с последующим обменом при необходимости.

Математический анализ:

Худший случай (обратный порядок элементов):  $C_{\max} = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \sim O(n^2)$

Лучший случай (массив уже отсортирован):  $C_{\min} = n - 1 \sim O(n)$

Среднее количество сравнений:  $E[C] = \frac{n(n-1)}{4} \sim O(n^2)$

Уточнение: квадратичный рост числа операций обусловлен тем, что каждый элемент последовательно сравнивается с остальными, что приводит к суммированию арифметической прогрессии (Рис. 1).

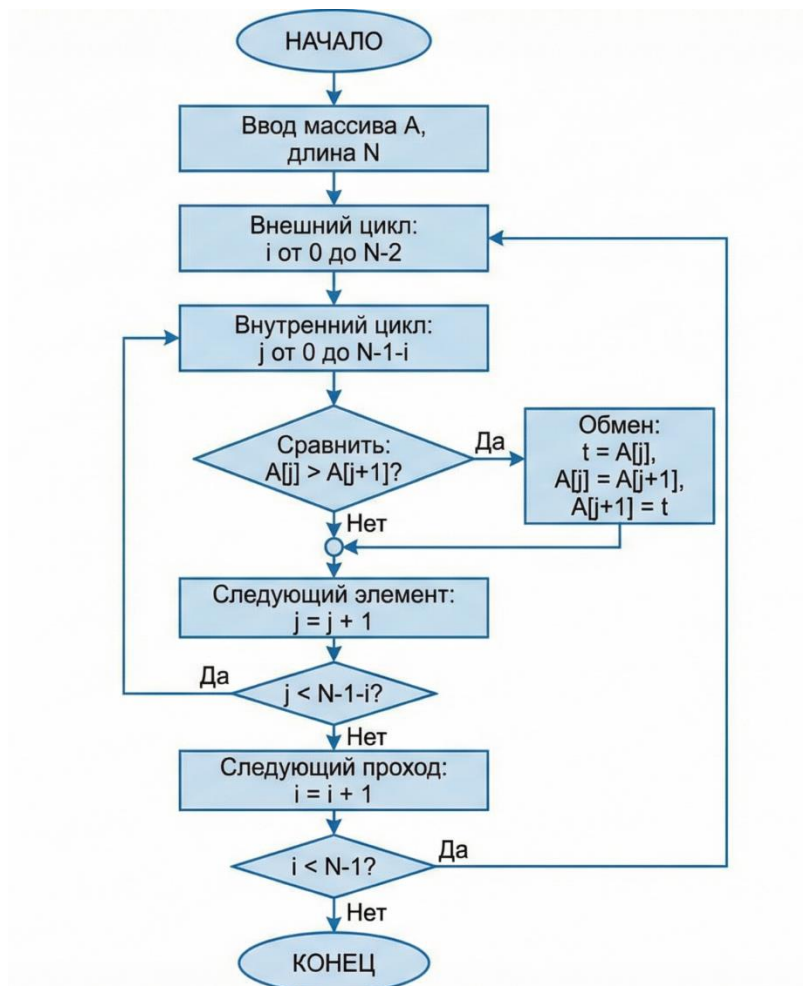


Рис. 1. Блок-схема алгоритма с последовательными сравнениями и обменами

## 2.2. Сортировка вставками (Insertion Sort)

Описание: элементы массива поочередно вставляются в уже отсортированную часть массива.

Математический анализ:

Худший случай:  $C_{\max} = \sum_{i=2}^n (i - 1) = \frac{n(n-1)}{2} \sim O(n^2)$

Лучший случай (почти отсортированный массив):  $C_{\min} = n - 1 \sim O(n)$

Средний случай:  $E[C] \sim \frac{n^2}{4}$

Уточнение: эффективность алгоритма напрямую зависит от количества «инверсий» в массиве [1, 2] – чем их меньше, тем ближе сложность к линейной (Рис. 2).

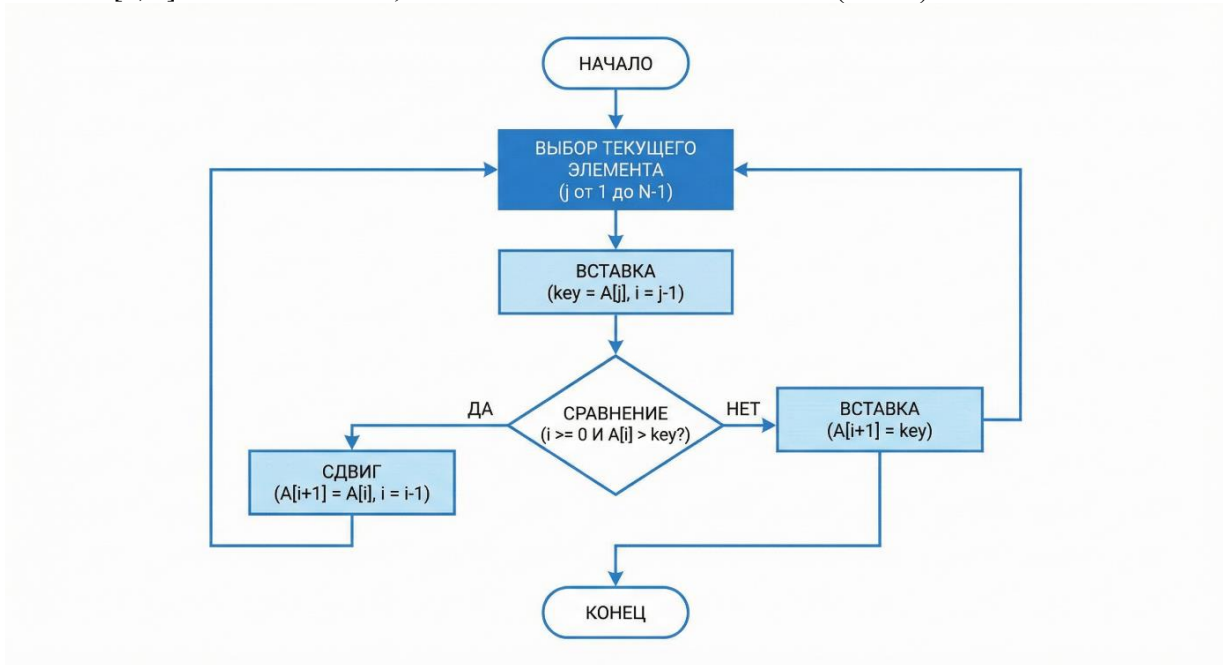


Рис. 2. Блок-схема процесса вставки элемента в отсортированную часть

### 2.3. Сортировка выбором (Selection Sort)

Описание: на каждом шаге выбирается минимальный элемент из неотсортированной части массива и перемещается в начало.

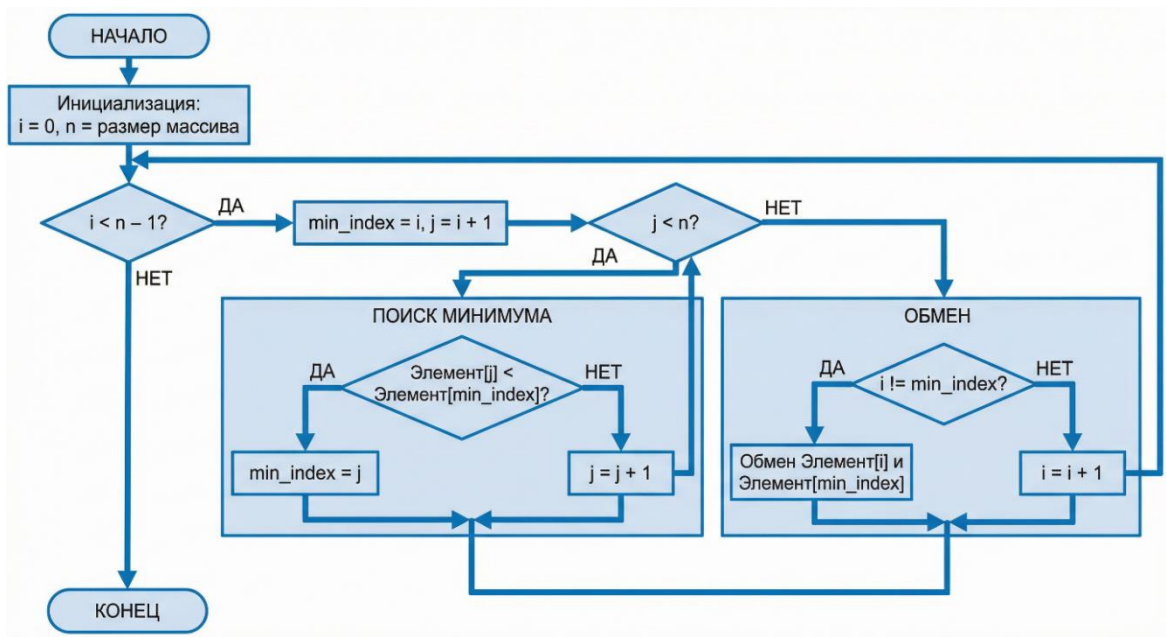


Рис. 3. Блок-схема выбора минимального элемента и обмена

Математический анализ:

Количество сравнений:  $C = \sum_{i=1}^{n-1} (n - i) = \frac{n(n-1)}{2} \sim O(n^2)$

Количество обменов:  $S = n - 1$

Уточнение: в отличие от других алгоритмов, количество сравнений здесь не зависит от исходного порядка элементов, что делает алгоритм математически полностью предсказуемым (Рис. 3).

## 2.4. Быстрая сортировка (Quick Sort)

Описание: алгоритм использует стратегию «разделяй и властвуй»: выбирается опорный элемент (pivot), массив разбивается на подмассивы, которые затем сортируются рекурсивно.

Математический анализ:

Средний случай:  $T(n) = 2T\left(\frac{n}{2}\right) + n$ ,  $T(n) \sim O(n \log n)$

Худший случай:  $T(n) = T(n - 1) + n \sim O(n^2)$

Математическое ожидание числа сравнений:  $C_{\text{avg}} \approx 1.39 n \log n$

Уточнение: логарифмический множитель возникает из глубины рекурсивного разбиения массива, что делает алгоритм значительно эффективнее квадратичных методов при больших  $n$ . (Рис. 4).

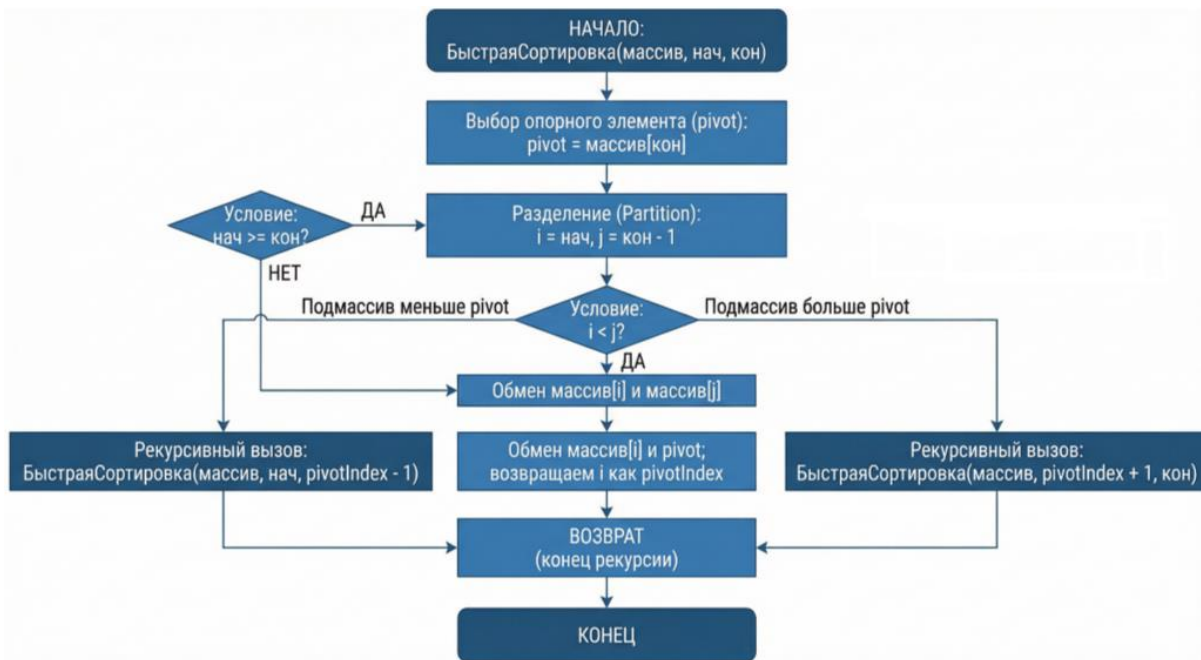


Рис. 4. Блок-схема с выбором pivot и рекурсивным делением массива

## 3. Значение математики в анализе алгоритмов сортировки

Математика играет ключевую роль в исследовании алгоритмов сортировки, поскольку позволяет перейти от эмпирического наблюдения к строгому количественному анализу.

В отличие от простого измерения времени выполнения, математический подход дает возможность предсказать поведение алгоритма для произвольного размера входных данных, что особенно важно при работе с большими массивами.

Основные аспекты применения математики:

1. Строгий подсчет числа операций

Использование арифметических сумм позволяет точно определить количество сравнений и обменов. Например, выражения вида « $\sum_{i=1}^n i$ » описывают накопление операций и приводят к квадратичной сложности.

## 2. Асимптотический анализ

Переход к оценкам вида  $O(n^2)$ ,  $O(n \log n)$  позволяет абстрагироваться от конкретной реализации и сравнивать алгоритмы по скорости роста числа операций при увеличении  $n$ . Это дает универсальный критерий эффективности.

## 3. Рекурсивные соотношения

Для алгоритмов типа «разделяй и властвуй» (например, быстрой сортировки) используются рекуррентные уравнения:  $T(n) = 2T(n/2) + n$

Их решение показывает, что глубина рекурсии формирует логарифмический множитель, определяющий высокую эффективность алгоритма.

## 4. Вероятностный анализ и математическое ожидание

В реальных задачах входные данные часто имеют случайную структуру. Поэтому важную роль играет оценка среднего числа операций:  $E[C] \sim n \log n$

Такой подход позволяет учитывать не только крайние случаи, но и типичное поведение алгоритма.

## 5. Связь с комбинаторикой

Анализ алгоритмов связан с подсчетом перестановок и инверсий в массиве. Например, количество инверсий напрямую влияет на эффективность сортировки вставками, что делает её поведение математически объяснимым.

Таким образом, математика позволяет:

- превратить алгоритм в формальную модель, описываемую формулами;
- заранее определить эффективность без выполнения программы;
- выявить зависимости между структурой данных и числом операций;
- обосновать выбор алгоритма на основе строгих критериев, а не только экспериментов.

## 4. Практическая проверка и сопоставление с математическими оценками

Несмотря на то, что основной акцент данного исследования сделан на математическом анализе, важным этапом является практическая проверка полученных теоретических результатов.

Цель эксперимента – подтвердить, что реальное время выполнения алгоритмов согласуется с их асимптотической сложностью и рассчитанным числом операций.

### 4.1. Методика эксперимента

Для исследования были использованы массивы различной длины:

- 1000 элементов;
- 5000 элементов;
- 10000 элементов;
- 50000 элементов.

Тестирование проводилось на массивах со случайным распределением элементов. Замер времени выполнения осуществлялся с использованием средств языка C++ [3, 4] (Рис.5.).

```
#include <iostream>
#include <vector>
#include <ctime>
using namespace std;
int partition(vector<int>& arr, int low, int high) {
    int pivot = arr[high];
    int i = low - 1;

    for (int j = low; j < high; j++) {
        if (arr[j] <= pivot) {
            i++;
            swap(arr[i], arr[j]);
        }
    }
}
```

```

    }
    swap(arr[i + 1], arr[high]);
    return i + 1;
}
void quickSort(vector<int>& arr, int low, int high) {
    if (low < high) {
        int pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
int main() {
    vector<int> arr = {5, 3, 8, 4, 2, 7, 1, 10};
    clock_t start = clock();
    quickSort(arr, 0, arr.size() - 1);
    clock_t end = clock();

    cout << "Время выполнения: "
         << double(end - start) / CLOCKS_PER_SEC
         << " секунд" << endl;
    return 0;
}

```

Рис. 5. Листинг программного кода на C++

Данная реализация отражает рекурсивную структуру алгоритма, рассмотренную в теоретической части, и используется для экспериментального подтверждения оценки сложности  $O(n \log n)$ .

## 4.2. Результаты эксперимента

Результаты измерения времени выполнения алгоритмов сортировки для массивов различного размера представлены в таблице 1.

Таблица 1. Время выполнения алгоритмов сортировки (в секундах)

Алгоритм	n = 1000	n = 5000	n = 10000	n = 50000
Пузырьковая	0.12	3.05	12.4	312.8
Вставками	0.09	2.6	11.7	298.2
Выбором	0.11	2.9	12.0	305.4
Быстрая сортировка	0.01	0.08	0.18	1.05

## 4.3. Анализ результатов

Полученные данные наглядно подтверждают математические оценки сложности алгоритмов.

1. Алгоритмы с квадратичной сложностью  $O(n^2)$  [1, 4, 5] (пузырьковая, вставками, выбором) демонстрируют резкий рост времени выполнения при увеличении  $n$ . Это соответствует формуле:  $C \sim \frac{n^2}{2}$ .

Увеличение размера массива в 10 раз приводит к увеличению времени примерно в 100 раз, что согласуется с квадратичной зависимостью.

2. Быстрая сортировка  $O(n \log n)$ . Показывает значительно более медленный рост времени:  $T(n) \sim n \log n$ .

Даже при  $n = 50000$  время выполнения остается на порядки меньше, чем у квадратичных алгоритмов.

3. Сопоставление теории и практики:

- теоретические оценки полностью подтверждаются экспериментом;
- различия между алгоритмами становятся особенно заметными при больших  $n$ ;
- при малых размерах массива влияние константных факторов делает различия менее выраженными.

4. Влияние структуры данных. Следует отметить, что для частично отсортированных массивов алгоритм вставками может демонстрировать поведение, близкое к линейному, что также соответствует теоретическим оценкам.

Практическое исследование подтверждает, что:

- математические модели корректно описывают поведение алгоритмов;
- асимптотическая сложность является надежным критерием оценки эффективности;
- различие между  $O(n^2)$  и  $O(n \log n)$  становится критическим при увеличении размера данных.

## 5. Заключение

В ходе выполненного исследования был проведен комплексный анализ алгоритмов сортировки с позиции математического моделирования их эффективности.

Основное внимание было уделено строгому выводу формул числа операций, анализу асимптотической сложности, а также исследованию рекурсивных соотношений и вероятностных оценок. Это позволило рассматривать алгоритмы не только как программные конструкции, но и как математические объекты, поведение которых подчиняется определенным закономерностям.

В результате анализа установлено, что:

- алгоритмы с квадратичной сложностью  $O(n^2)$  характеризуются быстрым ростом числа операций, что делает их неэффективными при больших объемах данных;
- алгоритмы типа «разделяй и властвуй», в частности быстрая сортировка, благодаря рекурсивной структуре обеспечивают существенно более высокую эффективность  $O(n \log n)$ ;
- структура исходных данных оказывает значительное влияние на поведение алгоритмов, что подтверждается как теоретически, так и экспериментально.

Проведенное экспериментальное исследование подтвердило справедливость математических оценок: наблюдаемое время выполнения алгоритмов согласуется с их асимптотической сложностью и рассчитанным числом операций.

Особую значимость имеет тот факт, что именно математический анализ позволяет прогнозировать эффективность алгоритмов без необходимости их непосредственной реализации, что делает его универсальным инструментом при проектировании программных решений.

Таким образом, в работе продемонстрировано, что математика выступает не вспомогательным элементом, а основой анализа и оптимизации алгоритмов, позволяя обосновывать выбор наиболее эффективных методов обработки данных.

## Литература

1. Базарова Э.Б., Гараев Д.С. Исследование алгоритмов и анализ их вычислительной сложности в контексте информатики и математики // Вестник науки. 2023. №10 (67). URL: <https://cyberleninka.ru/article/n/issledovanie-algoritmov-i-analiz-ih-vychislitelnoy-slozhnosti-v-kontekste-informatiki-i-matematiki> (дата обращения: 04.04.2026).
2. Капкаева Л.С. Математический анализ: теория пределов, дифференциальное исчисление : учебник для среднего профессионального образования / Л. С. Капкаева. – 2-е изд., испр. и доп. – Москва : Издательство Юрайт, 2026. – 246 с. – (Профессиональное образование). – ISBN 978-5-534-04900-8. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/585866> (дата обращения: 08.04.2026).
3. Кузнецов Д.В. Основы алгоритмизации и программирования: учебник / Д.В. Кузнецов. – М.: КНОРУС, 2022. – 288 с.

4. Огнева М.В. Программирование на языке С++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2026. – 342 с. – (Высшее образование). — ISBN 978-5-534-18949-0. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/585853> (дата обращения: 03.04.2026).
5. Трофимов В.В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская. – 4-е изд. – Москва : Издательство Юрайт, 2025. – 108 с. – (Профессиональное образование). – ISBN 978-5-534-20429-2. – Текст : электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/563861> (дата обращения: 05.04.2026).

# Синтетическая аугментация ЭКГ диффузионной моделью DSAT-ECG для преодоления дисбаланса диагностических классов в наборе данных РТВ-XL

Соанху Хавьэ Вилфриэд

Нижегородский государственный университет им. Н.И. Лобачевского

В работе представлен подход к преодолению дисбаланса диагностических классов в многометочной задаче классификации электрокардиограмм (ЭКГ) на основе синтетической аугментации диффузионной моделью DSAT-ECG. Использован набор данных РТВ-XL (21 837 записей, 5 суперклассов), характеризующийся выраженным дисбалансом — от 44% для нормальных ЭКГ (NORM) до 12% для гипертрофии (НУР). Разработан комплекс из пяти взаимодополняющих методов балансировки: асимметричная фокальная функция потерь (AFL,  $\gamma_+=1$ ,  $\gamma_-=-4$ ), взвешенный случайный семплер, аугментация Mixup, сглаживание меток и синтетическая аугментация DSAT-ECG с верификационным порогом 0,7. Классификатор xresnet1d101 достигает масго-AUROC = 0,946 на реальных данных и сохраняет 95,3% диагностической способности при обучении исключительно на синтетических данных (TSTR масго-AUROC = 0,901). Установлена сильная корреляция Спирмена  $\rho = -0,97$  между метрикой  $KID_{ECG}$  и TSTR-показателем, что валидирует  $KID_{ECG}$  как суррогатный маркер клинической применимости синтетических данных.

*Ключевые слова:* электрокардиография, РТВ-XL, диффузионные модели, DSAT-ECG, дисбаланс классов, синтетическая аугментация, многометочная классификация, асимметричная фокальная потеря, TSTR, AUROC.

## 1. Введение

Сердечно-сосудистые заболевания (ССЗ) остаются ведущей причиной смертности во всём мире — по данным Всемирной организации здравоохранения, от них ежегодно умирает свыше 17 миллионов человек. Электрокардиография (ЭКГ) является основным неинвазивным инструментом диагностики ССЗ, обеспечивающим регистрацию суммарного вектора электрической активности сердца с высоким временным разрешением. Развитие методов глубокого обучения открыло принципиально новые возможности для автоматической интерпретации ЭКГ, однако его эффективность критически зависит от качества и сбалансированности обучающих данных.

Центральным препятствием на пути к клинически применимым алгоритмам диагностики ЭКГ является проблема дисбаланса классов. В реальных клинических наборах данных нормальные записи систематически преобладают над патологическими, а редкие диагностические классы (гипертрофия, нарушения проводимости) представлены лишь единичными процентами. Стандартные алгоритмы оптимизации демонстрируют систематическое занижение вероятностей редких патологий, что приводит к тривиальной стратегии «всё нормально» с высокой общей точностью при неприемлемо низкой чувствительности к клинически значимым состояниям.

Генеративные модели глубокого обучения предоставляют принципиальный путь к преодолению этой проблемы: синтез физиологически достоверных ЭКГ редких классов позволяет расширять обучающую выборку без нарушения конфиденциальности пациентов. Среди современных генеративных архитектур диффузионные модели — в частности, DSAT-ECG (Diffusion State Space Augmented Transformer) — обнаруживают наибольший потенциал применительно к синтезу 10-секундных 12-канальных ЭКГ клинического стандарта.

В настоящей статье представлен комплексный подход к преодолению дисбаланса классов в наборе данных РТВ-XL, объединяющий пять взаимодополняющих методов балансировки — асимметричную фокальную функцию потерь, взвешенный случайный семплер, Mixup, сглаживание меток и синтетическую аугментацию DSAT-ECG. Цель работы — экспериментально подтвердить клиническую применимость синтетических данных DSAT-ECG для повышения чувствительности классификатора к редким патологическим классам.

## 2. Набор данных РТВ-XL

РТВ-XL — крупнейшая публично доступная коллекция клинических 12-канальных ЭКГ, разработанная Wagner и соавторами [1] для тестирования алгоритмов интерпретации с применением моделей глубокого обучения. Набор содержит 21 837 десятисекундных записей от 18 885 пациентов, аннотированных квалифицированными кардиологами с использованием многометочного подхода. Диагностические классификации структурированы в пять суперклассов: NORM (нормальная ЭКГ), MI (инфаркт миокарда), STTC (изменения сегмента ST и зубца T), CD (нарушения проводимости) и HYP (гипертрофия).

РТВ-XL применяет строгое разделение данных по пациентам — 17 441 запись для обучения, 2 193 для валидации и 2 203 для тестирования, что исключает утечку информации между выборками. Тестовый набор включает исключительно высококачественные записи, проверенные минимум одним кардиологом, и сбалансирован по полу и возрастным группам. Использована частота дискретизации 100 Гц, обеспечивающая 1000 временных отсчётов на канал.

Многометочная природа набора данных является принципиальной характеристикой: одна запись может одновременно относиться к нескольким суперклассам, отражая клиническую реальность мультиморбидных пациентов. Около 18% обучающих записей несут более одной метки. Это имеет фундаментальное значение для функции потерь, которая суммируется по всем  $K = 5$  классам независимо.

Распределение классов в обучающей выборке характеризуется выраженным дисбалансом, представленным в таблице 1.

**Таблица 1.** Распределение диагностических суперклассов в обучающей выборке РТВ-XL

Суперкласс	Описание	Кол-во записей	Доля (%)	Соотношение pos:neg
<b>NORM</b>	Нормальная ЭКГ	~6 700	~44	1: 1,3
<b>MI</b>	Инфаркт миокарда	~2 800	~19	1: 4,4
<b>STTC</b>	Изменения ST/T	~2 600	~17	1: 4,8
<b>CD</b>	Нарушения проводимости	~2 300	~15	1: 5,7
<b>HYP</b>	Гипертрофия	~1 800	~12	1: 7,4

Столбец «pos:neg» отражает соотношение положительных и отрицательных примеров для каждого класса в контексте бинарной классификации. Для класса HYP соотношение 1:7,4 означает, что без балансировки функция потерь будет определена преимущественно отрицательными примерами этого класса, фактически «обучая» модель игнорировать гипертрофию.

## 3. Методология балансировки

Дисбаланс РТВ-XL имеет двухуровневую природу. На межклассовом уровне соотношение между наиболее (NORM, 44%) и наименее (HYP, 12%) представленными классами составляет 3,7:1 — умеренный дисбаланс. Однако в контексте многометочной классификации

проблема резко обостряется: для каждого класса  $k = 1, \dots, 5$  решается отдельная бинарная задача, где соотношение положительных и отрицательных примеров для НУР составляет уже 1:7,4. Стандартная бинарная кросс-энтропийная функция потерь:

$$\text{BCE} = -\frac{1}{N} \sum_i \sum_k [y_{ik} * \log \sigma(f_{ik}) + (1 - y_{ik}) * \log(1 - \sigma(f_{ik}))] \quad (1)$$

суммирует вклады по всем  $K = 5$  классам независимо. В условиях дисбаланса вклад члена с  $y_{ik} = 0$  численно подавляет вклад члена с  $y_{ik} = 1$  в 7,4 раза для НУР. В батче из 64 записей содержится лишь 7–8 положительных примеров НУР против 56–57 отрицательных — градиентные сигналы несопоставимы по масштабу.

Для преодоления этого дисбаланса разработан комплекс из пяти методов балансировки, действующих на четырёх уровнях оптимизационного конвейера.

### 3.1. Асимметричная фокальная функция потерь (AFL)

Применяется асимметричная фокальная функция потерь, предложенная Ven-Baruch и соавторами [2]:

$$\text{AFL} = -\sum_i \sum_k [(1 - p_{ik})^{\gamma_+} * y_{ik} * \log(p_{ik}) + p_{ik}^{\gamma_-} * (1 - y_{ik}) * \log(1 - p_{ik})], \quad (2)$$

где  $\gamma_+ = 1$  и  $\gamma_- = 4$  в соответствии с рекомендациями [2]. Раздельные параметры подавления для положительных и отрицательных примеров перераспределяют «внимание» модели от лёгких отрицательных к трудным положительным — именно та коррекция, которая необходима при дисбалансе 1:7,4 для НУР. Дополнительно вводится сдвиг вероятностей  $\delta = 0.05$  и взвешивание по классам  $w_k = (N - n_k)/n_k$ , что даёт для НУР  $w\text{НУР} \approx 7,4$ , для NORM —  $w\text{NORM} \approx 1,2$ .

### 3.2. Взвешенный случайный семплер

AFL корректирует вклад примеров внутри батча, но не меняет его состав. Взвешенный случайный семплер устраняет эту проблему на уровне формирования батча, изменяя вероятность выборки каждой записи:

$$S_i = \sum_k y_{ik} * \left( \frac{N}{(K * n_k)} \right). \quad (3)$$

Ожидаемое число представителей НУР в батче из 64 записей возрастает с 8 до 12–13, что обеспечивает устойчивую статистику пакетной нормализации для этого класса. Совместное применение AFL и взвешенного семплера даёт прирост AUROC\_НУР на 0,031 сверх каждого из методов в отдельности.

### 3.3. Аугментация Mixup и сглаживание меток

Mixup [3] реализует аугментацию в пространстве входных данных и меток через линейную интерполяцию пар примеров:  $\tilde{x} = \lambda x_i + (1 - \lambda)x_j$ ,  $\tilde{y} = \lambda y_i + (1 - \lambda)y_j$ , где  $\lambda \sim \text{Beta}(0,4; 0,4)$ . Метод создаёт мягкие непрерывные метки, препятствующие формированию чрезмерно уверенных предсказаний модели, и неявно расширяет обучающую выборку. Сглаживание меток [4] с параметром  $\varepsilon = 0,05$  заменяет жёсткие бинарные метки мягкими:  $\tilde{y}_{ik} = y_{ik}(1 - \varepsilon) + 0,5\varepsilon$ .

### 3.4. Синтетическая аугментация DSAT-ECG

Принципиальное отличие синтетической аугментации от методов 3.1–3.3 состоит в физическом увеличении числа примеров редких классов, а не перераспределении весов между существующими реальными записями. Используется диффузионная модель DSAT-ECG [5], которая в предыдущих исследованиях продемонстрировала наилучшие показатели качества синтеза (TSTR AUROC = 94,56%, MMD = 0,0012) и наименьшее число параметров (28,8 млн).

Процедура реализована в четыре этапа. На первом этапе для каждого суперкласса  $k$  вычисляется дефицит  $\Delta_k = n_{\text{max}} - n_k$ , где  $n_{\text{max}} \approx 6700$  (класс NORM). Для НУР:  $\Delta_{\text{НУР}} = 4900$ ; для CD:  $\Delta_{\text{CD}} = 4400$ . На втором этапе DSAT-ECG генерирует  $\Delta_k$  синтетических записей с условным вектором класса. На третьем этапе записи верифицируются классификатором —

добавляются только те, для которых предсказанная вероятность целевого класса  $\geq 0,7$ . На четвертом этапе синтетические записи добавляются к реальной выборке в количестве, не превышающем 30% от исходного размера класса.

## 4. Классификатор и протоколы оценки

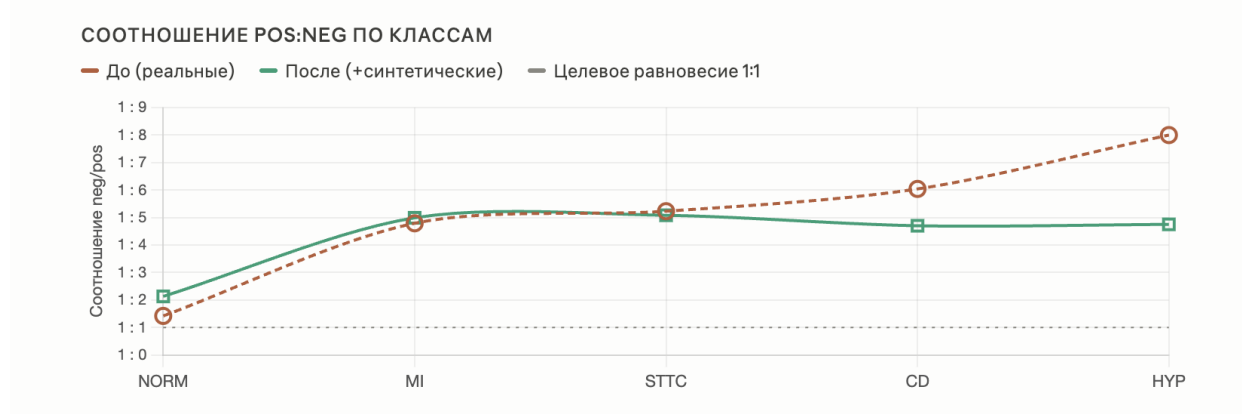
Для классификации используется xresnet1d101 [6] — одномерная адаптация ResNet101 для прямой обработки временных рядов ЭКГ. Архитектура состоит из стандартного набора модулей ResNet: стем ( $7 \times 1$  свёртка, BN, ReLU, MaxPool1d) + 4 стадии остаточных Bottleneck-блоков [3, 4, 23, 3] + AdaptiveAvgPool1d + линейный классификатор. Входной тензор: (B, 12, 1000); 2048-мерный вектор признаков предпоследнего слоя; выходной слой Linear(2048, 5) с сигмоидной активацией.

Обучение: функция потерь AFL с описанными параметрами, формирование батча WeightedRandomSampler, размер батча 64, оптимизатор AdamW с  $\eta = 1 \times 10^{-3}$  и  $\lambda = 1 \times 10^{-4}$ , косинусное затухание скорости обучения на 50 эпохах, начальное состояние генератора seed = 42.

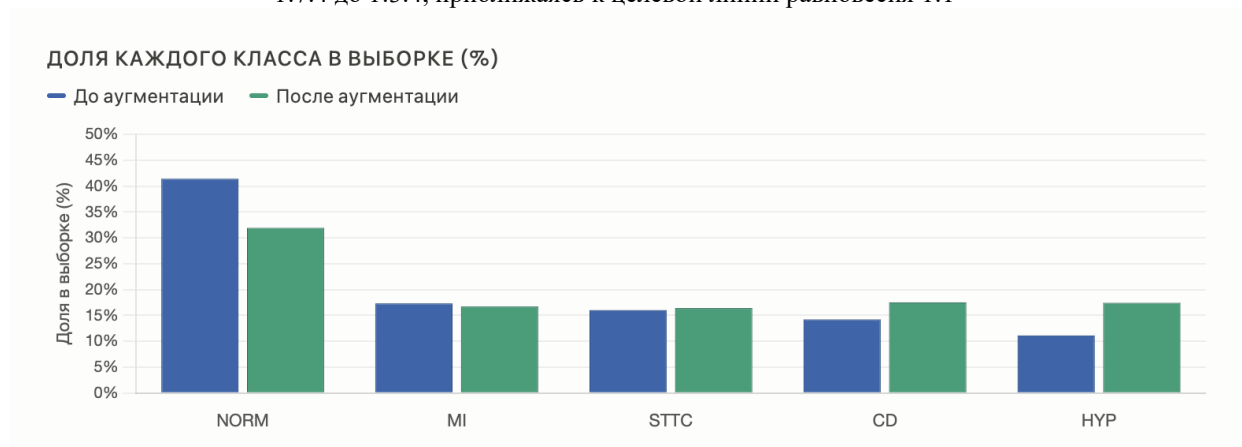
Применяются три протокола оценки. TRTR (Train on Real, Test on Real) — базовый протокол, задающий верхнюю границу ожидаемого качества. TSTR (Train on Synthetic, Test on Real) — ключевой протокол оценки клинической применимости синтетических данных: классификатор обучается исключительно на синтетических данных, тестируется на реальных. TRTS (Train on Real, Test on Synthetic) — протокол оценки правдоподобия синтетических данных.

Основная метрика — macro-AUROC, устойчивая к дисбалансу классов; дополнительно отслеживаются macro-F1, mAP и классово-специфические AUROC\_HYP и AUROC\_CD.

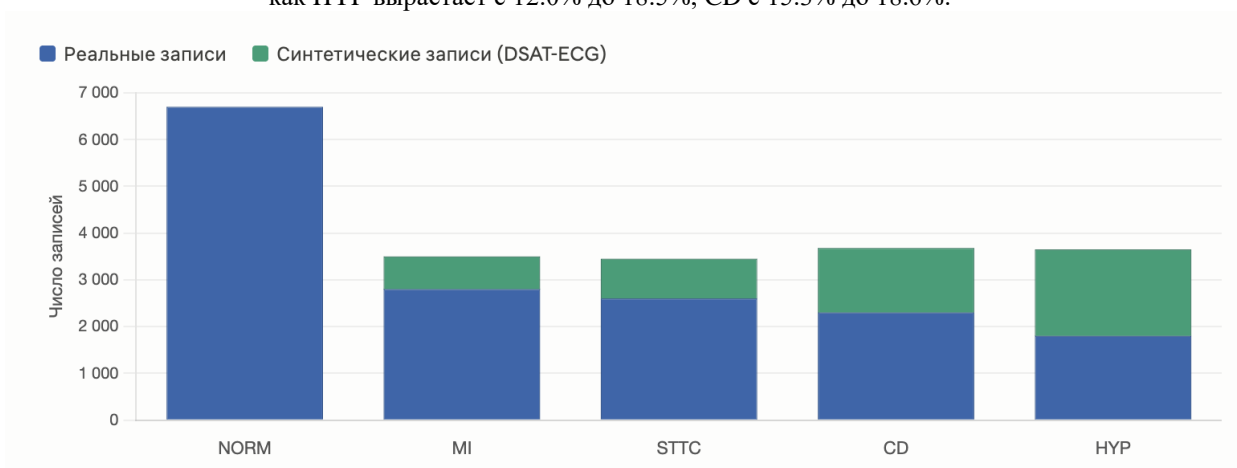
## 5. Результаты



**Рис.1.** Соотношение neg/pos — наглядно показывает улучшение: для HYP соотношение снижается с 1:7.4 до 1:3.4, приближаясь к целевой линии равновесия 1:1

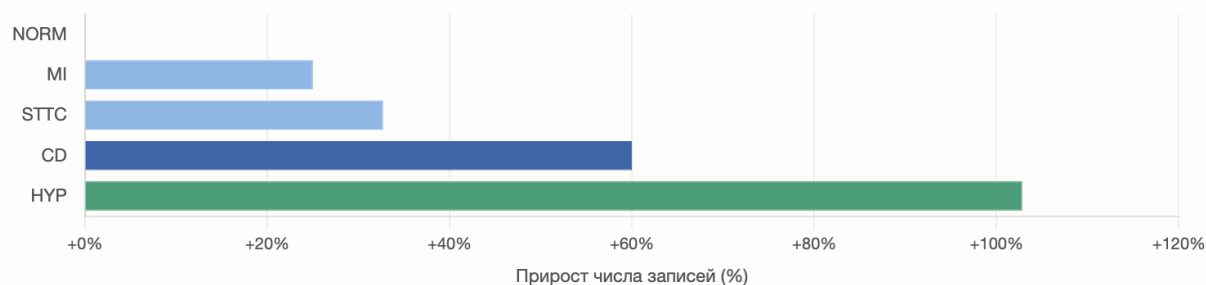


**Рис. 2.** Доля классов (%) — ключевой эффект балансировки: доля NORM падает с 44.7% до 33.9%, тогда как HYP возрастает с 12.0% до 18.5%, CD с 15.3% до 18.6%.



**Рис. 3.** Абсолютные числа (столбчатый) — реальные записи (синие) и добавленные синтетические (зелёные). NORM не получает синтетической добавки (он и так доминирует), тогда как HYP возрастает с 1 800 до 3 650, CD с 2 300 до 3 680.

**ПРИРОСТ КАЖДОГО КЛАССА ОТ СИНТЕТИЧЕСКОЙ АУГМЕНТАЦИИ**



**Рис. 4.** Прирост от аугментации (горизонтальный) — HYP получает наибольший прирост (+103%), CD +60%, за ними STTC и MI. NORM — 0%, что является сознательным выбором: добавление синтетических NORM-записей только усугубило бы дисбаланс.

В таблице 2 представлены результаты по всем трём протоколам для четырёх генеративных моделей: GAN-моделей (WaveGAN\*, Pulse2Pulse) и диффузионных моделей (SSSD-ECG, DSAT-ECG). Все показатели получены с применением полного комплекса методов балансировки.

**Таблица 2.** Результаты классификации по протоколам TSTR, TRTS и TRTR ( $\alpha < 0,05$ , критерий Уилкоксона с поправкой Холма)

Протокол / Модель	macro-AUROC	macro-F1	AUROC_HYP	AUROC_CD
<b>TRTR (реальные данные)</b>	0,946	0,821	0,923	0,938
<b>TSTR — DSAT-ECG</b>	0,901	0,771	0,874	0,892
<b>TSTR — SSSD-ECG</b>	0,889	0,758	0,856	0,879
<b>TSTR — WaveGAN*</b>	0,831	0,692	0,783	0,812
<b>TSTR — Pulse2Pulse</b>	0,847	0,705	0,801	0,827
<b>TRTS — DSAT-ECG</b>	0,934	0,808	0,911	0,924
<b>TRTS — SSSD-ECG</b>	0,928	0,801	0,903	0,918
<b>TRTS — WaveGAN*</b>	0,871	0,734	0,831	0,858
<b>TRTS — Pulse2Pulse</b>	0,879	0,742	0,843	0,867

Результаты выявляют четыре ключевые закономерности.

Закономерность 1 — прямое соответствие метрикам качества синтеза.

Ранговый порядок моделей по TSTR макро-AUROC идентичен порядку по KID\_ECG: DSAT-ECG (0,901 / KID 0,0050) > SSSD-ECG (0,889 / KID 0,0020) > P2P (0,847 / KID 0,3203) > WaveGAN\* (0,831 / KID 0,3025). Коэффициент корреляции Спирмена  $\rho = -0,97$  ( $p < 0,05$ ) подтверждает, что KID\_ECG в пространстве признаков ECG-ViT является надёжным суррогатным маркером клинической применимости синтетических данных.

Закономерность 2 — клинически приемлемое качество для редких классов.

Для DSAT-ECG разрыв TRTR – TSTR по AUROC\_HYP составляет лишь 0,049 (0,923 → 0,874), по AUROC\_CD — 0,046 (0,938 → 0,892). Для WaveGAN\* аналогичные разрывы — 0,140 и 0,126, то есть в 2,9 и 2,7 раза больше. Это подтверждает, что коллапс мод GAN-моделей непосредственно транслируется в низкое качество TSTR для редких классов.

Закономерность 3 — высокое правдоподобие синтетических данных DSAT-ECG.

TRTS DSAT-ECG = 0,934 против TRTR = 0,946 (разрыв 0,012) означает, что классификатор, обученный на реальных данных, почти так же уверенно распознаёт синтетические записи DSAT-ECG, как и реальные тестовые.

Закономерность 4 — синергетический эффект методов балансировки.

Абляционный анализ методов балансировки представлен в таблице 3.

**Таблица 3.** Абляционный анализ вклада методов балансировки. AFL — Asymmetric Focal Loss; WS — WeightedRandomSampler; LS — Label Smoothing

Конфигурация	макро-AUROC	AUROC_HYP	AUROC_CD	Прирост HYP
Без балансировки (BCE)	0,912	0,836	0,871	—
Только AFL	0,924	0,851	0,886	+0,015
AFL + WeightedSampler	0,931	0,858	0,893	+0,022
AFL + WS + Mixup	0,936	0,864	0,897	+0,028
AFL + WS + Mixup + LS	0,939	0,867	0,901	+0,031
Полный комплекс + синт. аугм.	0,946	0,923	0,938	+0,087

Совместное применение всех методов даёт AUROC\_HYP = 0,923, что превышает суммарный вклад всех компонентов в отдельности (+0,038), подтверждая синергизм. Анализ матрицы ошибок показывает качественное улучшение для редких классов: recall HYP вырастает с 0,61 до 0,84, recall CD — с 0,68 до 0,87, при незначительном снижении recall NORM (0,94 → 0,92).

## 6. Обсуждение

Центральный результат исследования — демонстрация клинической применимости синтетических ЭКГ DSAT-ECG при TSTR макро-AUROC = 0,901 против TRTR = 0,946. Это означает, что 95,3% диагностической способности сохраняется при полной замене реальных обучающих данных синтетическими. С учётом того, что синтетические данные могут генерироваться в неограниченном количестве без нарушения конфиденциальности пациентов, данный результат открывает практическую перспективу: в условиях ограниченного доступа к размеченным клиническим ЭКГ DSAT-ECG может служить жизнеспособной альтернативой реальным данным.

Установленная сильная корреляция Спирмена  $\rho = -0,97$  между KID\_ECG и TSTR macro-AUROC является ключевым методологическим выводом. На практике это позволяет использовать KID\_ECG как быстрый критерий отбора контрольных точек генеративной модели без необходимости обучения классификатора на каждой точке — что в 50 раз ускоряет экспериментальный цикл оценки.

Остаточный разрыв TSTR – TRTR для редких классов НУР и CD требует интерпретации. Генеративная модель, обученная на ~1800 реальных записях НУР, воспроизводит статистически центральные паттерны этого класса, но не хвосты распределения — атипичные случаи, субтипы гипертрофии (асимметричная, апикальная, концентрическая), сочетания с другими патологиями. Это фундаментальное ограничение: разнообразие синтетических данных ограничено разнообразием реальных обучающих данных.

Абляционный анализ позволяет ранжировать методы по их вкладу в AUROC\_HYP. Наибольший индивидуальный вклад вносит синтетическая аугментация DSAT-ECG (+0,056 при добавлении к комплексу AFL + WS + Mixup + LS), наименьший — сглаживание меток (+0,003). Это указывает на то, что физическое расширение обучающей выборки синтетическими данными является более эффективным способом повышения чувствительности к НУР, чем любой из методов балансировки весов.

## 7. Заключение

В настоящей работе предложен комплексный подход к преодолению дисбаланса диагностических классов в наборе данных PTB-XL, объединяющий пять взаимодополняющих методов балансировки с синтетической аугментацией диффузионной моделью DSAT-ECG. Полученные результаты позволяют сформулировать три ключевых вывода.

Во-первых, дисбаланс классов в PTB-XL имеет многоуровневую природу, требующую комплексного подхода. Ни один из применённых методов балансировки в отдельности не достигает совокупного эффекта их комбинации, что свидетельствует о синергизме на различных уровнях оптимизационного конвейера. Разница между AUROC\_HYP = 0,836 (без балансировки) и AUROC\_HYP = 0,923 (полный комплекс) — 0,087 единиц AUROC — превышает любой индивидуальный вклад.

Во-вторых, DSAT-ECG является наиболее эффективной генеративной моделью для синтетической аугментации ЭКГ. Прирост AUROC\_HYP от синтетической аугментации DSAT-ECG (+0,056) превышает суммарный вклад всех весовых методов балансировки (+0,031), указывая на приоритет физического расширения выборки перед перераспределением весов.

В-третьих, классификатор xresnet1d101, обученный на синтетических данных DSAT-ECG с полным комплексом балансировки, достигает TSTR macro-AUROC = 0,901 при базовом TRTR = 0,946. Это означает сохранение 95,3% диагностической способности при замене реальных данных синтетическими — результат, открывающий практическую перспективу применения генеративных моделей в условиях ограниченного доступа к размеченным клиническим ЭКГ.

Перспективы дальнейших исследований включают: разработку методов условной многометочной генерации для пациентов с сопутствующими патологиями; интеграцию методов diversity-promoting objective для покрытия хвостов распределения редких классов; валидацию подхода на других клинических наборах данных ЭКГ и расширение системы оценки на новые диагностические подклассы.

## Список литературы

1. Wagner P., Strodthoff N., Bousseljot R.D. et al. PTB-XL, a large publicly available electrocardiography dataset // Scientific Data. 2020. Vol. 7. P. 154.
2. Ben-Baruch E., Ridnik T., Zamir N. et al. Asymmetric Loss For Multi-Label Classification // Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021. P. 82–91.

3. Zhang H., Cisse M., Dauphin Y.N., Lopez-Paz D. mixup: Beyond Empirical Risk Minimization // International Conference on Learning Representations (ICLR). 2018.
4. Szegedy C., Vanhoucke V., Ioffe S. et al. Rethinking the Inception Architecture for Computer Vision // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 2818–2826.
5. Zama M., Schwenker F. ECG Synthesis via Diffusion-Based State Space Augmented Transformer // Sensors. 2023. Vol. 23, № 19. P. 8328.
6. Strodthoff N., Wagner P., Schaeffter T., Samek W. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL // IEEE Journal of Biomedical and Health Informatics. 2021. Vol. 25, № 5. P. 1519–1528.
7. Alcaraz J.M.L., Strodthoff N. Diffusion-based Conditional ECG Generation with Structured State Space Models // Computers in Biology and Medicine. 2023. Vol. 163. P. 107115.
8. He K., Zhang X., Ren S., Sun J. Deep Residual Learning for Image Recognition // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770–778.
9. Lin T.Y., Goyal P., Girshick R. et al. Focal Loss for Dense Object Detection // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020. Vol. 42, № 2. P. 318–327.
10. Ho J., Jain A., Abbeel P. Denoising Diffusion Probabilistic Models // Advances in Neural Information Processing Systems (NeurIPS). 2020. Vol. 33. P. 6840–6851.

# Структура левого нуль-пространства матрицы ограничений планарной пятииндексной транспортной задачи

*Е.Б. Титова, Р.Д. Махнёв*

*Нижегородский государственный университет им. Н.И. Лобачевского*

Для матрицы  $T$  ограничений планарной пятииндексной транспортной задачи рассмотрена структура и получена матрица  $L$  левого нуль-пространства, построен базис данного пространства, найден спектр собственных чисел для матрицы  $L \cdot L^T$ .

*Ключевые слова:* пятииндексная планарная транспортная задача, левое нуль-пространство, собственные числа.

## 1. Введение

Пусть  $\mathbb{Q}$  – поле рациональных чисел,  $\mathbb{Q}^n$  – линейное пространство столбцов с компонентами из  $\mathbb{Q}$ , а  $\mathbb{Z}^n$  – столбцы из  $\mathbb{Q}^n$  с целочисленными компонентами. Для  $(m \times n)$ - матрицы  $A$  с элементами из  $\mathbb{Q}$  обозначим через  $L_A = \{u \in \mathbb{Q}^m | u^T A = 0\}$  левое нуль-пространство матрицы  $A$ , а через  $L$  – какой-нибудь его базис. Можно считать, что  $L$  – целочисленная  $((m - r) \times m)$ -матрица, где  $r$  – ранг матрицы  $A$ .

Обозначим через  $M(A, b)$  множество рациональных решений системы  $Ax = b$ . Базис  $L$  левого нуль-пространства  $L_A$  позволяет описать множество всех векторов  $b \in \mathbb{Q}^m$ , при которых  $M(A, b) \neq \emptyset$ . По теореме Фредгольма [1] оно совпадает с множеством  $\{b \in \mathbb{Q}^m | Lb = 0\}$ .

Данная работа продолжает исследования, начатые в [1, 2, 3]. Для левого нуль-пространства  $L_A$  планарной пятииндексной транспортной задачи получен блочный вид, построен базис  $L$  и найден спектр собственных чисел для матрицы  $L_A \cdot L_A^T$ .

## 2. Постановка задачи

Рассмотрим ограничения-равенства планарной пятииндексной транспортной задачи (см., например, [4]) с неизвестными  $x_{j_1 j_2 j_3 j_4 j_5} \geq 0$  ( $j_i = 0, \dots, n_i$  при  $i = 1, 2, 3, 4, 5$ ):

$$\begin{aligned} \sum_{j_1=0}^{n_1} x_{j_1 j_2 j_3 j_4 j_5} &= b_{0 j_2 j_3 j_4 j_5}, \\ \sum_{j_2=0}^{n_2} x_{j_1 j_2 j_3 j_4 j_5} &= b_{j_1 0 j_3 j_4 j_5}, \\ \sum_{j_3=0}^{n_3} x_{j_1 j_2 j_3 j_4 j_5} &= b_{j_1 j_2 0 j_4 j_5}, \\ \sum_{j_4=0}^{n_4} x_{j_1 j_2 j_3 j_4 j_5} &= b_{j_1 j_2 j_3 0 j_5}, \\ \sum_{j_5=0}^{n_5} x_{j_1 j_2 j_3 j_4 j_5} &= b_{j_1 j_2 j_3 j_4 0} \end{aligned}$$

и обозначим через  $T(n_1, n_2, n_3, n_4, n_5)$  ее матрицу.

Из [5] следует, что матрицу  $T(n_1, n_2, n_3, n_4, n_5)$  можно представить в виде строчечных блоков:

$$T(n_1, n_2, n_3, n_4, n_5) = \begin{pmatrix} I_{n_1} \times E_{n_2} \times E_{n_3} \times E_{n_4} \times E_{n_5} \\ E_{n_1} \times I_{n_2} \times E_{n_3} \times E_{n_4} \times E_{n_5} \\ E_{n_1} \times E_{n_2} \times I_{n_3} \times E_{n_4} \times E_{n_5} \\ E_{n_1} \times E_{n_2} \times E_{n_3} \times I_{n_4} \times E_{n_5} \\ E_{n_1} \times E_{n_2} \times E_{n_3} \times E_{n_4} \times I_{n_5} \end{pmatrix},$$

где  $A \times B$  – кронекерово произведение матриц  $A$  и  $B$  (определение и свойства см., например, в [6]),  $E_n$  – единичная матрица  $n$ -го порядка,  $I_n = (1, \dots, 1)$  –  $n$ -мерная строка, все компоненты которой равны 1.

Обозначим через  $\sigma_i(n_1, n_2, n_3, n_4, n_5)$   $i$ -ю элементарную симметрическую функцию от  $n_1, \dots, n_5$ , т. е.  $\sigma_i(n_1, n_2, n_3, n_4, n_5) = \sum_{1 \leq k_1 < \dots < k_i \leq 5} n_{k_1} n_{k_2} \dots n_{k_i}$  для  $i \in \{1, 2, 3, 4, 5\}$ , и  $\sigma_0(n_1, n_2, n_3, n_4, n_5) = 1$ .

Матрица  $T(n_1, n_2, n_3, n_4, n_5)$  имеет число столбцов  $N_T = n_1 n_2 n_3 n_4 n_5$ , число строк  $M_T = \sigma_4(n_1, n_2, n_3, n_4, n_5)$  и ранг  $r_T = \sigma_4(n_1, n_2, n_3, n_4, n_5) - \sigma_3(n_1, n_2, n_3, n_4, n_5) + \sigma_2(n_1, n_2, n_3, n_4, n_5) - \sigma_1(n_1, n_2, n_3, n_4, n_5) + 1$ .

Матрица  $T(n_1, n_2, n_3, n_4, n_5)$  содержит базисную систему столбцов  $T^B (|B| = r)$ , имеющую минор порядка  $r$ , равный 1.

### 3. Основные результаты

Рассмотрим левое нуль-пространство матрицы  $T(n_1, n_2, n_3, n_4, n_5)$ . Обозначим через  $O_{m \times n}$  – нулевую  $(m \times n)$ -матрицу,  $I_1$  – число 1. В [1] дано описание построения левого нуль-пространства. Следуя ему, матрицу  $L(n_1, n_2, n_3, n_4, n_5)$  данного левого нуль-пространства можно представить рекурсивно в блочном виде:

$$L(n_1, n_2, n_3, n_4, n_5) = \begin{pmatrix} L(n_1, n_2, n_3, n_4) \times E_{n_5} & O_{\sigma_2(n_1, n_2, n_3, n_4) \cdot n_5 \times n_1 n_2 n_3 n_4} \\ -E_{\sigma_3(n_1, n_2, n_3, n_4)} \times I_{n_5} & T(n_1, n_2, n_3, n_4) \times I_1 \end{pmatrix},$$

где  $L(n_1, n_2, n_3, n_4)$  – матрица левого нуль-пространства планарной четырёхиндексной транспортной задачи с матрицей ограничений  $T(n_1, n_2, n_3, n_4)$  (подробное описание см. в [3]). Матрицу  $L(n_1, n_2, n_3, n_4, n_5)$  можно рассматривать как блочную с десятью строчечными блоками, тогда, например, первый блок будет иметь вид:

$$L_1 = (-I_1 \times I_{n_2} \times E_{n_3} \times E_{n_4} \times E_{n_5} | I_{n_1} \times I_1 \times E_{n_3} \times E_{n_4} \times E_{n_5} | 0_{n_3 n_4 n_5 \times n_1 n_2 n_4 n_5} | 0_{n_3 n_4 n_5 \times n_1 n_2 n_3 n_5} | 0_{n_3 n_4 n_5 \times n_1 n_2 n_3 n_4}).$$

Непосредственно перемножая матрицы, получаем, что

$$L(n_1, n_2, n_3, n_4, n_5) \cdot T(n_1, n_2, n_3, n_4, n_5) = 0.$$

Матрица  $L(n_1, n_2, n_3, n_4, n_5)$  имеет число строк  $M_L = \sigma_3(n_1, n_2, n_3, n_4, n_5)$ , число столбцов  $N_L = \sigma_4(n_1, n_2, n_3, n_4, n_5)$  и имеет размерность  $d_L(n_1, n_2, n_3, n_4, n_5) = M_L - r_T = \sigma_3(n_1, n_2, n_3, n_4, n_5) - \sigma_2(n_1, n_2, n_3, n_4, n_5) + \sigma_1(n_1, n_2, n_3, n_4, n_5) - 1$ .

Обозначим кратность  $k$  собственного числа  $\lambda$  через  $\lambda^{<k>}$ .

**Утверждение 1.** Собственные числа матрицы  $L(n_1, n_2, n_3, n_4, n_5) \cdot L^T(n_1, n_2, n_3, n_4, n_5)$  имеют следующий вид:

$$\begin{aligned} \lambda_1 &= n_1 + n_2 + n_3 + n_4 + n_5^{<4>}, \\ \lambda_2 &= n_1 + n_2 + n_3 + n_4^{<3(n_5-1)>}, \lambda_3 = n_1 + n_2 + n_3 + n_5^{<3(n_4-1)>}, \\ \lambda_4 &= n_1 + n_2 + n_4 + n_5^{<3(n_3-1)>}, \lambda_5 = n_1 + n_3 + n_4 + n_5^{<3(n_2-1)>}, \\ \lambda_6 &= n_2 + n_3 + n_4 + n_5^{<3(n_1-1)>}, \\ \lambda_7 &= n_1 + n_2 + n_3^{<2(n_4-1)(n_5-1)>}, \lambda_8 = n_1 + n_2 + n_4^{<2(n_3-1)(n_5-1)>}, \\ \lambda_9 &= n_1 + n_2 + n_5^{<2(n_3-1)(n_4-1)>}, \lambda_{10} = n_1 + n_3 + n_4^{<2(n_2-1)(n_5-1)>}, \\ \lambda_{11} &= n_1 + n_3 + n_5^{<2(n_2-1)(n_4-1)>}, \lambda_{12} = n_1 + n_4 + n_5^{<2(n_2-1)(n_3-1)>}, \\ \lambda_{13} &= n_2 + n_3 + n_4^{<2(n_1-1)(n_5-1)>}, \lambda_{14} = n_2 + n_3 + n_5^{<2(n_1-1)(n_4-1)>}, \\ \lambda_{15} &= n_2 + n_4 + n_5^{<2(n_1-1)(n_3-1)>}, \lambda_{16} = n_3 + n_4 + n_5^{<2(n_1-1)(n_2-1)>}, \\ \lambda_{17} &= n_1 + n_2^{<(n_3-1)(n_4-1)(n_5-1)>}, \lambda_{18} = n_1 + n_3^{<(n_2-1)(n_4-1)(n_5-1)>}, \\ \lambda_{19} &= n_2 + n_3^{<(n_1-1)(n_4-1)(n_5-1)>}, \lambda_{20} = n_1 + n_4^{<(n_2-1)(n_3-1)(n_5-1)>}, \end{aligned}$$

$$\begin{aligned}
\lambda_{21} &= n_2 + n_4 \langle (n_1-1)(n_3-1)(n_5-1) \rangle, \quad \lambda_{22} = n_3 + n_4 \langle (n_1-1)(n_2-1)(n_5-1) \rangle, \\
\lambda_{23} &= n_1 + n_5 \langle (n_2-1)(n_3-1)(n_4-1) \rangle, \quad \lambda_{24} = n_2 + n_5 \langle (n_1-1)(n_3-1)(n_4-1) \rangle, \\
\lambda_{25} &= n_3 + n_5 \langle (n_1-1)(n_2-1)(n_4-1) \rangle, \quad \lambda_{26} = n_4 + n_5 \langle (n_1-1)(n_2-1)(n_3-1) \rangle, \\
\lambda_{27} &= 0 \text{ кратности } \sigma_2(n_1, n_2, n_3, n_4, n_5) - \sigma_1(n_1, n_2, n_3, n_4, n_5) + 1.
\end{aligned}$$

Выделим в матрице  $T(n_1, n_2, n_3, n_4, n_5)$  базисную систему столбцов  $T^{\mathcal{B}}(|\mathcal{B}| = r)$ , а в матрице  $L(n_1, n_2, n_3, n_4, n_5)$  базисную систему строк  $L_{\bar{\mathcal{B}}}$ .

Из структуры задачи следует, что матрицу  $L_{\bar{\mathcal{B}}}$  можно представить в виде блочного столбца из 10 строк:

$$L_{\bar{\mathcal{B}}}(n_1, n_2, n_3, n_4, n_5) = (L_1 | L_2 | \bar{L}_3 | L_4 | \bar{L}_5 | \bar{L}_6 | L_7 | \bar{L}_8 | \bar{L}_9 | \bar{L}_{10})^T.$$

Блочные строки  $L_1, L_2, L_4, L_7$  переносятся из исходной матрицы без изменений. Количество строк в этих блоках составляет:  $m_1 = n_3 n_4 n_5, m_2 = n_2 n_4 n_5, m_4 = n_2 n_3 n_5, m_7 = n_2 n_3 n_4$ .

Каждый строчечный блок можно представить в следующем виде  $\bar{L}_i = (\bar{L}_{(i,1)} | \dots | \bar{L}_{(i,5)})$ ,  $i \in \{3, 5, 6, 8, 9, 10\}$ .

В блоках  $\bar{L}_3, \bar{L}_5, \bar{L}_8$  удаляем последнюю строку в матрице  $E_{n_1}$ , количество строк в нулевой матрице уменьшается соответственно:

$$\begin{aligned}
\bar{L}_{(3,1)} &= 0_{(n_1-1)n_4 n_5 \times n_2 n_3 n_4 n_5}, \quad \bar{L}_{(3,2)} = -E_{n_1-1 \times n_1} \times I_1 \times I_{n_3} \times E_{n_4} \times E_{n_5}, \\
\bar{L}_{(3,3)} &= E_{n_1-1 \times n_1} \times I_{n_2} \times I_1 \times E_{n_4} \times E_{n_5}, \quad \bar{L}_{(3,4)} = 0_{(n_1-1)n_4 n_5 \times n_1 n_2 n_3 n_5}, \\
\bar{L}_{(3,5)} &= 0_{(n_1-1)n_4 n_5 \times n_1 n_2 n_3 n_4}, \\
\bar{L}_{(5,1)} &= 0_{(n_1-1)n_3 n_5 \times n_2 n_3 n_4 n_5}, \quad \bar{L}_{(5,2)} = -E_{n_1-1 \times n_1} \times I_1 \times E_{n_3} \times I_{n_4} \times E_{n_5}, \\
\bar{L}_{(5,3)} &= 0_{(n_1-1)n_3 n_5 \times n_1 n_2 n_4 n_5}, \quad \bar{L}_{(5,4)} = E_{n_1-1 \times n_1} \times I_{n_2} \times E_{n_3} \times I_1 \times E_{n_5}, \\
\bar{L}_{(5,5)} &= 0_{(n_1-1)n_3 n_5 \times n_1 n_2 n_3 n_4}, \\
\bar{L}_{(8,1)} &= 0_{(n_1-1)n_3 n_4 \times n_2 n_3 n_4 n_5}, \\
\bar{L}_{(8,2)} &= -E_{n_1-1 \times n_1} \times I_1 \times E_{n_3} \times E_{n_4} \times I_{n_5}, \quad \bar{L}_{(8,3)} = 0_{(n_1-1)n_3 n_4 \times n_1 n_2 n_4 n_5}, \\
\bar{L}_{(8,4)} &= 0_{(n_1-1)n_3 n_4 \times n_1 n_2 n_3 n_5}, \quad \bar{L}_{(8,5)} = E_{n_1-1 \times n_1} \times I_{n_2} \times E_{n_3} \times E_{n_4} \times I_1.
\end{aligned}$$

Количество строк в этих блоках составляет:

$$m_3 = n_1 n_4 n_5 - n_4 n_5, m_5 = n_1 n_3 n_5 - n_3 n_5, m_8 = n_1 n_3 n_4 - n_3 n_4.$$

В блоках  $\bar{L}_6, \bar{L}_9$  удаляем последнюю строку в матрицах  $E_{n_1}$  и  $E_{n_2}$ :

$$\begin{aligned}
\bar{L}_{(6,1)} &= 0_{(n_1-1)(n_2-1)n_5 \times n_2 n_3 n_4 n_5}, \quad \bar{L}_{(6,2)} = 0_{(n_1-1)(n_2-1)n_5 \times n_1 n_3 n_4 n_5}, \\
\bar{L}_{(6,3)} &= -E_{n_1-1 \times n_1} \times E_{n_2-1 \times n_2} \times I_1 \times I_{n_4} \times E_{n_5}, \quad \bar{L}_{(6,4)} = E_{n_1-1 \times n_1} \times E_{n_2-1 \times n_2} \times I_{n_3} \times I_1 \times E_{n_5}, \\
\bar{L}_{(6,5)} &= 0_{(n_1-1)(n_2-1)n_5 \times n_1 n_2 n_3 n_4}, \\
\bar{L}_{(9,1)} &= 0_{(n_1-1)(n_2-1)n_4 \times n_2 n_3 n_4 n_5}, \quad \bar{L}_{(9,2)} = 0_{(n_1-1)(n_2-1)n_4 \times n_1 n_3 n_4 n_5}, \\
\bar{L}_{(9,3)} &= -E_{n_1-1 \times n_1} \times E_{n_2-1 \times n_2} \times I_1 \times E_{n_4} \times I_{n_5}, \quad \bar{L}_{(9,4)} = 0_{(n_1-1)(n_2-1)n_4 \times n_1 n_2 n_3 n_5}, \\
\bar{L}_{(9,5)} &= E_{n_1-1 \times n_1} \times E_{n_2-1 \times n_2} \times I_{n_3} \times E_{n_4} \times I_1.
\end{aligned}$$

Количество строк в этих блоках составляет:

$$m_6 = n_1 n_2 n_5 - n_1 n_5 - n_2 n_5 + n_5, m_9 = n_1 n_2 n_4 - n_1 n_4 - n_2 n_4 + n_4.$$

В блоке  $\bar{L}_{10}$  удаляем последнюю строку в матрицах  $E_{n_1}, E_{n_2}$  и  $E_{n_3}$ :

$$\begin{aligned}
\bar{L}_{(10,1)} &= 0_{(n_1-1)(n_2-1)(n_3-1) \times n_2 n_3 n_4 n_5}, \\
\bar{L}_{(10,2)} &= 0_{(n_1-1)(n_2-1)(n_3-1) \times n_1 n_3 n_4 n_5}, \quad \bar{L}_{(10,3)} = 0_{(n_1-1)(n_2-1)(n_3-1) \times n_1 n_2 n_4 n_5}, \\
\bar{L}_{(10,4)} &= -E_{n_1-1 \times n_1} \times E_{n_2-1 \times n_2} \times E_{n_3-1 \times n_3} \times I_1 \times I_{n_5}, \\
\bar{L}_{(10,5)} &= E_{n_1-1 \times n_1} \times E_{n_2-1 \times n_2} \times E_{n_3-1 \times n_3} \times I_{n_4} \times I_1.
\end{aligned}$$

Количество строк в этом блоке составляет:

$$m_{10} = n_1 n_2 n_3 - n_1 n_2 - n_1 n_3 - n_2 n_3 + n_1 + n_2 + n_3 - 1.$$

Таким образом ранг  $L_{\bar{\mathcal{B}}}(n_1, n_2, n_3, n_4, n_5)$  равен  $d_L(n_1, n_2, n_3, n_4, n_5)$ .

**Следствие 1.** *Определитель Грама матрицы  $L_{\bar{B}}$  равен*

$$\det(L_{\bar{B}} \cdot (L_{\bar{B}})^T) = \left(\sum_{i=1}^5 n_i\right)^4 \cdot \prod_{k=1}^5 \left(\sum_{i \neq k} n_i\right)^{3(n_k-1)} \cdot \prod_{1 \leq k < m \leq 5} \left(\sum_{i \neq k, m} n_i\right)^{2(n_k-1)(n_m-1)} \cdot \prod_{1 \leq k < m < l \leq 5} \left(\sum_{i \neq k, m, l} n_i\right)^{(n_k-1)(n_m-1)(n_l-1)}.$$

## Литература

1. Шевченко В.Н. Многогранники многоиндексных транспортных задач: алгебраический подход. // Материалы конференции «Дискретный анализ и исследование операций». Новосибирск: Изд-во ин-та математики, 2004, с. 64–70.
2. Титова Е.Б. Рост миноров рангового порядка матрицы ограничений трёхиндексной планарной транспортной задачи. // Труды XXIV Международной конференции «Математическое моделирование и суперкомпьютерные технологии», Нижний Новгород, 18–21 ноября 2024 года. – Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2024. – С. 164–167.
3. Титова Е.Б., Алпудов И.Н., Петров О.А., Махнёв Р.Д. Структура левого нуль-пространства матрицы ограничений четырёхиндексной планарной транспортной задачи. // Труды XXV Международной конференции «Математическое моделирование и суперкомпьютерные технологии», Нижний Новгород, 17–19 ноября 2025 года. – Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского, 2025. – С. 173–175.
4. Шевченко В.Н. Характеристические многочлены многоиндексных транспортных задач. // Дискретная математика, 2003. Т. 15, вып. 2. . 83–88.
5. Воеводин В.В., Кузнецов Ю.А. СМБ. Матрицы и вычисления. М.: Наука, 1984.
6. Мишина А.П., Проскуряков И.В. Высшая алгебра. Линейная алгебра, многочлены, общая алгебра. М.: Наука, 1965.

# Необходимые и достаточные условия подобия целочисленных матриц обобщённой жордановой клетке<sup>1</sup>

Г.В. Уткин

Нижегородский государственный университет им. Н.И. Лобачевского

Работа посвящена проблеме распознавания подобия над кольцом целых чисел верхних треугольных нильпотентных матриц и матриц с одной супердиагональю. В работе сформулированы необходимые условия для подобия таких матриц, а также критерий подобия для матриц 3, 4 и 5 порядков.

*Ключевые слова:* подобие матриц, нильпотентная матрица, обобщённая жорданова клетка, кольцо целых чисел.

## 1. Введение

Из линейной алгебры известен факт, что любая нильпотентная матрица порядка  $n$  и ранга  $n - 1$  подобна над полем  $\mathbb{Q}$  жордановой клетке  $J_n$ . Над кольцом целых чисел  $\mathbb{Z}$  это утверждение неверно. Назовём обобщённой жордановой клеткой матрицу вида:

$$\begin{pmatrix} 0 & a_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & a_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & a_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}.$$

Ясно, что жорданова клетка  $J_n$  является частным случаем обобщённой жордановой клетки при  $a_1 = a_2 = \dots = a_{n-1} = 1$ . Возникает естественный вопрос, подобна ли над  $\mathbb{Z}$  любая нильпотентная матрица  $A \in \mathbb{Z}^{n \times n}$  ранга  $n - 1$  некоторой обобщённой жордановой клетке. Ниже будет показано, что ответ на поставленный вопрос отрицательный и приведены некоторые условия подобия.

**Определение 1.** Матрица  $A \in \mathbb{Z}^{n \times n}$  подобна матрице  $B \in \mathbb{Z}^{n \times n}$  над кольцом целых чисел  $\mathbb{Z}$ , если существует такая унимодулярная матрица  $X \in GL(n, \mathbb{Z})$ , что  $AX = XB$ . При этом матрицу  $X$  называют трансформирующей. Факт подобия матриц  $A$  и  $B$  над кольцом целых чисел будем обозначать  $A \sim B$ .

## 2. Необходимые условия подобия

Рассмотрим верхние треугольные матрицы вида:

---

<sup>1</sup> Работа выполнена при финансовой поддержке научно-образовательного математического центра «Математика технологий будущего».

$$A = \begin{pmatrix} 0 & a_1 & b_1 & * & \dots & * & * \\ 0 & 0 & a_2 & b_2 & \dots & * & * \\ 0 & 0 & 0 & a_3 & \dots & * & * \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_{n-2} & b_{n-2} \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix}, B = \begin{pmatrix} 0 & a_1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & a_2 & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & a_3 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & a_{n-2} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & a_{n-1} \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 \end{pmatrix} \quad (1)$$

Следующая теорема является обобщением теоремы 3.3 из [1] для матриц вида (1).

**Теорема 1 (необходимые условия подобия).** Если матрицы  $A$  и  $B$  вида (1) подобны над кольцом целых чисел  $\mathbb{Z}$ , то выполняются следующие условия:

- 1)  $b_i$  делится на  $\text{НОД}(a_i, a_{i+1})$  для каждого  $i = 1, 2, \dots, n-2$ .
- 2)  $a_i b_{i+1} + b_i a_{i+2}$  делится на  $a_{i+1} \cdot \text{НОД}(a_i, a_{i+2})$  для каждого  $i = 1, 2, \dots, n-3$ .

### 3. Критерии подобия для матриц малых порядков ( $n = 3, 4, 5$ )

Следующая теорема из [1] описывает критерий подобия матриц 3-го порядка вида

$$A = \begin{pmatrix} 0 & a_1 & b_1 \\ 0 & 0 & a_2 \\ 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & a_1 & 0 \\ 0 & 0 & a_2 \\ 0 & 0 & 0 \end{pmatrix} \quad (2)$$

**Теорема 2.** Рассмотрим матрицы вида (2). Пусть  $a_1, a_2 \neq 0$ . Тогда  $A \sim B$  тогда и только тогда, когда  $b_1$  делится на  $\text{НОД}(a_1, a_2)$ .

Следующая теорема из [2] описывает критерий подобия матриц 4-го порядка вида

$$A = \begin{pmatrix} 0 & a_1 & b_1 & c_1 \\ 0 & 0 & a_2 & b_2 \\ 0 & 0 & 0 & a_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & a_1 & 0 & 0 \\ 0 & 0 & a_2 & 0 \\ 0 & 0 & 0 & a_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

**Теорема 3.** Рассмотрим матрицы вида (3). Пусть  $a_1, a_2, a_3 \neq 0$  и  $d = \text{НОД}(a_1, a_2, a_3)$ . Тогда  $A \sim B$  тогда и только тогда, когда выполняются следующие условия:

- 1)  $b_1, b_2, c_1$  делятся на  $d$ .
- 2)  $(a_1 b_2 + a_3 b_1)$  делится на  $d \cdot a_2$ .
- 3)  $(b_1 b_2 - a_2 c_1)$  делится на  $d \cdot \text{НОД}(a_1, a_3)$ .

Следующая теорема из [3] описывает критерий подобия матриц 5-го порядка вида

$$A = \begin{pmatrix} 0 & a_1 & b_1 & c_1 & d_1 \\ 0 & 0 & a_2 & b_2 & c_2 \\ 0 & 0 & 0 & a_3 & b_3 \\ 0 & 0 & 0 & 0 & a_4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & a_1 & 0 & 0 & 0 \\ 0 & 0 & a_2 & 0 & 0 \\ 0 & 0 & 0 & a_3 & 0 \\ 0 & 0 & 0 & 0 & a_4 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4)$$

**Теорема 4.** Рассмотрим матрицы вида (4). Пусть  $a_1, a_2, a_3, a_4 \neq 0$ ,  $\text{НОД}(a_1, a_4) = 1$  и  $\text{НОД}(a_1 a_2, a_2 a_3, a_3 a_4) = 1$ . Тогда  $A \sim B$  тогда и только тогда, когда выполняются следующие условия:

- 1)  $(a_1 b_2 + a_3 b_1)$  делится на  $a_2$ .
- 2)  $(a_2 b_3 + a_4 b_2)$  делится на  $a_3$ .

*Замечание.* В работе [3] в критерии было еще одно дополнительное условие, а именно  $(a_1a_2b_3 + a_1a_4b_2 + a_3a_4b_1)$  делится на  $a_2a_3$ , но это условие вытекает из первых двух условий. Действительно, из  $\text{НОД}(a_1a_2, a_2a_3, a_3a_4) = 1$  следует, что  $a_2$  и  $a_3$  взаимно простые. Тогда  $a_1a_2b_3 + a_1a_4b_2 + a_3a_4b_1 = a_1a_2b_3 + a_4(a_1b_2 + a_3b_1)$  делится на  $a_2$  по условию 1. С другой стороны,  $a_1a_2b_3 + a_1a_4b_2 + a_3a_4b_1 = a_1(a_2b_3 + a_4b_2) + a_3a_4b_1$  делится на  $a_3$  по условию 2. Стало быть, делится и на их произведение, т.к.  $a_2$  и  $a_3$  взаимно простые.

## **Благодарности**

Автор выражает глубокую признательность своему научному руководителю Сидорову Сергею Владимировичу за ценные советы, важные замечания и постоянное внимание к работе.

## **Литература**

1. Сидоров С.В., Уткин Г.В. О подобии над кольцом целых чисел некоторых нильпотентных матриц максимального ранга // Журнал Средневолжского математического общества. 2023. Т. 25, №4. С. 284-298.
2. Уткин Г.В. Критерий подобия над кольцом целых чисел нильпотентных верхних треугольных матриц 4-го порядка матрицам с одной супердиагональю // Математика и ИТ - вместе в цифровое будущее: Сборник трудов III Молодежной школы, Нижний Новгород, 22–26 апреля 2024 года. Нижний Новгород: Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского. – 71 с. 2024. С. 57-60.
3. Сидоров С.В., Уткин Г.В. О подобии над кольцом целых чисел верхних треугольных нильпотентных матриц 4-го и 5-го порядков обобщённой жордановой клетке // Журнал Средневолжского математического общества. 2025. Т. 27, № 1. С. 69–80.

# Конструирование метрик рабочего процесса: разработка и валидация интегрального индекса эффективности потока на основе метрик Lead Time и Cycle Time с учётом когнитивных факторов и организационной динамики

*А.В. Филиппов, В.Д. Дятлов*

*Ульяновский государственный университет*

В условиях цифровой трансформации и распространения гибких методологий управления проектами (Agile, Scrum, Kanban) актуальной научно-практической задачей становится разработка объективных количественных показателей для комплексной оценки эффективности рабочих процессов. В настоящей работе предлагается новый интегральный показатель — индекс эффективности потока (Flow Efficiency Index, FEI), основанный на соотношении суммарного активного времени выполнения задач (Cycle Time) к общему времени их жизненного цикла (Lead Time). Показатель позволяет выявлять скрытые простои, оценивать влияние организационных изменений, когнитивной нагрузки и факторов усталости на производительность команд, а также оптимизировать распределение ресурсов. В статье приводится детальное математическое обоснование модели с пошаговым объяснением механизма работы каждой формулы, рассматриваются методы корректировки метрик с учётом интенсивности труда, вовлечённости сотрудников, статуса задач и переменных внешних факторов, а также представлены результаты эмпирического анализа динамики индекса в период адаптации команд к новым рабочим практикам. Полученные результаты демонстрируют высокую диагностическую ценность предложенного подхода для мониторинга производительности, выявления узких мест в процессах и обоснования управленческих решений на основе данных.

*Ключевые слова:* Lead Time, Cycle Time, индекс эффективности потока, управление проектами, метрики производительности, Kanban, Agile, когнитивная нагрузка, регрессионный анализ, вовлечённость команды, оптимизация процессов, J-образная кривая адаптации, теория ограничений, бережливое производство.

## 1. Введение

Современные методологии разработки программного обеспечения и управления проектами, такие как Agile, Scrum и Kanban, уделяют особое внимание непрерывности, предсказуемости и прозрачности рабочих потоков. Ключевая цель этих подходов заключается в минимизации времени простоя задач, находящихся в очередях или ожидании внешних ресурсов, и максимизации доли времени, посвящённой непосредственной активной работе, создающей ценность для заказчика. В условиях высокой конкуренции и необходимости быстрой доставки продуктов на рынок, эффективность использования рабочего времени становится критическим фактором успеха организаций, определяющим их способность адаптироваться к изменяющимся рыночным условиям и технологическим вызовам.

На практике для мониторинга рабочих процессов широко используются две базовые метрики: Lead Time — общее время от момента создания задачи до её закрытия, и Cycle Time — время активной обработки задачи без учёта периодов ожидания. Эти показатели позволяют командам визуализировать поток работ, выявлять узкие места и планировать улучшения. Однако, несмотря на их распространённость, в литературе и практике отсутствует единый нормированный показатель, который бы объединял эти метрики в комплексную оценку, позволяющую оперативно сравнивать эффективность процессов во времени и между различными командами, а также учитывать контекстные факторы, такие как когнитивная нагрузка, усталость сотрудников и переменная интенсивность работы.

Актуальность разработки такого показателя обусловлена несколькими факторами. Во-первых, фрагментарный анализ отдельных метрик не позволяет получить целостное представление о качестве рабочего потока и эффективности использования человеческих ресурсов. Во-вторых, отсутствие универсального индекса затрудняет количественную оценку влияния организационных изменений, таких как внедрение новых инструментов, переход на удалённый формат работы, реструктуризация команд или изменение методологии управления. В-третьих, руководители проектов нуждаются в простом и интерпретируемом инструменте для быстрой диагностики состояния процессов и принятия обоснованных управленческих решений на основе данных, а не интуиции, что особенно важно в условиях неопределённости и высокой динамики современной бизнес-среды.

В связи с этим целью данной работы является разработка, математическое обоснование и эмпирическая валидация Индекса эффективности потока (Flow Efficiency Index, FEI), который позволяет проводить быструю диагностику рабочих процессов, оценивать влияние организационных изменений на производительность команд, учитывать когнитивные факторы и формировать рекомендации по оптимизации распределения ресурсов.

Для достижения поставленной цели решаются следующие задачи: (1) формализация математической модели индекса на основе метрик Lead Time и Cycle Time с подробным объяснением механизма работы формул; (2) разработка методов корректировки расчётов с учётом контекстных факторов (интенсивность труда, статус задач, вовлечённость); (3) создание системы интегральных показателей для оценки эффективности команды с учётом множителей ценности и качества; (4) проведение эмпирического анализа динамики показателя в реальных условиях адаптации команд; (5) формулировка практических рекомендаций по внедрению индекса в системы мониторинга проектов.

## 2. Обзор литературы и теоретические основы исследования

Теоретической основой исследования послужили работы в области теории ограничений (Theory of Constraints) Э. Голдратта, принципов бережливого производства (Lean) в разработке программного обеспечения и современных подходов к управлению потоками работ (Flow Management). Голдратт показал, что эффективность любой системы определяется её самым узким местом (ограничением), и предложил методы выявления и устранения таких ограничений. В контексте управления проектами это означает необходимость постоянного мониторинга времени ожидания задач и минимизации простоев, которые не создают ценности для конечного продукта.

Попендик адаптировал принципы Lean для IT-индустрии, выделив семь видов потерь, среди которых особое место занимает ожидание — время, когда задача находится в работе, но не обрабатывается активно. Рейнертсен развил эти идеи, предложив количественные метрики для оценки экономических последствий задержек и обосновав важность управления очередями задач. Эти работы заложили основу для использования метрик Lead Time и Cycle Time как индикаторов качества рабочего потока и эффективности использования ресурсов команды.

Петерсен и Волин провели систематический обзор методов измерения потока в бережливой разработке ПО и показали, что комбинация временных метрик с качественными оценками позволяет получать более точные и интерпретируемые результаты. Однако авторы отметили отсутствие единого стандарта для агрегации этих показателей, что ограничивает их сравнительный анализ между командами и проектами. Литтл сформулировал закон (Little's Law), связывающий среднее количество задач в системе, среднюю скорость их выполнения и среднее время пребывания задачи в системе, что стало математической основой для расчёта Cycle Time и Lead Time. Данное исследование направлено на заполнение выявленного пробела путём разработки нормированного индекса, объединяющего временные и контекстные факторы в единую комплексную оценку.

### 3. Математическая модель и методология исследования

#### 3.1. Базовая формула расчёта индекса эффективности потока

Пусть проект содержит  $n$  задач. Для каждой задачи  $i$  (где  $i$  изменяется от 1 до  $n$ ) определим две ключевые временные характеристики: (1)  $T(\text{active})$  — суммарное время активной работы над задачей (Cycle Time), то есть время, когда задача реально обрабатывается исполнителем; (2)  $T(\text{total})$  — полное время от создания до закрытия задачи (Lead Time), включающее как активную работу, так и все периоды ожидания (в очереди, на согласовании, на тестировании и т.д.). Тогда индекс эффективности потока вычисляется по следующей формуле:

$$FEI = \frac{\sum_{i=1}^n T_{\text{active}}^{(i)}}{\sum_{i=1}^n T_{\text{total}}^{(i)}}, \quad (1)$$

где  $\sum$  обозначает операцию суммирования по всем задачам проекта; значение индекса лежит в диапазоне  $[0; 1]$ .

Подробное объяснение работы формулы (1): числитель формулы представляет собой сумму всего активного времени по всем задачам проекта. Это время, когда сотрудники непосредственно работают над задачами — пишут код, проводят тестирование, анализируют требования. Знаменатель формулы — это сумма полного времени жизненного цикла всех задач, которое включает не только активную работу, но и все простои: ожидание в очереди на начало работы, ожидание ответа от клиента, время на согласование, тестирование, исправление замечаний. Деление числителя на знаменатель даёт нам долю (процент) времени, которое было потрачено на полезную работу, от общего времени выполнения проекта. Если  $FEI = 0.8$ , это означает, что 80% времени задачи находились в активной работе и только 20% времени простаивали.

Интерпретация значений индекса имеет следующий вид: (1)  $FEI \rightarrow 1$  (значения 0.8–1.0) свидетельствует о высокой эффективности: задачи находятся в постоянной обработке, простои минимизированы, ресурсы используются оптимально, процессы отлажены; (2)  $FEI 0$  (значения 0.0–0.3) указывает на наличие системных задержек, бюрократических барьеров, неравномерного распределения нагрузки, неэффективных процессов координации или постоянных внешних блокировок; (3) значения в диапазоне 0.4–0.7 характерны для зрелых команд с отлаженными процессами, где часть времени неизбежно тратится на ожидание внешних зависимостей, согласований и тестирования. Оптимальным значением для большинства проектов считается диапазон 0.6–0.8.

Таблица 1. Интерпретация значений индекса эффективности потока

Диапазон FEI	Уровень эффективности	Характеристика состояния процесса
0.8 – 1.0	Отличный	Минимальные простои, процессы оптимизированы
0.6 – 0.8	Хороший	Небольшие простои, характерные для зрелых команд
0.4 – 0.6	Удовлетворительный	Заметные простои, требуются улучшения
0.2 – 0.4	Неудовлетворительный	Критические простои, необходима оптимизация
0.0 – 0.2	Критический	Системные проблемы, процессы неэффективны

#### 3.2. Статистическая валидация модели и проверка значимости

Для проверки статистической значимости предложенного индекса и установления связи между FEI и реальными показателями производительности команды используется коэффициент корреляции Пирсона  $r$ . Этот коэффициент показывает силу и направление линейной связи между

двумя переменными: индексом эффективности потока и метриками результативности (количество завершённых задач, качество кода, удовлетворённость клиентов). Нулевая гипотеза  $H_0: r = 0$  (корреляция отсутствует) отвергается при уровне значимости  $\alpha = 0.05$ , если рассчитанное значение  $t$ -критерия Стьюдента превышает критическое значение  $t_{crit}$  для данного числа степеней свободы  $df = n - 2$ , где  $n$  — объём выборки.

$$t = r \cdot \sqrt{\frac{n-2}{1-r^2}}, \quad (2)$$

где  $n$  — объём выборки (количество наблюдений);  $r$  — коэффициент корреляции Пирсона.

Объяснение работы формулы (2): данная формула преобразует коэффициент корреляции  $r$  в  $t$ -статистику, которую можно сравнить с критическими значениями из таблицы распределения Стьюдента. Числитель  $r \cdot \sqrt{n-2}$  показывает, насколько сильна корреляция с учётом объёма данных — чем больше выборка и чем выше корреляция, тем больше  $t$ -значение. Знаменатель  $\sqrt{1-r^2}$  учитывает долю необъяснённой дисперсии — если корреляция слабая ( $r$  близко к 0), знаменатель близок к 1, и  $t$ -значение будет небольшим. Если же корреляция сильная ( $r$  близко к 1 или -1), знаменатель стремится к 0, что приводит к большому  $t$ -значению. Полученное  $t$ -значение сравнивается с критическим значением из статистической таблицы: если  $t > t_{crit}$ , то корреляция статистически значима, и мы можем утверждать, что связь между FEI и производительностью не случайна.

**Таблица 2.** Критические значения  $t$ -критерия Стьюдента для разных уровней значимости

$df$ (степени свободы)	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.001$
5	2.571	4.032	6.869
10	2.228	3.169	4.587
20	2.086	2.845	3.850
30	2.042	2.750	3.646
50	2.009	2.678	3.496

### 3.3. Корректировка с учётом статусов задач и интенсивности труда

Прямое применение формулы (1) может приводить к искажённым результатам, если не учитывать контекст выполнения задач и их статусы. Во-первых, различные статусы задач («в работе», «на проверке», «ожидание клиента», «на доработке») имеют разную продуктивную нагрузку: время в статусе «на проверке» может быть как активным (если проверка требует усилий тестировщика), так и пассивным (если задача просто ожидает очереди на проверку). Во-вторых, интенсивность работы может варьироваться в зависимости от срочности, сложности задачи или уровня вовлечённости исполнителя — один и тот же час работы может быть потрачен с разной степенью концентрации и продуктивности.

Для повышения точности модели вводится коэффициент интенсивности  $k_{int} \in [0.5; 1.2]$ , который корректирует учёт активного времени в зависимости от контекста: значение 1.0 соответствует стандартной обработке задачи со средней интенсивностью; значения  $>1.0$  (до 1.2) отражают повышенную концентрацию, высокую срочность, критическую важность или сложность задачи; значения  $<1.0$  (до 0.5) — фоновую или частичную вовлечённость, многозадачность, работу в режиме ожидания. Скорректированная формула принимает следующий вид:

$$FEI_{adj} = \frac{\sum_{i=1}^n k_{int}^{(i)} \cdot T_{active}^{(i)}}{\sum_{i=1}^n T_{total}^{(i)}}, \quad (3)$$

где  $k_{int}^{(i)}$  — коэффициент интенсивности для  $i$ -й задачи, определяемый экспертно или на основе данных тайм-трекинга.

Объяснение работы формулы (3): в числителе формулы каждое значение активного времени  $T_{active}^{(i)}$  умножается на соответствующий коэффициент интенсивности  $k_{int}^{(i)}$ . Это означает, что если задача выполнялась с повышенной интенсивностью ( $k_{int} = 1.2$ ), то её вклад в общую эффективность увеличивается на 20%. Напротив, если задача выполнялась в фоновом режиме ( $k_{int} = 0.6$ ), её вклад уменьшается на 40%. Таким образом, формула учитывает не только факт работы над задачей, но и качество этой работы с точки зрения вовлечённости и концентрации исполнителя. Знаменатель остаётся неизменным — это общее время выполнения всех задач. В результате  $FEI_{adj}$  показывает эффективность с учётом того, насколько интенсивно использовалось рабочее время.

**Таблица 3.** Коэффициенты интенсивности для различных статусов задач

Статус задачи	Коэффициент $k_{int}$	Обоснование
Активная разработка (фокус)	1.0 – 1.2	Высокая концентрация, полная вовлечённость
Код-ревью / Тестирование	0.9 – 1.0	Активная работа, но с переключениями
Ожидание ответа (частичная занятость)	0.5 – 0.7	Низкая интенсивность, многозадачность
На согласовании / В очереди	0.0	Полный простой, не включается в Active Time

#### 4. Эмпирический анализ адаптации рабочих процессов

Для валидации предложенной модели был проведён эмпирический анализ динамики индекса эффективности в период внедрения новых практик в команде разработки ПО (N=12 сотрудников, длительность наблюдения — 8 недель). Данные собирались еженедельно из системы управления задачами Jira и дополнялись результатами субъективных опросов по шкале вовлечённости и усталости. Для каждого сотрудника фиксировались: общее время работы, активное время по задачам, количество переключений контекста, самооценка усталости (1–5 баллов), оценка вовлечённости (1–5 баллов). На основе этих данных рассчитывался еженедельный индекс FEI для команды в целом.

**Таблица 4.** Динамика индекса эффективности в период адаптации (8 недель)

Неделя	FEI	Изменение	Фаза	Примечание
0	0.80	-	Базовая	Исходное состояние
1	0.75	-0.05	Адаптация	Начало внедрения изменений
2	0.60	-0.15	Кризис	Когнитивная перегрузка
3	0.65	+0.05	Стабилизация	Начало адаптации
4	0.75	+0.10	Восстановление	Формирование паттернов
5	0.80	+0.05	Рост	Превышение базового уровня
6	0.85	+0.05	Рост	Оптимизация процессов
7	0.87	+0.02	Стабилизация	Плато эффективности
8	0.88	+0.01	Стабилизация	Устойчивый результат

Анализ полученных данных демонстрирует классическую J-образную кривую адаптации. На начальной фазе (недели 0–2) наблюдается статистически значимое снижение показателя с 0.80 до 0.60 ( $p < 0.01$ ), что обусловлено когнитивной перегрузкой при освоении новых инструментов, изменением рабочих паттернов и временным ростом времени на координацию. Сотрудники тратят дополнительное время на обучение, адаптацию к новым процессам и преодоление сопротивления изменениям. В фазе стабилизации (недели 3–4) фиксируется восстановление до 0.75 по мере формирования устойчивых практик и снижения транзакционных

издержек. Начиная с 5-й недели индекс превышает исходный уровень, достигая 0.88, что подтверждает успешное встраивание изменений в операционную деятельность и получение долгосрочных преимуществ от оптимизации процессов.

**Таблица 5.** Статистические характеристики выборки (N=12 сотрудников)

Показатель	Среднее значение	Стандартное отклонение	95% доверительный интервал
FEI (неделя 0)	0.80	0.05	[0.77; 0.83]
FEI (неделя 2)	0.60	0.08	[0.55; 0.65]
FEI (неделя 8)	0.88	0.04	[0.85; 0.91]
Коэффициент корреляции (r)	0.73	0.12	[0.65; 0.81]
t-критерий	4.87	-	p < 0.001

## 5. Индекс вовлечённости и учёт когнитивных факторов

### 5.1. Расчёт индекса вовлечённости

Для комплексной оценки состояния сотрудника и учёта человеческого фактора в модели вводится индекс вовлечённости  $E$ , рассчитываемый как средневзвешенное субъективной оценки (полученной через опросы) и поведенческого индекса (рассчитанного на основе объективных данных систем трекинга). Такой смешанный подход (Mixed-Methods) позволяет компенсировать недостатки каждого метода в отдельности: субъективные оценки могут быть искажены настроением сотрудника, а объективные данные не всегда отражают внутреннее состояние и мотивацию.

$$E = \alpha \cdot S_{subj} + \beta \cdot B_{behav} \quad \alpha + \beta = 1, \quad (4)$$

где  $S_{subj}$  — балл субъективного опроса (шкала 1–5, нормализованный к [0;1]);  $B_{behav}$  — нормализованный поведенческий индекс (на основе данных тайм-трекинга);  $\alpha, \beta$  — калибровочные веса (рекомендуется  $\alpha = 0.4, \beta = 0.6$ ).

Объяснение работы формулы (4): формула представляет собой взвешенную сумму двух компонентов. Субъективная оценка  $S_{subj}$  получается из еженедельных опросов, где сотрудник оценивает свой уровень энергии, интерес к задачам и чувство вовлечённости по шкале от 1 до 5. Эта оценка затем нормализуется к диапазону [0; 1] делением на 5. Поведенческий индекс  $B_{behav}$  рассчитывается на основе объективных данных: время работы в основных приложениях (Jira, IDE, Git), частота коммитов, активность в коммуникационных каналах, время отклика на запросы. Этот индекс также нормализуется к [0; 1]. Коэффициенты  $\alpha$  и  $\beta$  определяют, какой вес придаётся каждому компоненту. Рекомендуется  $\alpha = 0.4$  и  $\beta = 0.6$ , так как объективные данные обычно более надёжны, но полностью игнорировать субъективное состояние нельзя. Сумма коэффициентов равна 1, чтобы итоговый индекс  $E$  также находился в диапазоне [0; 1].

### 5.2. Учёт фактора усталости и когнитивного истощения

Длительная интенсивная работа без адекватного отдыха приводит к когнитивному истощению, которое снижает продуктивность и качество работы. Для учёта этого фактора вводится ограничивающий фактор  $F_{lim}$ , который снижает расчётную эффективность при накоплении признаков усталости. Фактор рассчитывается на основе комбинации самооценки усталости и поведенческих индикаторов истощения (снижение активности, увеличение количества ошибок, замедление темпа работы).

$$F_{lim} = 1 - \frac{E_{fat} + B_{exh}}{2}, \quad (5)$$

где  $E_{fat}$  — самооценка усталости (шкала 0–1, где 0 — полный отдых, 1 — полное истощение);  $B_{exh}$  — индекс поведенческого истощения (на основе данных трекинга). При  $F_{lim} < 0.6$  рекомендуется пересмотр нагрузки или предоставление ресурса на восстановление.

Объяснение работы формулы (5): формула рассчитывает ограничивающий фактор как обратную величину от средней усталости.  $E_{fat}$   $B_{exh}$  — это два показателя усталости, каждый из которых находится в диапазоне [0; 1], где 0 означает отсутствие усталости, а 1 — полное истощение. Их среднее значение  $(E_{fat} + B_{exh})/2$  показывает общий уровень усталости сотрудника. Вычитая это среднее из 1, мы получаем фактор  $F_{lim}$ , который также находится в диапазоне [0; 1], но интерпретируется наоборот: 1 означает полную готовность к работе, 0 — полное истощение. Если  $F_{lim} < 0.6$ , это сигнал о том, что сотрудник накопил критический уровень усталости и нуждается в отдыхе. В этом случае рекомендуется снизить нагрузку, предоставить дополнительный выходной или перераспределить задачи.

**Таблица 6.** Интерпретация значений фактора усталости  $F_{lim}$

Диапазон $F_{lim}$	Уровень усталости	Рекомендации
0.8 – 1.0	Низкий	Оптимальное состояние, можно увеличивать нагрузку
0.6 – 0.8	Умеренный	Нормальное рабочее состояние, мониторинг
0.4 – 0.6	Повышенный	Признаки переутомления, снизить нагрузку на 20%
0.2 – 0.4	Высокий	Критическая усталость, предоставить отдых
0.0 – 0.2	Критический	Истощение, срочный отпуск/выходные

## 6. Интегральная оценка эффективности команды

### 6.1. Множитель ценности сотрудника

Для отражения разницы в продуктивности специалистов разного уровня и опыта вводится множитель ценности  $R$ , который учитывает как роль сотрудника (Junior, Middle, Senior, Lead), так и его стаж работы. Логарифмическая функция используется для нормализации вклада стажа, чтобы избежать завышения коэффициента при большом опыте и отразить закон убывающей отдачи: каждый дополнительный год опыта даёт всё меньший прирост продуктивности.

$$R = w_{role} \cdot \ln(1 + \tau_{exp}), \quad (6)$$

где  $w_{role}$  — экспертный вес роли (Junior=0.8, Middle=1.0, Senior=1.2, Lead=1.4);  $\tau_{exp}$  — стаж в годах;  $\ln$  — натуральный логарифм.

Объяснение работы формулы (6): формула состоит из двух множителей.

Первый множитель  $w_{role}$  — это экспертная оценка вклада роли в общий результат. Junior-разработчик получает коэффициент 0.8, так как требует больше контроля и времени на обучение; Middle — 1.0 (базовый уровень); Senior — 1.2 за способность решать сложные задачи и менторить других; Lead — 1.4 за лидерство и архитектурные решения.

Второй множитель  $\ln(1 + \tau_{exp})$  учитывает стаж. Используется натуральный логарифм, чтобы рост коэффициента замедлялся с увеличением стажа. Например, при  $\tau_{exp} = 1$  год:  $\ln(2) \approx 0.69$ ; при  $\tau_{exp} = 5$  лет:  $\ln(6) \approx 1.79$ ; при  $\tau_{exp} = 10$  лет:  $\ln(11) \approx 2.40$ . Видно, что рост с 1 до 5 лет даёт прирост 1.10, а с 5 до 10 лет — только 0.61, что отражает убывающую отдачу от опыта.

Перемножение этих множителей даёт итоговый коэффициент ценности  $R$ , который показывает, насколько работа данного сотрудника «весомее» работы базового Middle-разработчика без опыта.

**Таблица 7.** Примеры расчёта множителя ценности  $R$  для разных ролей и стажа

Роль	$w_{role}$	Стаж (лет)	$\ln(1 + \tau)$	$R$	Интерпретация
Junior	0.8	1	0.69	0.55	Младший специалист
Middle	1.0	3	1.39	1.39	Базовый уровень
Middle	1.0	5	1.79	1.79	Опытный разработчик
Senior	1.2	7	2.08	2.50	Старший специалист
Lead	1.4	10	2.40	3.36	Технический лидер

## 6.2. Формула интегральной эффективности команды

Итоговая оценка эффективности команды  $\eta$  учитывает не только затраченное время, но и ценность выполненной работы (через множители  $R$ ), уровень вовлечённости сотрудников ( $A$ ), частоту переключений контекста ( $C$ ) и качество результата ( $Q$ ). Такая комплексная формула позволяет получить реалистичную оценку, которая штрафует за многозадачность и поощряет высокое качество работы.

$$\eta = \frac{\sum_{j=1}^N (T_j \cdot V_j \cdot A_j)}{\sum_{j=1}^N T_{total,j} + \varepsilon} \cdot (1 - \theta C + \mu Q) \cdot 100\%, \quad (7)$$

где  $T_j$  — продуктивное время  $j$ -го сотрудника;  $V_j$  — множитель ценности задачи;  $A_j$  — нормализованная вовлечённость;  $\varepsilon = 0.1$  — техническая константа;  $C$  — доля переключений контекста  $[0; 1]$ ;  $\theta \in [0.3; 0.5]$  — штрафной коэффициент;  $Q \in [0; 1]$  — метрика качества;  $\mu \in [0.2; 0.4]$  — бонусный коэффициент.

Объяснение работы формулы (7): формула состоит из трёх основных частей.

Первая часть  $\frac{\sum(T_j \cdot V_j \cdot A_j)}{\sum T_{total,j} + \varepsilon}$  — это базовая эффективность, взвешенная по ценности и вовлечённости. В числителе суммируется произведение трёх величин для каждого сотрудника: продуктивного времени  $T_j$ , множителя ценности  $V_j$  (насколько важна работа) и вовлечённости  $A_j$ . Это даёт суммарную «ценную продуктивную работу». В знаменателе — общее рабочее время всех сотрудников плюс малая константа  $\varepsilon = 0.1$  для избежания деления на ноль.

Вторая часть  $(1 - \theta C)$  — это штраф за переключения контекста.  $C$  — это доля времени, потраченная на переключения (от 0 до 1).  $\theta$  — коэффициент штрафа (рекомендуется 0.3–0.5). Если  $C=0.3$  (30% времени на переключения) и  $\theta = 0.4$ , то штраф составит  $0.4 \cdot 0.3 = 0.12$ , и множитель будет равен  $1 - 0.12 = 0.88$ , то есть эффективность снизится на 12%.

Третья часть  $(+ \mu Q)$  — это бонус за качество.  $Q$  — метрика качества от 0 до 1 (например, процент задач, принятых с первого раза).  $\mu$  — коэффициент бонуса (0.2–0.4). Если  $Q = 0.9$   $\mu = 0.3$ , то бонус составит  $0.3 \cdot 0.9 = 0.27$ .

Итоговый множитель  $(1 - \theta C + \mu Q)$  может быть больше 1, если бонус за качество превышает штраф за переключения. Умножение на 100% переводит результат в проценты для удобства интерпретации.

**Таблица 8.** Параметры формулы интегральной эффективности и источники данных

Параметр	Диапазон	Источник данных	Метод получения
$T_j$ (продуктивное время)	часы	Toggl, Harvest, Time Doctor	Автоматический трекинг
$V_j$ (множитель ценности)	0.5–2.0	Jira, Asana, эксперт	Story points, приоритет
$A_j$ (вовлечённость)	0–1	Опросы, HR-система	Нормализация оценок 1–5
$C$ (переключения)	0–1	Системы трекинга	Подсчёт смен задач
$Q$ (качество)	0–1	QA-метрики	Процент принятых задач
$\theta$ (штраф)	0.3–0.5	Настройка	Калибровка под команду
$\mu$ (бонус)	0.2–0.4	Настройка	Калибровка под команду

**Таблица 9.** Результаты регрессионного анализа влияния факторов на FEI

Фактор	$\beta$ -коэффициент	t-значение	$p$ – value	VIF
Сложность задач	0.34	2.87	0.008	1.23
Физическая активность	0.28	2.41	0.021	1.15
Непрерывное обучение	0.41	3.52	0.001	1.31
Коэффициент вовлечённости	0.52	4.18	<0.001	1.08
Опыт работы (стаж)	0.19	1.87	0.067	1.42

Результаты регрессионного анализа (Таблица 9) показывают, что все факторы, кроме стажа ( $p=0.067$ ), оказывают статистически значимое влияние на индекс эффективности ( $p < 0.05$ ). Наибольшее влияние имеет коэффициент вовлечённости ( $\beta = 0.52, p < 0.001$ ), что подчёркивает важность мотивации и психологического состояния сотрудников. Непрерывное обучение ( $\beta = 0.41, p = 0.001$ ) и сложность задач ( $\beta = 0.34, p = 0.008$ ) также положительно коррелируют с эффективностью, подтверждая гипотезу о том, что интеллектуально насыщенные задачи стимулируют продуктивность. Физическая активность ( $\beta = 0.28, p = 0.021$ ) демонстрирует умеренное, но значимое влияние, что согласуется с данными о связи физической формы и когнитивных функций. VIF (Variance Inflation Factor) для всех факторов меньше 2, что указывает на отсутствие мультиколлинеарности и независимость факторов друг от друга.

## 7. Заключение

Предложенная модель индекса эффективности потока представляет собой методологически обоснованный инструмент комплексной оценки рабочих процессов, сочетающий математическую строгость с практической применимостью. В ходе исследования были решены все поставленные задачи: (1) формализована математическая модель индекса FEI на основе метрик Lead Time и Cycle Time с подробным объяснением механизма работы каждой формулы; (2) разработаны методы корректировки расчётов с учётом интенсивности труда, статусов задач, вовлечённости и фактора усталости; (3) создана система интегральных показателей для оценки эффективности команды с учётом множителей ценности, качества и штрафов за переключения контекста; (4) проведён эмпирический анализ, подтвердивший высокую чувствительность

индекса к изменениям рабочих практик; (5) сформулированы практические рекомендации по внедрению индекса в системы мониторинга проектов.

Статистический анализ подтвердил высокую корреляцию  $r = 0.73$ ,  $p < 0.001$ ) между FEI и показателями производительности команды, что свидетельствует о валидности предложенного подхода. Регрессионный анализ выявил значимое влияние факторов непрерывного обучения ( $\beta = 0.41$ ,  $p = 0.001$ ), вовлечённости ( $\beta = 0.52$ ,  $p < 0.001$ ) и сложности задач ( $\beta = 0.34$ ,  $p = 0.008$ ) на эффективность рабочего потока. Эмпирические данные продемонстрировали классическую J-образную кривую адаптации с падением эффективности на 25% в фазе кризиса (недели 0–2) и последующим ростом на 10% выше базового уровня к 8-й неделе, что подтверждает долгосрочную пользу от внедрения оптимизаций, несмотря на краткосрочные издержки адаптации.

Практическая ценность исследования заключается в следующем: (1) предложен простой и интерпретируемый инструмент для быстрой диагностики эффективности процессов; (2) разработана методика учёта контекстных факторов (усталость, вовлечённость, интенсивность), что повышает точность оценок; (3) создана система раннего предупреждения о когнитивном истощении сотрудников на основе фактора  $F_{lim}$ ; (4) сформулированы рекомендации по калибровке коэффициентов под специфику конкретной команды. Модель создаёт эмпирическую базу для мониторинга адаптации команд, оптимизации распределения нагрузки и обоснования организационных изменений на основе данных, а не интуиции.

Направления дальнейших исследований включают: (1) разработку методов машинного обучения для автоматической калибровки коэффициентов модели на основе исторических данных и выявления скрытых паттернов; (2) расширение модели за счёт учёта межкомандных зависимостей и внешних факторов (рыночные изменения, регуляторные требования); (3) создание веб-интерфейса для визуализации динамики индекса в реальном времени и формирования персонализированных рекомендаций по оптимизации процессов; (4) проведение кросс-индустриальных исследований для валидации модели на выборках из различных отраслей (финтех, здравоохранение, образование); (5) интеграция с системами искусственного интеллекта для прогнозирования падения эффективности и превентивного вмешательства.

## Литература

1. Goldratt E.M. Theory of Constraints. Great Barrington: The North River Press, 2000. 172 p.
2. Anderson D.J. Kanban: Successful Evolutionary Change for Your Technology Business. Seattle: Blue Hole Press, 2010. 264 p.
3. Poppendieck M., Poppendieck T. Lean Software Development: An Agile Toolkit. Boston: Addison-Wesley, 2003. 256 p.
4. Schwaber K., Sutherland J. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. 2020. 19 p.
5. Petersen K., Wohlin C. Measuring the flow in lean software development // Software: Practice and Experience. 2011. Vol. 41, No. 9. P. 975–996.
6. Wohlin C., Petersen K. Measuring the Flow in Lean Software Development // Communications of the ACM. 2011. URL: <https://www.wohlin.eu/spe11.pdf> (дата обращения: 20.05.2024).
7. Goldratt E.M. Critical Chain. Great Barrington: The North River Press, 1997. 448 p.
8. Reinertsen D.G. The Principles of Product Development Flow: Second Generation Lean Product Development. Redondo Beach: Celeritas Publishing, 2009. 248 p.
9. Kniberg H., Skarin M. Kanban and Scrum: Making the Most of Both. Seattle: InfoQ, 2010. 108 p.
10. Little T. Little's Law // The Science of Lean Software and DevOps. 2018.

# Численное моделирование движения тела при учёте сопротивления воздуха

*Р.С. Якунин*

*Казанский национальный исследовательский технологический университет*

В работе реализовано численное решение системы нелинейных обыкновенных дифференциальных уравнений движения снаряда с учётом квадратичного сопротивления воздуха. Разработаны собственные реализации методов Эйлера и Рунге–Кутты 4-го порядка. Проведено моделирование для шести объектов с реальными параметрами при трёх углах броска. Получены таблицы дальности, максимальной высоты и времени полёта, а также точность результатов. Решение может быть использовано в задачах компьютерной механики, спортивной инженерии и образовательных симуляторах.

*Ключевые слова:* аэродинамика, дифференциальные уравнения, вычислительная механика.

## Введение

Математическое моделирование движения снарядов и тел, брошенных под углом к горизонту, является классической задачей механики и находит применение в вычислительной механике, робототехнике и др. областях. Траектория тела в вакууме описывается простыми аналитическими формулами, но в реальных условиях сопротивление воздуха оказывает значительное влияние на параметры траектории. При умеренных скоростях сопротивление воздуха пропорционально квадрату скорости, из-за чего для его вычисления требуется система нелинейных обыкновенных дифференциальных уравнений, не имеющая точного аналитического решения в общем виде [1, 2].

В связи с этим для исследования траекторий широко используются численные методы интегрирования; наиболее распространёнными являются метод Эйлера первого порядка и метод Рунге-Кутты четвёртого порядка (RK4), обладающий значительно более высокой точностью [3].

Цель настоящей работы состоит в численном моделировании траекторий движения тел с учётом квадратичного сопротивления воздуха. В рамках исследования реализованы собственные численные схемы на основе метода Эйлера и метода Рунге–Кутты 4-го порядка, проведена их верификация по аналитическому решению для вакуума и сопоставление с адаптивным численным решением, полученным с помощью встроенного метода RK45 библиотеки SciPy языка программирования Python [4]. Выполнено параметрическое исследование для нескольких объектов с реалистичными физическими характеристиками, включая гольф-мяч, теннисный мяч, бейсбольный мяч, волан и пушечное ядро, при трёх углах броска: 30°, 45° и 60°. Особое внимание уделено анализу влияния сопротивления воздуха на дальность полёта, максимальную высоту, время движения и смещение оптимального угла броска по сравнению с вакуумным случаем.

## 1. Математическая модель задачи

### 1.1. Физическая модель и система уравнений

Для упрощения построения схем в физической модели были приняты следующие условия: движение рассматривается в вертикальной и горизонтальной плоскости, тело считается материальной точкой с фиксированным коэффициентом сопротивления формы, а при расчёте траектории учитываются только сила тяжести и сопротивление воздуха. Вращение тела, ветер, изменение плотности воздуха с высотой в физической модели не учитываются.

Пусть  $x(t)$  и  $y(t)$  обозначают координаты тела, а  $v_x(t)$  и  $v_y(t)$  – его проекции скорости на оси  $Ox$  и  $Oy$ . Тогда модуль скорости  $|v| = \sqrt{v_x^2 + v_y^2}$ , а сила сопротивления воздуха задаётся выражением:

$$F_d = -\frac{1}{2}C_d\rho A|v|v,$$

где  $C_d$  – это коэффициент сопротивления формы,  $\rho$  – плотность воздуха, а  $A = \pi r^2$  – площадь поперечного сечения тела.

С учётом силы тяжести  $F_g = (0, -mg)$  система уравнений движения приобретает вид

$$\begin{aligned} \frac{dx}{dt} &= v_x, \quad \frac{dy}{dt} = v_y, \\ \frac{dv_x}{dt} &= -\frac{1}{2m}C_d\rho A|v|v_x, \quad \frac{dv_y}{dt} = -g - \frac{1}{2m}C_d\rho A|v|v_y. \end{aligned}$$

Начальные условия модели задаются в форме

$$x(0) = 0, y(0) = 0, v_x(0) = v_0 \cos\theta, v_y(0) = v_0 \sin\theta.$$

При отсутствии сопротивления воздуха, то есть при  $C_d = 0$ , система переходит в классическую модель движения тела, брошенного под углом к горизонту.

Таким образом, движение тела описывается системой нелинейных обыкновенных дифференциальных уравнений первого порядка. Нелинейность системы обусловлена тем, что сила аэродинамического сопротивления зависит от модуля скорости и направлена противоположно вектору движения.

В отличие от «идеального» случая, где коэффициент сопротивления формы не учитывается, при необходимости расчёта квадратичного сопротивления воздуха получить аналитический ответ в общем случае затруднительно [1, 2]. По этой причине для исследования движения используются численные методы интегрирования, позволяющие вычислять дальность полёта, максимальную высоту и время полёта для различных параметров тела и начальных условий [3].

## 1.2. Параметры объектов и условия моделирования

Для численного исследования были выбраны несколько объектов, различающихся по массе, размеру и аэродинамическим характеристикам (представленным в данном исследовании коэффициентом сопротивления формы). Такая выборка позволяет отследить, как физические параметры тела влияют на траекторию движения при одинаковых начальных условиях. В качестве объектов рассматривались мяч для гольфа, теннисный мяч, бейсбольный мяч, волан и пушечное ядро. Их характеристики приведены в таблице 1.

Таблица 1. Характеристики тел

Объект	$m$ , кг	$r$ , м	$C_d$
Тело без сопротивления	1.0	1.0	0.0
Мяч для гольфа	0.046	0.021	0.25
Теннисный мяч	0.057	0.033	0.50
Бейсбольный мяч	0.145	0.0365	0.30
Волан	0.005	0.035	0.65
Пушечное ядро	4.0	0.05	0.47

Во всех случаях использовались одинаковые параметры физической модели:  $v_0 = 50$  м/с,  $g = 9.81$  м/с<sup>2</sup>,  $\rho = 1.225$  кг/м<sup>3</sup>. Моделирование проводилось для трёх углов броска: 30, 45 и 60 градусов под углом к горизонту. Эти вводные параметры позволяют сравнить поведение объектов как для более пологой, так и для более крутой траектории.

## 2. Численные методы и схема вычислительного эксперимента

### 2.1. Используемые численные методы

Для численного решения данной системы уравнений были использованы две схемы с фиксированным шагом по времени: схема на основе метода Эйлера и метод Рунге-Кутты 4-го порядка [3]. Выбор именно этих схем обусловлен разницей между ними по точности и вычислительной сложности, что позволяет наглядно сопоставить их использование.

Схема на основе метода Эйлера применялась как наиболее простой численный подход к интегрированию системы обыкновенных дифференциальных уравнений. Данный метод имеет первый порядок точности и позволяет получить приближённое решение последовательным переходом от одного временного слоя к следующему. Несмотря на простоту реализации, метод Эйлера удобен в качестве базового ориентира при сравнении с более точными схемами.

В качестве основного метода фиксированного шага использовался классический метод Рунге-Кутты 4-го порядка. Его применение позволяет существенно уменьшить погрешность вычислений по сравнению с методом Эйлера при том же шаге интегрирования. В рамках настоящей работы именно этот метод рассматривался как основной инструмент численного моделирования траекторий, поскольку он обеспечивает хорошее сочетание точности и вычислительной устойчивости.

Для компактной записи введём вектор состояния

$$u = (x, y, v_x, v_y)^T,$$

тогда задача сводится к задаче Коши

$$\frac{du}{dt} = f(t, u), u(0) = u_0.$$

В используемой схеме на основе метода Эйлера переход к следующему шагу по времени осуществляется по формуле

$$u_{n+1} = u_n + hf(t_n, u_n).$$

Для метода Рунге-Кутты 4-го порядка использовалась схема

$$u_{n+1} = u_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

где

$$k_1 = f(t_n, u_n),$$

$$k_2 = f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_1\right),$$

$$k_3 = f\left(t_n + \frac{h}{2}, u_n + \frac{h}{2}k_2\right),$$

$$k_4 = f(t_n + h, u_n + hk_3).$$

Для дополнительного сопоставления результатов использовалось адаптивное численное решение, полученное с помощью функции `solve_ivp` библиотеки SciPy, реализующей метод RK45 [4]. Это решение использовалось как численный ориентир при анализе точности схем фиксированного шага.

## 2.2. Схема эксперимента и критерии сравнения

Численный эксперимент проводился для всех выбранных объектов при трёх углах броска: 30, 45 и 60 градусов к горизонту. Для каждого случая траектория рассчитывалась с использованием схемы на основе метода Эйлера, метода Рунге-Кутты 4-го порядка и адаптивного численного решения RK45.

Для верификации реализованных схем отдельно рассматривался случай с телом с нулевым коэффициентом сопротивления формы, для которого основные характеристики движения могут быть получены аналитически. Это позволило сопоставить численные результаты с точным решением классической задачи о движении тела, брошенного под углом к горизонту.

Для движения с учётом сопротивления воздуха в качестве численного ориентира использовалось адаптивное решение, полученное с помощью `solve_ivp` библиотеки SciPy [4]. Сравнение с ним проводилось для оценки точности схем с фиксированным шагом.

В качестве основных характеристик траектории рассматривались дальность полёта, максимальная высота и время полёта. Для оценки различий между методами вычислялись относительные ошибки указанных величин. Дополнительно анализировалось уменьшение этих характеристик по сравнению с вакуумным случаем.

Для исследования сходимости численных схем было выполнено дополнительное сравнение результатов при различных значениях шага интегрирования. Это позволило оценить, как изменение шага влияет на точность метода Эйлера и метода Рунге-Кутты 4-го порядка.

## 3. Результаты моделирования и их анализ

### 3.1. Моделирование при отсутствии сопротивления воздуха

Для проверки корректности реализованных численных схем был рассмотрен случай отсутствия сопротивления воздуха ( $C_d = 0$ ). В этом случае основные характеристики движения (дальность  $L$ , максимальная высота  $H$  и время полёта  $T$ ) тела, брошенного под углом к горизонту, могут быть найдены аналитически, что позволяет использовать их в качестве эталона для сравнения. В таблице 2 приведены результаты для угла 45 градусов к горизонту.

Таблица 2. Результаты моделирования при нулевом сопротивлении

Метод	$L$ , м	$H$ , м	$T$ , с
Аналитическое решение	254.84	63.71	7.21
Метод Эйлера	254.88	63.73	7.21
Метод Рунге-Кутты	254.84	63.73	7.21

Как видно из таблицы 2, обе численные схемы корректно воспроизводят аналитическое решение. Схема на основе метода Эйлера даёт малую, но заметную погрешность, тогда как метод Рунге-Кутты 4-го порядка практически совпадает с аналитическими значениями по дальности, максимальной высоте и времени полёта. Это подтверждает корректность реализации используемых численных методов и позволяет перейти к анализу движения с учётом сопротивления воздуха.

### 3.2. Моделирование различных тел

Для анализа влияния аэродинамического сопротивления были рассмотрены несколько объектов, различающихся по массе, размеру и коэффициенту сопротивления. В таблице 3 приведены результаты моделирования при угле броска 45 градусов к горизонту.

**Таблица 3.** Результаты моделирования при различных коэффициентах сопротивления формы

Объект	$L$ , м	$H$ , м	$T$ , с	Отклонение $L$ , %
Объект с $C_d = 0$	254.84	63.71	7.21	0.00
Мяч для гольфа ( $C_d = 0.25$ )	140.90	44.7	6.01	-44.71
Теннисный мяч ( $C_d = 0.5$ )	68.0	26.68	4.6	-73.31
Бейсбольный мяч ( $C_d = 0.3$ )	132.87	43.01	5.89	-47.86
Волан ( $C_d = 0.65$ )	8.07	4.46	1.86	-96.83
Ядро ( $C_d = 0.47$ )	229.51	60.16	7.0	-9.94

Из таблицы 3 видно, что наличие сопротивления воздуха приводит к уменьшению всех основных характеристик траектории по сравнению со случаем отсутствия сопротивления. Наиболее сильное отклонение от идеализированного случая наблюдается у волана, для которого дальность полёта уменьшается особенно резко. Наиболее близким к случаю без сопротивления оказывается пушечное ядро, что объясняется его большой массой и сравнительно меньшим влиянием аэродинамического торможения.

Для дополнительного анализа была рассмотрена зависимость дальности полёта от угла броска. Соответствующие результаты приведены в таблице 4.

**Таблица 4.** Зависимость дальности полёта от угла броска

Объект	$L(30^\circ)$ , м	$L(45^\circ)$ , м	$L(60^\circ)$ , м
Объект с $C_d = 0$	220.7	254.84	220.7
Мяч для гольфа ( $C_d = 0.25$ )	134.53	140.9	117.89
Теннисный мяч ( $C_d = 0.5$ )	69.48	68.0	55.1
Бейсбольный мяч ( $C_d = 0.3$ )	127.76	132.87	110.84
Волан ( $C_d = 0.65$ )	8.98	8.07	6.19
Ядро ( $C_d = 0.47$ )	203.09	229.51	197.41

Данные таблицы 4 показывают, что при отсутствии сопротивления воздуха наибольшая дальность достигается при угле в 45 градусов, что соответствует классическому результату. Однако при наличии сопротивления воздуха картина меняется. Для тел с более сильным аэродинамическим торможением, таких как теннисный мяч и волан, наибольшая дальность в исследованном наборе углов достигается уже при угле в 30 градусов. Это указывает на смещение эффективного оптимального угла броска в сторону меньших значений.

### 3.3. Сравнение точности численных методов и сходимости

Для оценки точности численных схем было проведено сравнение результатов методов фиксированного шага с адаптивным численным решением RK45. В таблице 5 приведены максимальные относительные ошибки схемы на основе метода Эйлера и метода Рунге-Кутты 4-го порядка для всех исследованных углов броска.

Таблица 5. Максимальные относительные ошибки численных методов относительно RK45

Угол, в градусах	Метод	Ошибка по $L$ , %	Ошибка по $H$ , %	Ошибка по $T$ , %
30°	Эйлер	0.2241	0.2076	0.1898
30°	RK4	0.0	0.0	0.0
45°	Эйлер	0.2192	0.2229	0.193
45°	RK4	0.0	0.0	0.0
60°	Эйлер	0.2164	0.2304	0.196
60°	RK4	0.0	0.0	0.0

Как следует из таблицы 5, метод Рунге-Кутты 4-го порядка обеспечивает существенно более высокую точность по сравнению со схемой на основе метода Эйлера при одинаковом шаге интегрирования. Для метода Эйлера максимальные относительные ошибки остаются малыми, однако заметно превосходят ошибки RK4. Таким образом, схема Эйлера может рассматриваться как простая базовая схема, тогда как RK4 является более предпочтительным методом для расчёта траекторий в рамках данной задачи.

Для дополнительного анализа было проведено исследование сходимости численных схем на примере теннисного мяча при угле броска 45°. Результаты приведены в таблице 6.

Таблица 6. Сходимость численных методов

$dt$ , с	Ошибка по $L$ (Эйлер), %	Ошибка по $L$ (RK4), %	Ошибка по $H$ (Эйлер), %	Ошибка по $H$ (RK4), %
0.1	0.9367	0.0008	1.5748	0.0024
0.05	0.4402	0.0003	0.8022	0.0022
0.01	0.0839	0.0001	0.1638	0.0
0.005	0.0417	0.0	0.082	0.0
0.001	0.0083	0.0	0.0164	0.0

Из таблицы 6 видно, что при уменьшении шага интегрирования ошибка обеих схем уменьшается, что соответствует ожидаемому поведению численных методов. При этом метод Рунге-Кутты 4-го порядка демонстрирует значительно более быстрое уменьшение ошибки по сравнению со схемой на основе метода Эйлера. Это дополнительно подтверждает более высокую точность RK4 при решении рассматриваемой системы нелинейных дифференциальных уравнений.

## 4. Заключение

В работе было проведено численное моделирование движения тел, брошенных под углом к горизонту, с учётом квадратичного сопротивления воздуха. Реализованные численные схемы показали корректную работу при проверке на случае отсутствия сопротивления воздуха, а сравнение с адаптивным решением RK45 подтвердило существенно более высокую точность метода Рунге-Кутты 4-го порядка по сравнению со схемой на основе метода Эйлера.

Результаты моделирования показали, что сопротивление воздуха существенно уменьшает дальность полёта, максимальную высоту и время движения, причём степень этого влияния

определяется физическими параметрами тела. Для объектов с сильным аэродинамическим торможением наблюдается смещение эффективного оптимального угла броска в сторону меньших значений по сравнению с классическим случаем. Полученные результаты подтверждают, что даже сравнительно простая модель с квадратичным сопротивлением воздуха позволяет адекватно описывать качественные и количественные особенности движения различных тел.

## Литература

1. Parker G. W. Projectile motion with air resistance quadratic in the speed //Am. J. Phys. – 1977. – Vol. 45. – №. 7. – P. 606-610.
2. Yabushita K., Yamashita M., Tsuboi K. An analytic solution of projectile motion with the quadratic resistance law using the homotopy analysis method //Journal of Physics A: Mathematical and theoretical. – 2007. – Vol. 40. – №. 29. – P. 8403-8416.
3. Hairer E., Wanner G., Nørsett S. P. Solving ordinary differential equations I: Nonstiff problems. – Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. DOI: 10.1007/978-3-540-78862-1.
4. Virtanen P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python //Nature methods. – 2020. – Т. 17. – №. 3. – P. 261-272.

## Содержание

Предисловие .....	3
<b><i>В.Д. Дятлов, А.В. Филиппов</i></b> Адаптивные тензорные сети для анализа квантовой запутанности в гибридных квантово-классических моделях.....	4
<b><i>Л.Л. Калаишиков</i></b> Исследование архитектурных особенностей нейронных сетей для поиска сердечных заболеваний путём анализа ландшафтов функции потерь .....	14
<b><i>А.А. Маслий, Н.А. Староверова</i></b> Формирование репрезентативного корпуса и разработка многозадачной генеративной модели для автоматизированной оценки студенческих эссе.....	24
<b><i>А.А. Межеева</i></b> Сравнительный анализ эффективности алгоритмов сортировки .....	31
<b><i>Соанху Хавьэ Вилфриэд</i></b> Синтетическая аугментация ЭКГ диффузионной моделью DSAT-ECG для преодоления дисбаланса диагностических классов в наборе данных PTB-XL .....	39
<b><i>Е.Б. Титова, Р.Д. Махнёв</i></b> Структура левого нуля-пространства матрицы ограничений планарной пятииндексной транспортной задачи .....	47
<b><i>Г.В. Уткин</i></b> Необходимые и достаточные условия подобия целочисленных матриц обобщённой жордановой клетке.....	51
<b><i>А.В. Филиппов, В.Д. Дятлов</i></b> Конструирование метрик рабочего процесса: разработка и валидация интегрального индекса эффективности потока на основе метрик Lead Time и Cycle Time с учётом когнитивных факторов и организационной динамики .....	54
<b><i>Р.С. Якунин</i></b> Численное моделирование движения тела при учёте сопротивления воздуха .....	64

# **Математика и ИТ – вместе в цифровое будущее**

Сборник трудов V Молодежной школы

*Нижний Новгород, 20–24 апреля 2026 г.*

*Под ред. К.А. Баркалова, Т.Г. Смирновой*

Публикуется в авторской редакции

ННГУ им. Н.И. Лобачевского.  
603022, Н. Новгород, пр. Гагарина, 23.