

ISSN 2686-679X

ВЕСТНИК РГГУ

Серия
«Информатика.
Информационная безопасность.
Математика»

Научный журнал

RSUH/RGGU BULLETIN

“Information Science.
Information Security. Mathematics”
Series

Academic Journal

Основан в 2018 г.
Founded in 2018

3
2026

VESTNIK RGGU. Seriya "Informatica. Informacionnaya bezopasnost. Matematika"

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" Series
Academic Journal

There are 4 issues of the printed version of the journal a year.

Founder and Publisher

Russian State University for the Humanities (RSUH)

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" series is included: in the Russian Science Citation Index; in the List of leading scientific journals and other editions for publishing PhD research findings peer-reviewed publications fall within the following research area:

1.1.6. Computational Mathematics (physical and mathematical sciences)

2.3.6. Information security methods and systems, information security
(technical science)

2.3.8. Informatics and information processes (technical science)

Objectives and areas of research

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" series publishes the results of research by scientists from RSUH and other universities and other Russian and foreign academic institutions. The areas covered by contributions include theoretical and applied computer science, up-to-date IT, means and technologies of information protection and information security as well as the issues of theoretical and applied mathematics including analytical and imitation models of different processes and objects. Special emphasis is put on articles and reviews covering research in indicated directions in the areas of social and humanitarian problems and also issues of personnel training for these directions.

RSUH/RGGU BULLETIN. "Information Science. Information Security. Mathematics" series is registered by Federal Service for Supervision of Communications Information Technology and Mass Media. 25.05.2018, reg. No. FS77-72977

Editorial staff office: bldg. 6, bld. 6, Miusskaya sq., Moscow, Russia, 125047

e-mail: dalex@inbox.ru

ВЕСТНИК РГГУ. Серия «Информатика. Информационная безопасность. Математика»
Научный журнал

Выходит 4 номера печатной версии журнала в год.

Учредитель и издатель – Российский государственный гуманитарный университет (РГГУ)

ВЕСТНИК РГГУ, серия «Информатика. Информационная безопасность. Математика», включен: в систему Российского индекса научного цитирования (РИНЦ); в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук по следующим научным специальностям и соответствующим им отраслям наук:

1.1.6. Вычислительная математика (физико-математические науки)

2.3.6. Методы и системы защиты информации, информационная безопасность (технические науки)

2.3.8. Информатика и информационные процессы (технические науки)

Цели и область

В журнале «Вестник РГГУ», серия «Информатика. Информационная безопасность. Математика», публикуются результаты научных исследований ученых и специалистов РГГУ, а также других университетов и научных учреждений России и зарубежных стран. Направления публикаций включают теоретическую и прикладную информатику, современные информационные технологии, методы, средства и технологии защиты информации и обеспечения информационной безопасности, а также проблемы теоретической и прикладной математики, включая разработку аналитических и имитационных моделей процессов и объектов различной природы. Особое внимание уделяется статьям и обзорам, посвященным исследованиям по указанным направлениям в области социальных и гуманитарных проблем, а также вопросам подготовки кадров по соответствующим специальностям для данных направлений.

ВЕСТНИК РГГУ. Серия «Информатика. Информационная безопасность. Математика», зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций 25.05.2018 г., регистрационный номер ПИ № ФС77-72977.

Адрес редакции: 125047, г. Москва, вн. тер. г. муниципальный округ Тверской, Мусская пл., д. 6, стр. 6

Электронный адрес: dalex@inbox.ru

Founder and Publisher

Russian State University for the Humanities (RSUH)

Editor-in-chief

E.N. Nadezhdin, Dr. of Sci. (Engineering), professor, Russian State University for the Humanities (RSUH), Moscow, Russian Federation

Editorial Board

V.I. Korolev, Dr. of Sci. (Engineering), professor, The Institute of Informatics Problems of the Russian Academy of Sciences (IPI RAN), Moscow, Russian Federation (*deputy editor-in-chief*)

D.A. Mityushin, Cand. of Sci. (Engineering), associate professor, Russian State University for the Humanities (RSUH), Moscow, Russian Federation (*executive secretary*)

L.A. Aslanyan, Dr. of Sci. (Physics and Mathematics), professor, corresponding member, National Academy of Sciences of the Republic of Armenia, Institute for Informatics and Automation Problems of the National Academy of Sciences of the Republic of Armenia, Yerevan, Republic of Armenia

S.N. Baibekov, Dr. of Sci. (Engineering), professor, Kazakh University of Technology and Business, Astana, Republic of Kazakhstan

S.B. Veprev, Dr. of Sci. (Engineering), professor, Russian Presidential Academy of National Economy and Public Administration, Moscow, Russian Federation

G.S. Ivanova, Dr. of Sci. (Engineering), professor, Bauman Moscow State Technical University, Moscow, Russian Federation

V.M. Maximov, Dr. of Sci. (Physics and Mathematics), professor, Russian State University for the Humanities (RSUH), Moscow, Russian Federation

R.S. Motul'skii, Dr. of Sci. (Pedagogics), professor, Institute of Modern Knowledge, Minsk, Republic of Belarus

Yu.I. Ozhigov, Dr. of Sci. (Physics and Mathematics), professor, Lomonosov Moscow State University, Moscow, Russian Federation

S.M. Sokolov, Dr. of Sci. (Physics and Mathematics), professor, Keldysh Institute of Applied Mathematics, Moscow, Russian Federation

V.A. Tsvetkova, Dr. of Sci. (Engineering), professor, Library for Natural Sciences of the RAS, Moscow, Russian Federation

Executive editor:

D.A. Mityushin, Cand. of Sci. (Engineering), associate professor,
Russian State University for the Humanities (RSUH)

Учредитель и издатель

Российский государственный гуманитарный университет (РГГУ)

Главный редактор

Е.Н. Надеждин, доктор технических наук, профессор, Российский государственный гуманитарный университет (РГГУ), Москва, Российская Федерация

Редакционная коллегия

В.И. Королев, доктор технических наук, профессор, ФГУ «Федеральный исследовательский центр «Информатика и управление» РАН, Москва, Российская Федерация (*заместитель главного редактора*)

Д.А. Митюшин, кандидат технических наук, доцент, Российский государственный гуманитарный университет (РГГУ), Москва, Российская Федерация (*ответственный секретарь*)

Л.А. Асланян, доктор физико-математических наук, профессор, член-корреспондент Национальной академии наук Республики Армения, Институт проблем информатики и автоматизации НАН Республики Армения, Ереван, Республика Армения

С.Н. Байбеков, доктор технических наук, профессор, Казахский университет технологий и бизнеса, Астана, Республика Казахстан

С.Б. Вепрев, доктор технических наук, профессор, Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации (РАНХиГС), Москва, Российская Федерация

Г.С. Иванова, доктор технических наук, профессор, Московский государственный технический университет им. Н.Э. Баумана, Москва, Российская Федерация

В.М. Максимов, доктор физико-математических наук, профессор, Российский государственный гуманитарный университет (РГГУ), Москва, Российская Федерация

Р.С. Мотульский, доктор педагогических наук, профессор, Институт современных знаний, Минск, Республика Беларусь

Ю.И. Ожигов, доктор физико-математических наук, профессор, Московский государственный университет им. М.В. Ломоносова (МГУ), Москва, Российская Федерация

С.М. Соколов, доктор физико-математических наук, профессор, Институт прикладной математики им. М.В. Келдыша РАН, Москва, Российская Федерация

В.А. Цветкова, доктор технических наук, профессор, Библиотека по естественным наукам РАН, Москва, Российская Федерация

Ответственный за выпуск:

Д.А. Митюшин, кандидат технических наук, доцент,
Российский государственный гуманитарный университет (РГГУ)

CONTENTS

Information Science

<i>Nikolai P. Artyukhin, Marina Yu. Baryshnikova</i> Parameter-efficient fine-tuning methods for specialized large language models	8
<i>Elena O. Shershneva, Artur M. Silkin</i> Design and development of an automated system for scheduling consultations of university lecturers	26
<i>Andrei P. Titov, Natalia V. Grishina, Daria N. Titova</i> Comparative analysis of collision detection algorithms in simulation tasks	51
<i>Andrei T. Soldatov, Nail Z. Sultanov</i> GPU memory management system with deferred buffer reuse for WebGPU	72

Information Security

<i>Sergei I. Kozminykh, Aleksandr Yu. Dubinin</i> Directions and methods for detecting and countering APT attacks on information systems	94
<i>Dzhakhed Rakhmani, Anna V. Sugrobova</i> Fault-tolerant authentication architecture through external identity providers in critical applications	114
<i>Liliya A. Ezhova</i> Information security as an element of sustainability of digital ecosystems	141
<i>Aleksandr V. Vinokurov</i> Structural and parametric synthesis of a user biometric authentication system	155

СОДЕРЖАНИЕ

Информатика

<i>Николай П. Артюхин, Марина Ю. Барышникова</i> Параметрически-эффективные методы дообучения для специализированных больших языковых моделей	8
<i>Елена О. Шершнева, Артур М. Силкин</i> Проектирование и разработка автоматизированной системы составления графика консультаций преподавателей вуза	26
<i>Андрей П. Титов, Наталия В. Гришина, Дарья Н. Титова</i> Сравнительный анализ алгоритмов обнаружения коллизий в задачах имитационного моделирования	51
<i>Андрей Т. Солдатов, Наиль З. Султанов</i> Система управления памятью на видеоускорителе с отложенным повторным использованием буферов для WebGPU	72

Информационная безопасность

<i>Сергей И. Козьминых, Александр Ю. Дубинин</i> Направления и методы обнаружения и противодействия APT-атакам на объектах информатизации	94
<i>Джахед Рахмани, Анна В. Сугрובה</i> Архитектура отказоустойчивой аутентификации через внешних поставщиков идентификации в критичных приложениях	114
<i>Лилия А. Ежова</i> Информационная безопасность как элемент устойчивости цифровых экосистем	141
<i>Александр В. Винокуров</i> Структурно-параметрический синтез биометрической системы аутентификации пользователя	155

Параметрически-эффективные методы дообучения для специализированных больших языковых моделей

Николай П. Артюхин

Московский государственный технический университет
им. Н.Э. Баумана, Москва, Россия, arperinn@gmail.com

Марина Ю. Барышникова

Московский государственный технический университет
им. Н.Э. Баумана, Москва, Россия, baryshnikovam@bmstu.ru

Аннотация. Активное развитие больших языковых моделей расширяет возможности автоматизации обработки текстовых данных, однако их внедрение в специализированные предметные области сдерживается высокими вычислительными затратами на их полное дообучение. В данной работе рассматривались основные параметрически-эффективные методы дообучения моделей (*Parameter-Efficient Fine-Tuning, PEFT*): метод добавления адаптеров (*Adapter Tuning*), метод низкоранговой адаптации (*LoRA/QLoRA*), методы добавления «подсказок» (*Prompt Tuning* и *Prefix Tuning*), метод выборочной настройки (*BitFit*), – применительно к русскоязычной медицинской предметной области, которая остается недостаточно изученной. Данные методы позволяют адаптировать предварительно обученные модели под конкретные задачи, обновляя только часть параметров модели. Целью исследования является сравнительный анализ параметрически-эффективных методов (*LoRA, BitFit, Prompt Tuning, Adapter Tuning*) и метода полного дообучения (*Full fine-tuning*) русскоязычной модели в предметной области медицины по скорости дообучения и качеству генерируемых ответов, оцениваемых по значению перплексии и метрике *BLEU*. В результате проведенного исследования было получено, что самым быстрым из рассмотренных параметрически-эффективных методов дообучения русскоязычной модели на основе *GPT2* является *BitFit* (более чем в 2 раза быстрее по сравнению с методом полного дообучения). По метрике *BLEU* лучший результат среди параметрически-эффективных методов показал метод низкоранговой адаптации (*LoRA*), улучшив качество базовой модели в 9 раз, а по полученному значению перплексии лучшим оказался *BitFit*.

Ключевые слова: большие языковые модели, параметрически-эффективные методы дообучения, полное дообучение, метод низкоранговой адаптации, добавление адаптеров, предобученная модель, медицина, перплексия, BLEU

Для цитирования: Артюхин Н.П., Барышникова М.Ю. Параметрически-эффективные методы дообучения для специализированных больших языковых моделей // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 8–25. DOI: 10.28995/2686-679X-2026-3-8-25

Parameter-efficient fine-tuning methods for specialized large language models

Nikolai P. Artyukhin

*Bauman Moscow State Technical University, Moscow, Russia,
arperinn@gmail.com*

Marina Yu. Baryshnikova

*Bauman Moscow State Technical University, Moscow, Russia,
baryshnikovam@bmstu.ru*

Abstract. The rapid development of large language models expands the possibilities for automating text data processing, but their implementation in specialized subject areas is hampered by the high computational costs of their full fine-tuning. This paper examines the main Parameter-Efficient Fine-Tuning (PEFT) methods: Adapter Tuning, Low-Rank Adaptation (LoRA/QLoRA), Prompt Tuning and Prefix Tuning, and BitFit, as applied to the Russian-language medical domain, which remains understudied. These methods enable pre-trained models to be adapted to specific tasks by updating only some of the model's parameters. The aim of this study is to compare Parameter-Efficient methods (LoRA, BitFit, Prompt Tuning, Adapter Tuning) and the Full fine-tuning method for a Russian-language model in the medical domain in terms of retraining speed and the quality of the generated responses, assessed by the perplexity value and the BLEU metric. The study found that BitFit is the fastest of the considered Parameter-Efficient methods for fine-tuning a Russian-language model based on GPT2 (more than twice as fast as the Full fine-tuning method). In terms of the BLEU metric, the low-rank adaptation (LoRA) method demonstrated the best result among the Parameter-Efficient methods, improving the quality of the baseline model by a factor of 9. BitFit proved to be the best in terms of the obtained perplexity value.

Keywords: large language models, parameter-efficient fine-tuning, full fine-tuning, low-rank adaptation, adapter tuning, pretrained model, medicine, perplexity, BLEU

For citation: Artyukhin, N.P. and Baryshnikova, M.Yu. (2026), "Parameter-efficient fine-tuning methods for specialized large language models", *RSUH/RGGU Bulletin. "Information Science. Information Security. Mathematics" Series*, no. 3, pp. 8–25, DOI: 10.28995/2686-679X-2026-3-8-25

Введение

Большие языковые модели стали новым значимым прорывом в области искусственного интеллекта (ИИ), меняющим способы его использования от решения узких задач, требующих больших объёмов данных с готовыми ответами для машинного обучения (размеченные данные), к генеративным моделям, способным к дообучению на небольшом количестве примеров (или вообще без примеров) с готовыми ответами и при этом имеющим более широкие возможности применения [Андрейченко, Гусев 2023, с. 48]. Они позволяют автоматизировать процессы в самых различных областях, например, в медицине, юриспруденции, финансах, сфере информационных технологий и т. д.

В последнее время широкое распространение получили модели общего назначения, обученные на крупных корпусах текстов, которые, однако, могут давать ошибочные или неточные ответы в узкоспециализированных предметных областях из-за нехватки детальных знаний. Это, в свою очередь, вызывает потребность их дообучения на специализированных корпусах текстов.

Проведенные исследования показывают, что более крупные модели, как правило, обеспечивают более высокую производительность. Недостатком является высокая стоимость точной настройки и хранения всех параметров модели, что делает актуальным новое направление исследований, посвященное адаптации предобученных моделей алгоритмами, которые оптимизируют небольшую часть параметров, сохраняя при этом остальные неизменными, что значительно сокращает затраты на вычисление и хранение.

Методы дообучения больших языковых моделей

По данным целого ряда статей [Zheng et al. 2024; Kim, Hwang, Jang 2025] и др. выделяют следующие методы дообучения больших языковых моделей для решения специализированных задач:

- полное дообучение (*Full fine-tuning*);
- параметрически-эффективное дообучение (*PEFT, Parameter-efficient fine-tuning*) [Ding et al. 2023; Han et al. 2024];
- добавление адаптеров (*Adapter Tuning*);
- низкоранговая адаптация (*LoRA, QLoRA*);
- добавление подсказок (*Prompt Tuning, Prefix Tuning*);
- выборочная настройка (*selective methods – BitFit, Diff Pruning*).
- гибридные методы (комбинация нескольких *PEFT* методов, использование обучения с подкреплением).

Иногда вместо перечисленных выше методов дообучения моделей, которые обновляют веса модели, используют метод контекстного обучения (*In-Context Learning, ICL*). В этом случае процесс адаптации происходит без обновления весов (параметров) модели, то есть без традиционного дообучения (*fine-tuning*). Модель обучается новым задачам «на лету», используя только примеры (демонстрации), предоставленные в контексте (промпте) запроса. В зависимости от количества предоставленных примеров *ICL* делят на три основных типа:

- *Few-shot Learning* – обучение на нескольких примерах (обычно от 2 до 10);
- *One-shot Learning* – обучение на одном примере;
- обучение без примеров: модель полностью полагается на знания, полученные во время предварительного обучения.

Краткое описание принципов дообучения больших языковых моделей, используемых в указанных методах, и их математические модели приведены в табл. 1.

Следует заметить, что метод полного дообучения, обеспечивая наилучшее качество ответов, требует большого объема памяти и вычислительных ресурсов, достаточных для хранения и обработки всех компонентов, обновляемых при обучении. Кроме того, его применение может привести к эффекту забывания, когда обновленная модель начинает хуже решать те задачи, которые решала предобученная модель. Это происходит из-за того, что в процессе дообучения модифицируются все веса исходной модели.

Таблица 1

Методы дообучения больших языковых моделей

Название метода	Суть метода	Математическая модель
Полное дообучение (<i>Full fine-tuning</i>)	Обновляются все веса (параметры) и слои предобученной модели [Xu et al. 2023]	$y = W'x$, где y – вектор выходных значений; W' – обновленная матрица весов; x – вектор входных значений
<i>LoRA (Low-Rank Adaptation)</i>	Матрица обновления весов представлена в виде произведения двух низкоранговых матриц (рис. 1), что значительно сокращает количество обучаемых параметров [Zhu et al. 2024; Yang, Hu 2022]	$y = Wx + \Delta Wx = Wx + BAx$, где y – вектор выходных значений; W – исходная матрица весов; x – вектор входных значений; $B \in R^{d \times r}$, $A \in R^{r \times k}$ – обучаемые низкоранговые матрицы; $r \ll \min(d, k)$ – ранг разложения (гиперпараметр, обычно r in [4, 64]) [Hu et al. 2021]
<i>QLoRA (Quantized Low-Rank Adaptation)</i>	Модификация метода <i>LoRA</i> , сочетающая низкоранговую адаптацию с квантизацией весов, что позволяет дообучать модели, представленные в 4-битном формате. Такой подход значительно снижает требования к памяти при сохранении качества [Dettmers et al. 2023]	$y = \text{dequantize}(W_{4bit})x + (\alpha/r)BAx$, где y – вектор выходных значений; W_{4bit} – квантизованная исходная матрица весов (веса нейронной сети преобразованы из высокоточных 32-битных чисел с плавающей запятой в формат 4-битных чисел); x – вектор входных значений; $B \in R^{d \times r}$, $A \in R^{r \times k}$ – обучаемые низкоранговые матрицы, α – гиперпараметр масштабирования; $r \ll \min(d, k)$ – ранг разложения (гиперпараметр, обычно r in [4, 64])
<i>Adapter Tuning</i>	В архитектуру трансформера (рис. 2) добавляются небольшие дополнительные модули (адаптеры). Эти модули обучаются, в то время как основные параметры модели остаются «замороженными» [Houlsby et al. 2019]	$\text{Adapter}(y) = W_{up} \times \text{ReLU}(W_{down} \times y) + y$, где y – вектор выходных значений слоя, после которого ставится адаптер; W_{down} in $R^{d \times r}$ – слой понижения размерности; ReLU – нелинейная функция активации $f(x) = \max(0, x)$; W_{up} in $R^{d \times r}$ – слой повышения размерности; $r \ll d$ – размерность «бутылочного горлышка» (гиперпараметр <i>bottleneck</i>)

Окончание табл. 1

Название метода	Суть метода	Математическая модель
<i>Prompt Tuning</i>	Обучаются непрерывные промпт-токены, которые соединяются с входными векторами (эмбедингами) [Lester, Al-Rfou, Constant 2021], а основные параметры модели остаются неизменными	$E' = [P_1, P_2, \dots, P_k; E_{input}]$, где E' – обновленный входной вектор (эмбединг), P_i in R^d – обучаемые векторы промптов (<i>soft prompts</i>); k – количество промпт-токенов (гиперпараметр); E_{input} – векторы (эмбединги) входных токенов
<i>Prefix Tuning</i>	Модификация метода <i>Prompt Tuning</i> , при котором обучаемые промпт-токены добавляются на каждом слое трансформера, а не только на уровне входных векторов (эмбедингов) [Li, Liang 2021]	Каждый слой модели-трансформера (1): $[h_1^{(l)}, \dots, h_k^{(l)}, h_{input}^{(l)}] = \text{Transformer Layer}([P_1^{(l)}, \dots, P_k^{(l)}, h_{input}^{(l-1)}])$, где $h_i^{(l)}$ – скрытые состояния префиксных токенов слоя l (не передаются на следующий слой); $h_{input}^{(l)}$ – скрытые состояния входных токенов слоя l трансформера (передаются на следующий слой); $P_i^{(l)}$ – обучаемые векторы промптов слоя l ; $h_{input}^{(l-1)}$ – скрытые состояния входных токенов слоя $(l-1)$ трансформера
<i>BitFit (Bias-term Fine-tuning)</i>	Обучаются только параметры смещения (<i>bias terms</i>) в модели, в то время как все весовые матрицы остаются «замороженными» [Ben-Zaken, Ravfogel, Goldberg 2021]	$y = Wx + b'$, где y – вектор выходных значений; W – исходная матрица весов; x – вектор входных значений; $b' = b - \nu \Delta b$ – обновленный вектор смещений

В отличие от полного дообучения методы параметрически-эффективного дообучения (*PEFT*), при использовании которых обновляется только небольшое подмножество параметров исходной (предобученной) модели, не только существенно снижают требования к памяти, но и позволяют решить проблему эффекта забывания, то есть адаптированная модель не забывает ранее изученную информацию.

При этом данные методы могут использовать различные подходы к сокращению количества обучаемых параметров: если в методе

LoRA эта задача решается за счет применения низкоранговых матриц изменения весов (рис. 1), то в методе *Adapter Tuning* обучению подлежит небольшое количество дополнительных модулей (адаптеров), встроенных в архитектуру трансформера (рис. 2).

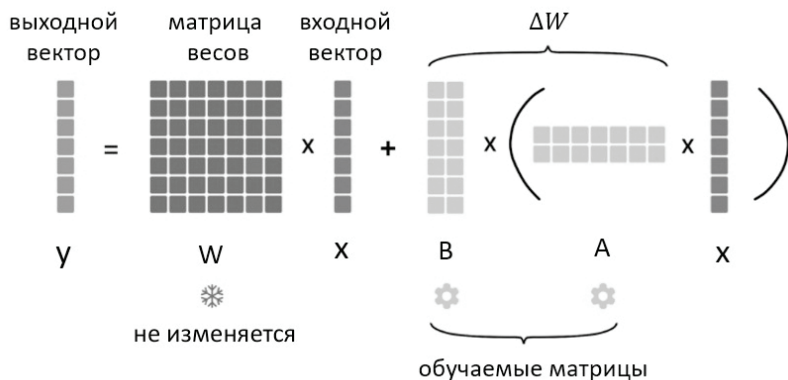


Рис. 1. Метод низкоранговой адаптации (*LoRA*)

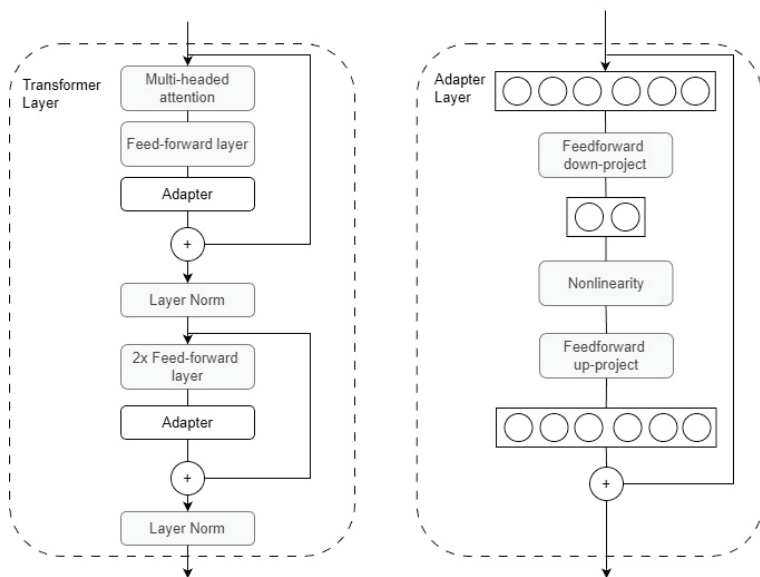


Рис. 2. Пример встраивания слоя адаптера в архитектуру нейронной сети

Теоретическое сравнение методов дообучения больших языковых моделей

Проведенный обзор существующих подходов к дообучению больших языковых моделей демонстрирует широкий спектр методов от полной перенастройки всех весов модели (*Full Fine-Tuning*) до параметрически-эффективного дообучения (*PEFT*), таких как *LoRA*, *Prefix Tuning*, *Adapter Tuning*, *BitFit* и т. д. Однако многообразие технических решений создает проблему выбора наилучшего подхода, поскольку не существует универсального метода одинаково эффективного во всех сценариях. Каждый подход представляет собой поиск компромисса между вычислительной эффективностью, скоростью обучения и качеством ответов модели, полученных в результате проведения дообучения.

В данной работе будет производиться выбор метода дообучения базовой большой языковой модели (предобученной на русскоязычном корпусе) для ее использования в ИИ-ассистенте, который должен по заданным вопросам пользователей на тему своего здоровья направлять их к соответствующим специалистам и давать им общие рекомендации.

При дообучении базовой модели важны экономическая целесообразность и возможность ее частого обновления. Медицинские протоколы и рекомендации меняются, поэтому метод дообучения должен позволять быстро и дешево внедрять новые знания без необходимости проведения очень длительного и ресурсоемкого процесса обучения. Исходя из этого, необходимо сравнивать рассматриваемые методы по скорости процесса дообучения и по объему используемой памяти. Несмотря на ограниченность ресурсов, модель для медицинского консультанта должна демонстрировать высокую точность ответов, сопоставимую с полным дообучением (метод *Full fine-tuning*). Отклонение от эталонного качества ответов, получаемого методом полного дообучения, должно быть минимальным. Также необходимо избегать эффекта забывания, когда дообученная модель теряет способности к рассуждению (в данном случае базовая модель обучалась «формулированию мыслей» на русском языке), а только запоминает шаблоны ответов из обучающего набора данных. Наибольший риск забывания у метода полного дообучения, так как он обновляет все параметры базовой (предобученной) модели.

В табл. 2 представлены результаты теоретического сравнения методов дообучения больших языковых моделей на основе статей [Hu et al. 2021; Dettmers et al. 2023; Ben-Zaken, Ravfogel, Goldberg 2021; Lester, Al-Rfou, Constant 2021; Houlsby et al. 2019; Li, Liang 2021]. При этом используются следующие обозначения:

- ОВ – очень высокий уровень;
- В – высокий уровень;
- С – средний уровень;
- Н – низкий уровень;
- ОН – очень низкий уровень.

Краткие обозначения названий методов дообучения больших языковых моделей:

- *Full FT* – метод полного дообучения (*Full fine-tuning*);
- *LoRA/QLoRA* – метод низкоранговой адаптации из группы методов *PEFT*;
- *PT* – методы добавления подсказок (*Prefix Tuning, Prompt Tuning*) из группы методов *PEFT*;
- *Adapter* – метод добавления адаптеров (*Adapter Tuning*) из группы методов *PEFT*;
- *BitFit* – методы выборочной настройки из группы методов *PEFT*.

Критерии сравнения рассматриваемых методов дообучения моделей:

- относительное число обновляемых параметров предобученной (базовой) модели – от 0 до 100%;
- использование памяти (в сравнении с самым неэкономным по памяти рассматриваемым методом – *full fine-tuning*);
- скорость процесса дообучения (в сравнении с самым медленным по скорости дообучения рассматриваемым методом – *full fine-tuning*);
- качество результатов (в сравнении с *full fine-tuning*);
- риск забывания знаний предобученной модели (чем больше параметров базовой модели обновляется, тем выше риск).

Таблица 2

Сравнение методов дообучения больших языковых моделей

	Full FT	LoRA/QLoRA	PT	Adapter	BitFit
Обновляемые параметры (%)	100	0,1...1	0,01	1...5	0,01 (bias)
Память	ОВ	Н	ОН	С	ОН
Скорость обучения	ОН	В	В	С	ОВ
Качество результатов	ОВ	В	С	С	С/Н
Риск забывания	ОВ	Н	ОН	Н	ОН

Таким образом, по данным статей [Hu et al. 2021; Dettmers et al. 2023; Ben-Zaken, Ravfogel, Goldberg 2021; Lester, Al-Rfou, Constant 2021; Houlsby et al. 2019; Li, Liang 2021] наилучшим и самым сбалансированным по совокупности выделенных критериев среди рассмотренных методов дообучения больших языковых моделей в общем случае является метод низкоранговой адаптации из группы методов *PEFT (LoRA и QLoRA)*. Далее будет проведено сравнение рассматриваемых методов дообучения больших языковых моделей на примере конкретной базовой модели, набора данных, выбранных метрик качества и специализированной задачи дообучения модели для медицинского консультанта на основе ИИ.

Исследование качества результатов моделей и скорости их дообучения в зависимости от выбора соответствующего метода дообучения

Базовая модель для проведения дообучения для решения специализированной задачи – `ai-forever/rugpt3small_based_on_gpt2` с платформы *Hugging Face*¹. Выбор данной модели обусловлен ее доступностью (open source, модель с открытым исходным кодом) и небольшим размером для возможности проведения исследования в условиях ограниченного числа вычислительных ресурсов (ее дообучение можно проводить на пользовательских *GPU* или даже использовать только *CPU*).

Набор данных для проведения дообучения исходной большой языковой модели – `Mykes/rus_med_dialogues` получен с платформы *Hugging Face*². Он состоит из вопросов пользователей по теме медицины (столбец `user_question` на рис. 3) и эталонных ответов ИИ-ассистента на них (столбец `assistant_answer` на рис. 3). Количество обучающих примеров – 3010. Количество примеров для проведения тестирования – 335. Также в этом наборе данных указаны специализации врачей, которым соответствует заданный вопрос (столбцы `topic` и `to_doctor` на рис. 3). Таким образом, русскоязычная модель общего назначения должна научиться консультировать пользователей по вопросам, связанным со здоровьем, и направлять их к соответствующим врачам.

¹ Hugging Face. URL: https://huggingface.co/ai-forever/rugpt3small_based_on_gpt2 (дата обращения: 16.01.2026).

² Hugging Face. URL: https://huggingface.co/datasets/Mykes/rus_med_dialogues (дата обращения: 16.01.2026).

topic string · classes	user_question string · lengths	assistant_answer string · lengths	to_doctor string · classes
травматолог... 4.7%	187-133 18.6%	374-467 28.9%	травматолог 4.7%
кардиология	У меня учащенное сердцебиение после физической нагрузки. Это нормально...	Учащенное сердцебиение после физической нагрузки – это нормальная реакция...	кардиолог
терапия	Я принимаю антибиотик Амоксициллин уже 5 дней, но мой кашель не...	Понимаю, что кашель не проходит, несмотря на прием антибиотика...	терапевт
эндокринология	У меня постоянно скачет давление. Врач сказал, что у меня проблемы с...	По скачкам давления у Вас может быть действительно дело в щитовидке. Но для...	эндокринолог
травматология	У меня вот уже неделю болит колено после падения. Боль усиливается при ходьбе и подъеме по лестнице. Что мне делать?	Из-за боли в колене после падения, я рекомендую Вам обратиться к травматологу. Он осмотрит Вас, определит причину боли и назначит соответствующее лечение. Возможно, Вам потребуется рентген для исключения переломов или травмы связок. Пока не наблюдаете Вы у врача, можете прикладывать холод к месту боли, принимать обезболивающие препараты по инструкции и ограничить физические нагрузки. Но будьте осторожны и не занимайтесь самолечением.	травматолог

Рис. 3. Фрагмент датасета
для проведения дообучения базовой модели

В данной работе методы дообучения больших языковых моделей (полное дообучение, *LoRA*, *Prompt Tuning*, *Adapter Tuning*, *BitFit*) сравнивались по скорости дообучения и качеству полученных результатов. Качество дообученных моделей оценивалось по перплексии (*perplexity*) и *BLEU (Bilingual Evaluation Understudy)*.

Перплексия (*perplexity*, *PPL*) – это метрика качества языковых моделей, измеряющая их способность предсказывать следующее слово. Низкая перплексия означает, что модель хорошо предсказывает данные, а высокая – что модель «путается», то есть она имеет много равновероятных вариантов выбора каждого последующего слова при генерации ответов на вопросы пользователей. Перплексия принимает значения от 1 до бесконечности. Идеальное значение перплексии равно 1, то есть каждое последующее слово при генерации ответа выбирается со 100% вероятностью.

*BLEU (Bilingual Evaluation Understudy)*³ – это измерение различий между автоматическим переводом и эталонными переводами того же предложения, выполненными людьми (0 – полное несовпадение эталонного и сгенерированного ответов, 1 – полное совпадение эталонного и сгенерированного ответов). Поскольку целевая задача не является задачей машинного перевода, то высоких оце-

³ Что такое оценка BLEU? // Learn. URL: <https://learn.microsoft.com/ru-ru/azure/ai-services/translator/custom-translator/concepts/bleu-score> (дата обращения: 16.01.2026).

нок (близких к 1) по метрике *BLEU* получить не ожидается. Более высокие значения данной метрики будут означать, что модель использует термины эталонной медицинской специализации при генерации ответов на соответствующие вопросы пользователей.

Используемый язык программирования – *Python*⁴.

Используемые библиотеки:

- *transformers*⁵ для загрузки базовой модели и исходного набора данных для проведения дообучения, настройки гиперпараметров дообучения модели;
- *peft*⁶, содержащая реализации сравниваемых методов дообучения моделей.

Диаграммы длительности дообучения базовой модели рассматриваемыми методами, относительного количества обновляемых параметров (метод полного дообучения модели обновляет все параметры – 100%), перплексии и BLEU дообученных моделей представлены на рис. 4–7. Количество эпох обучения, проводимого каждым из методов, равно 3.

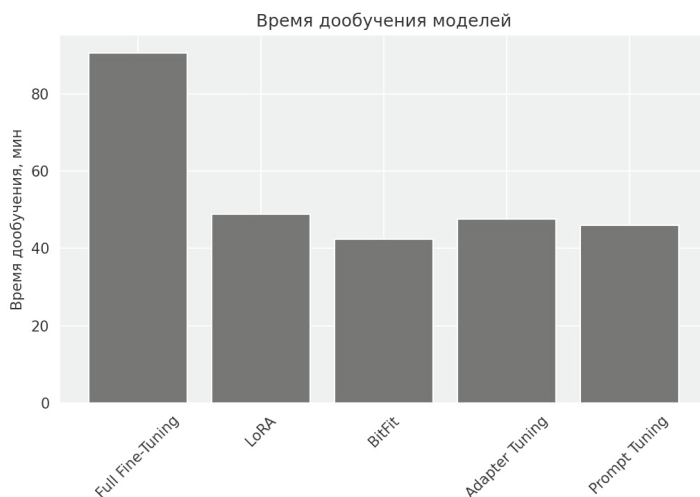


Рис. 4. Время дообучения базовой модели рассматриваемыми методами (в минутах)

⁴ Python. URL: <https://www.python.org/> (дата обращения: 16.01.2026).

⁵ Hugging Face // Transformers. URL: <https://huggingface.co/docs/transformers/index> (дата обращения: 16.01.2026).

⁶ peft 0.19.1. URL: <https://pypi.org/project/peft/> (дата обращения: 16.01.2026).



Рис. 5. Относительное количество обновляемых параметров рассматриваемыми методами дообучения

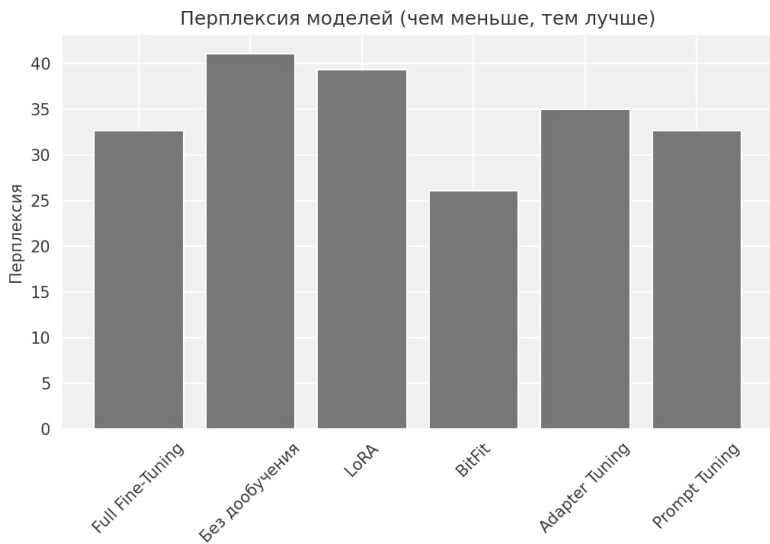


Рис. 6. Перплексия базовой модели, дообученной рассматриваемыми методами дообучения

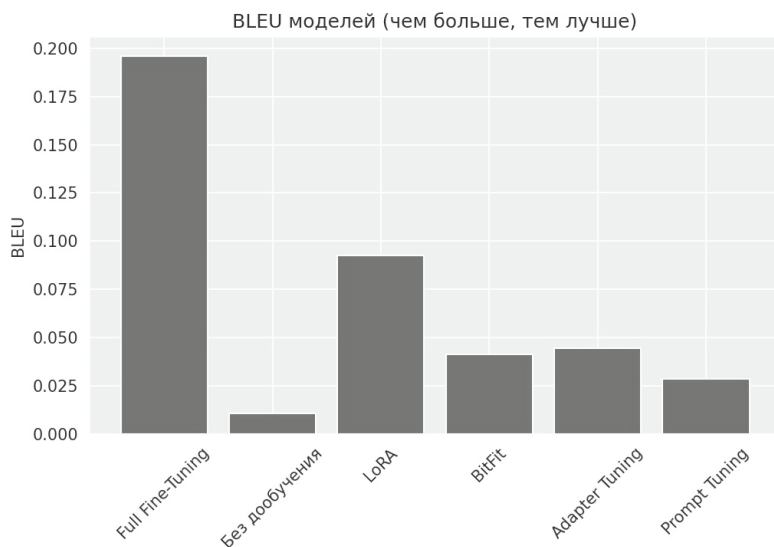


Рис. 7. BLEU базовой модели, дообученной рассматриваемыми методами дообучения

Заключение

В результате проведенного сравнительного анализа параметрически-эффективных методов (*PEFT*) и полного дообучения было получено:

1. Самым быстрым из параметрически-эффективных методов оказался *BitFit*, на 114% быстрее (42 минуты вместо 90 минут), чем полное дообучение, так как он обновляет только смещения (*bias*). Метод *LoRA* оказался медленнее на 15% (работал 48 минут), чем *BitFit*.

2. Лучшее значение перплексии показали методы полного дообучения и *BitFit*, однако в контексте решения специализированной задачи значения этой метрики не очень существенны (основным критерием в этом случае является не ухудшение значения перплексии по сравнению с базовой моделью). Все методы дообучения улучшили значение перплексии по сравнению с базовой моделью.

3. По метрике *BLEU* ожидаемо лучший результат показал метод полного дообучения, однако он затрачивает очень много вычислительных ресурсов и из-за этого далеко не всегда пригоден

к использованию. Из параметрически-эффективных методов лучший результат показал метод низкоранговой адаптации (*LoRA*). Базовая модель (без проведения дообучения) показала наиболее низкое значение по метрике *BLEU*. Метод *LoRA* улучшил качество результатов (по метрике *BLEU*) базовой модели в 9 раз, но уступил методу полного дообучения в 2 раза.

Результат *LoRA* и других параметрически-эффективных методов можно улучшить за счет увеличения числа эпох обучения, но при этом будут возрастать и затрачиваемое время, и ресурсы. Поэтому при подборе гиперпараметров обучающего алгоритма необходимо находить баланс между требованиями к качеству результата и к ограничению по используемым вычислительным ресурсам.

Таким образом, проведенное исследование показало, что наиболее оптимальным по совокупности выделенных критериев методом дообучения модели для ИИ-консультанта в медицинской сфере на основе вопросно-ответного набора данных является метод низкоранговой адаптации (*LoRA/QLoRA*).

Литература

- Андрейченко, Гусев 2023 – Андрейченко А.Е., Гусев А.В. Перспективы применения больших языковых моделей в здравоохранении // Национальное здравоохранение. 2023. Т. 4, № 4. С. 48–55. DOI: 10.47093/2713-069X.2023.4.4.48-55. EDN MPDTJU.
- Ben-Zaken, Ravfogel, Goldberg 2021 – Ben-Zaken E. Ravfogel S., Goldberg Y. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models // Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Vol. 2: Short Papers). Dublin: Association for Computational Linguistics, 2022. P. 1–9.
- Dettmers et al. 2023 – Dettmers T., Pagnoni A., Holtzman A., Zettlemoyer L. QLoRA: efficient finetuning of quantized LLMs // Advances in Neural Information Processing Systems. 2023. Vol. 36. P. 20418–20434.
- Ding et al. 2023 – Ding N. et al. Parameter-efficient fine-tuning of large-scale pre-trained language models // Nature Machine Intelligence. 2023. Vol. 5 (3). P. 220–235. DOI: 10.1038/s42256-023-00626-4.
- Han et al. 2024 – Han Z., Gao C., Liu J. et al. Parameter-efficient fine-tuning for large models: a comprehensive survey, 2024. URL: <https://doi.org/10.48550/arXiv.2403.14608> (дата обращения: 16.01.2026).
- Houlsby et al. 2019 – Houlsby N., Giurgiu A., Jastrzebski S. et al. Parameter-efficient transfer learning for NLP // Proceedings of the 36th International Conference on Machine Learning (ICML). Long Beach: PMLR, 2019. P. 2790–2799.

- Hu et al. 2021 – *Hu E.J., Shen Y., Wallis P. et al.* LoRA: Low-rank adaptation of large language models, 2021. URL: <https://www.alphaxiv.org/abs/2106.09685v2> (дата обращения: 21.01.2026).
- Kim, Hwang, Jang 2025 – *Kim G., Hwang S., Jang B.* Efficient compressing and tuning methods for large language models: a systematic literature review // *ACM Computing Surveys*. 2025. Vol. 57 (10). Art. 253. DOI: 10.1145/3728636.
- Lester, Al-Rfou, Constant 2021 – *Lester B., Al-Rfou R., Constant N.* The power of scale for parameter-efficient prompt tuning // *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. New York: Association for Computational Linguistics, 2021. P. 3045–3059.
- Li, Liang 2021 – *Li X.L., Liang P.* Prefix-tuning: optimizing continuous prompts for generation // *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*. New York: Association for Computational Linguistics, 2021. Vol. 1. P. 4582–4597.
- Xu et al. 2023 – *Xu L., Xie H., Qin S.-Z. J. и др.* Parameter-efficient fine-tuning methods for pretrained language models: a critical review and assessment, 2023. DOI: <https://doi.org/10.48550/arXiv.2312.12148> (дата обращения: 25.01.2026).
- Yang, Hu 2022 – *Yang G., Hu E.J.* Feature learning in infinite-width neural networks // *Proceedings of the 38th International Conference on Machine Learning (ICML)*. New York: PMLR, 2021. P. 11727–11737.
- Zheng et al. 2024 – *Zheng J., Hong H., Liu F. et al.* DragFT: adapting large language models with dictionary and retrieval augmented fine-tuning for domain-specific machine translation, 2024. URL: <https://arxiv.org/abs/2402.15061> (дата обращения: 23.01.2026).
- Zhu et al. 2024 – *Zhu J., Greenewald K., Nadjahi K. et al.* Asymmetry in low-rank adapters of foundation models // *Proceedings of the 41st International Conference on Machine Learning (ICML)*. Vienna: PMLR, 2024. P. 62369–62385.

References

- Andreychenko, A. and Gusev, A. (2023), “Perspectives on the application of large language models in healthcare”, *National Health Care (Russia)*, vol. 4 (4), pp. 48–55.
- Ben-Zaken, E., Ravfogel, S. and Goldberg, Y. (2022), “BitFit: simple parameter efficient fine-tuning for transformer-based masked language-models”, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Association for Computational Linguistics, Dublin, Ireland, pp. 1–9.
- Dettmers, T., Pagnoni, A., Holtzman, A. and Zettlemoyer L. (2023), “QLoRA: efficient finetuning of quantized LLMs”, *Advances in Neural Information Processing Systems*, vol. 36, pp. 20418–20434.
- Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.-M., Chen, W., Zhao, J., Du, Z., Liu, H., Zheng, H.-T., Li, Q., Zhang, M., Tang, J. and

- Sun, M. (2023), "Parameter-efficient fine-tuning of large-scale pre-trained language models", *Nature Machine Intelligence*, vol. 5 (3), pp. 220-235.
- Han, Z., Gao, C., Liu, J., Zhang, Q. and Mao, Y. (2024), "Parameter-efficient fine-tuning for large models: a comprehensive survey", available at: <https://doi.org/10.48550/arXiv.2403.14608> (Accessed 16 Jan. 2026).
- Houlsby, N., Giurghi, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M. and Gelly, S. (2019), "Parameter-efficient transfer learning for NLP", *Proceedings of the 36th International Conference on Machine Learning*, PMLR, Long Beach, CA, USA, pp. 2790–2799.
- Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W. (2022), "LoRA: low-rank adaptation of large language models", *International Conference on Learning Representations (ICLR)*, 25-29 April 2022, available at: <https://www.alphaxiv.org/abs/2106.09685v2>. (Accessed 21 Jan 2026).
- Kim, G., Hwang, S. and Jang, B. (2025), "Efficient compressing and tuning methods for large language models: a systematic literature review", *ACM Computing Surveys*, vol. 57 (10), article 253.
- Lester, B., Al-Rfou, R. and Constant, N. (2021), "The power of scale for parameter-efficient prompt tuning", *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, New York, USA, pp. 3045–3059.
- Li, X.L. and Liang, P. (2021), "Prefix-tuning: optimizing continuous prompts for generation", *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (Volume 1: Long Papers), Association for Computational Linguistics, New York, USA, pp. 4582–4597.
- Xu, L., Xie, H., Qin, S.-Z.J., Tao, X. and Li, F.L. (2023), "Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment", available at: <https://doi.org/10.48550/arXiv.2312.12148> (Accessed 25 Jan. 2026).
- Yang, G. and Hu, E.J. (2021), "Feature learning in infinite-width neural networks", *Proceedings of the 38th International Conference on Machine Learning*, PMLR, New York, USA, pp. 11727–11737.
- Zheng, J., Hong, H., Liu, F., Wang, X., Su, J., Liang, Y. and Wu, S. (2024), "DragFT: adapting large language models with dictionary and retrieval augmented fine-tuning for domain-specific machine translation", available at: <https://arxiv.org/abs/2402.15061> (Accessed 23 Jan. 2026).
- Zhu, J., Greenewald, K., Nadjahi, K., Borde, H.S.d.O., Gabrielsson, R.B., Choshen, L., Ghassemi, M., Yurochkin, D. and Solomon, J. (2024), "Asymmetry in Low-Rank Adapters of Foundation Models", *Proceedings of the 41st International Conference on Machine Learning*, PMLR, Vienna, Austria, pp. 62369–62385.

Информация об авторах

Николай П. Артюхин, студент, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 105005, Россия, Москва, 2-я Бауманская ул., д. 5; arperinn@gmail.com

Марина Ю. Барышникова, кандидат педагогических наук, доцент, Московский государственный технический университет им. Н.Э. Баумана, Москва, Россия; 105005, Россия, Москва, 2-я Бауманская ул., д. 5; baryshnikovam@bmstu.ru

Information about the authors

Nikolai P. Artyukhin, student, Bauman Moscow State Technical University, Moscow, Russia; bld. 5, 2nd Baumanskaya Str., Moscow, 105005, Russia; arperinn@gmail.com

Marina Yu. Baryshnikova, Cand. Of Sci. (Pedagogy), associate professor, Bauman Moscow State Technical University, Moscow, Russia; 5, 2nd Bauman-skaya Str., Moscow, 105005, Russia; baryshnikovam@bmstu.ru

Проектирование и разработка автоматизированной системы составления графика консультаций преподавателей вуза

Елена О. Шершнева

*Сибирский государственный автомобильно-дорожный
университет, Омск, Россия, helen_volf@mail.ru*

Артур М. Силкин

*Сибирский государственный автомобильно-дорожный
университет, Омск, Россия, atms-lc@mail.ru*

Аннотация. В статье представлено описание разработанной автоматизированной системы составления графика консультаций преподавателя вуза. Дан анализ российских и зарубежных исследований, посвященных методам и алгоритмам автоматизации построения расписания учебных занятий. График консультаций представлен в виде системы, включающей в себя совокупность взаимосвязанных компонентов (подсистем): преподаватель, дисциплина, аудитория. Применение методов системного анализа позволило выделить основные признаки реализованных компонентов (подсистем) и на их основе построить оптимальное решение задачи. Проведен анализ существующих автоматизированных систем составления расписаний вуза, определены требования к разрабатываемой системе. С использованием технологии функционального моделирования *IDEF0* построена контекстная диаграмма процесса создания графика консультаций преподавателей вуза, а также ее детализация, приведена блок-схема архитектуры автоматизированной системы составления графика консультаций. Продемонстрированы фрагменты кода на языке объектно-ориентированного программирования *C#*, реализованные в ходе разработки системы. Приведен алгоритм последовательности действий пользователя системы. В качестве примера представлено окно разработанной автоматизированной системы составления графика консультаций преподавателя вуза, которое при получении на вход различных видов расписания даст возможность отображать его в удобном для восприятия пользователем системы виде, осуществлять выбор дня, времени и места консультации; выводить график консультаций в необходимом формате. Система применена на кафедре университета для составления графика консультаций, что позволило существенно повысить качество организационного обеспечения и методического сопровождения учебного процесса.

Ключевые слова: информационная система, график консультаций, расписание, диаграмма последовательности действий, объектно-ориентированное программирование

Для цитирования: Шершнева Е.О., Силкин А.М. Проектирование и разработка автоматизированной системы составления графика консультаций преподавателей вуза // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 26–50. DOI: 10.28995/2686-679X-2026-3-26-50

Design and development of an automated system for scheduling consultations of university lecturers

Elena O. Shershneva

*The Siberian State Automobile and Highway University,
Omsk, Russia, helen_volf@mail.ru*

Artur M. Silkin

*The Siberian State Automobile and Highway University,
Omsk, Russia, ams-lc@mail.ru*

Annotation. This article describes the developed automated system for scheduling consultations for university professors. It analyzes Russian and international studies devoted to methods and algorithms for automating the construction of class schedules. The consultation schedule is presented as a system comprising a set of interconnected components (subsystems): professor, discipline, and classroom. The application of systems analysis methods allowed us to identify the key features of the implemented components (subsystems) and, based on them, construct an optimal solution to the problem. An analysis of existing automated systems for scheduling university programs was conducted, and the requirements for the developed system were determined. Using IDEF0 functional modeling technology, a context diagram of the process of creating a consultation schedule for university professors was constructed, as well as its detailing, and a block diagram of the architecture of the automated system for scheduling consultations is provided. Code fragments in the object-oriented programming language C#, implemented during the development of the system, are demonstrated. An algorithm for the sequence of user actions is depicted. As an example, a window is presented for a developed automated system for scheduling university lecturer consultations. When receiving various types of schedules as input, it allows the user to display them in a user-friendly format, select the day, time, and location of the consultation, and output the consultation schedule in

the required format. The system has been used at a university department to create a consultation schedule, significantly improving the quality of organizational support and methodological support for the educational process.

Keywords: information system, consultation schedule, schedule, sequence diagram, object-oriented programming

For citation: Shersheva, E.O. and Silkin, A.M. (2026), "Design and development of an automated system for scheduling consultations of university lecturers", *RSUH/RGGU Bulletin. "Computer Science. Information Security. Mathematics" Series*, no. 3, pp. 26–50, DOI: 10.28995/2686-679X-2026-3-26-50

Введение

Составление оптимального учебного расписания – одна из ключевых задач в организации образовательного процесса в высшем учебном заведении. От эффективности ее решения зависят качество образовательного процесса за счет снижения стресса и утомляемости студентов и преподавателей, рациональное использование научно-педагогического потенциала, продуктивность работы профессорско-преподавательского состава. Создание информационной системы составления оптимального расписания – не просто техническая задача, а стратегический шаг к повышению конкурентоспособности вуза в условиях цифровизации образования.

Проблеме составления оптимального расписания в вузе посвящено большое количество работ, так как она представляет практическую значимость и играет огромную роль в организации учебного процесса вуза [Ненашев 2023].

В общем случае задача составления графика консультаций в вузе является многокритериальной задачей оптимизации: необходимо учитывать большое количество требований и критериев [Хасухаджиев, Сибикина 2016]. На практике часто используют обобщенные критерии, представляемые в виде свертки частных критериев.

Рассмотрим постановку задачи.

1. Пусть X – матрица расписания, где:

$x_{ij} = 1$, если консультация преподавателя назначена на временной слот i в аудитории j ;

$x_{ij} = 0$ – в противном случае;

i – индекс преподавателя ($i = 1, \dots, I$);

j – индекс временного слота (например, день и номер пары: $j = 1, \dots, J$).

Задачу составления графика консультаций преподавателей вуза сформулируем как задачу дискретной оптимизации. Для заданного набора временных интервалов (учебных пар) требуется построить график консультаций преподавателя, максимизирующий целевую функцию в виде линейной свертки частных критериев:

Пусть вектор искомых переменных представлен в следующем виде:

$$X = [x_{ijk}], i = 1, \dots, I, j = 1, \dots, J, k = 1, \dots, K; x_{ijk} = \{0, 1\},$$

где $x_{ijk} = 1$, если преподаватель i проводит консультацию в слоте j в аудитории k ;

$x_{ijk} = 0$ – в противном случае;

i – индекс преподавателя ($i = 1, \dots, I$);

j – индекс временного слота ($j = 1, \dots, J$), где слот – фиксированный интервал времени (например, 1 пара = 90 мин);

k – индекс аудитории ($k = 1, \dots, K$);

Введем дополнительные переменные:

L_i – общее количество консультаций, которые должен проводить преподаватель i ;

S_d – множество доступных слотов в учебный день d ; здесь d – индекс дня недели ($d = 1, \dots, D$);

M_i – максимальное количество консультаций в день для преподавателя i (ограничение нагрузки);

P_{ij} – доступность преподавателя i в слоте j ;

$P_{ij} = 1$, если преподаватель доступен; $P_{ij} = 0$, если недоступен (например, отпуск).

Введем ограничения:

1. Каждый преподаватель должен провести все назначенные консультации:

$$\sum_{j=1}^J \sum_{k=1}^K x_{ijk} = L_i.$$

2. Преподаватель не может быть в двух аудиториях одновременно:

$$\sum_{k=1}^K x_{ijk} \leq 1.$$

3. Преподаватель не может провести консультацию, если он в это время занят:

$$x_{ijk} \leq P_{ij},$$

- Если $P_{ij} = 0$ (недоступен), то $x_{ijk} \leq 0 \Rightarrow x_{ijk} = 0$ для всех k .
- Если $P_{ij} = 1$ (доступен), ограничение не ограничивает выбор (x_{ijk} может быть 0 или 1).

4. Преподаватель не должен проводить больше консультаций в день, чем допустимо:

$$\sum_{j \in S_d} \sum_{k=1}^K x_{ijk} \leq M_i.$$

Целевая функция F имеет следующий вид:

$$F(X) = \alpha \cdot Q_1(X) + \beta \cdot Q_2(X) + \gamma \cdot Q_3(X) - \delta \cdot S(X) \rightarrow \max,$$

где $Q_1(X)$ – минимизация «окон»:

$$Q_1(X) = \frac{1}{I} \sum_{i=1}^I \left(1 - \frac{e_i}{e_{\max,i}} \right),$$

e_i – количество окон;

$e_{\max,i}$ – максимально возможное количество окон;

$Q_2(X)$ – компактность аудиторного использования:

$$Q_2(X) = \frac{1}{I} \sum_{i=1}^I \left(1 - \frac{K_i - 1}{K - 1} \right),$$

K_i – количество аудиторий для преподавателя i ;

$Q_3(X)$ – равномерность нагрузки:

$$Q_3(X) = 1 - \frac{1}{I} \sum_{i=1}^I \sigma_i,$$

σ_i – стандартное отклонение количества консультаций по дням для преподавателя i .

$S(X)$ – штраф за нарушения «мягких» ограничений:

$$S(X) = s_1 \cdot e_1 + s_2 \cdot e_2 + s_3 \cdot e_3,$$

e_1 – количество окон;

e_2 – количество переходов между аудиториями;

e_3 – мера неравномерности.

Коэффициенты:

α, β, γ – веса для позитивных критериев (нормированы: $\alpha + \beta + \gamma = 1$);

δ – вес штрафа (подбирается так, чтобы штрафы доминировали над позитивными критериями при грубых нарушениях).

Итоговая постановка задачи

Найти матрицу $X = (x_{ijk})$, такую что:

1. Выполняются все ограничения.
2. Целевая функция достигает максимума:

$$F(X) = \alpha \cdot Q_1(X) + \beta \cdot Q_2(X) + \gamma \cdot Q_3(X) - \delta \cdot S(X) \rightarrow \max.$$

Целью статьи является разработка автоматизированной информационной системы составления графика консультаций (АССГК) преподавателя вуза. Будем полагать, что в основу построения функционала АССГК положен алгоритм решения задачи дискретной оптимизации.

Ожидаемый график консультаций – это расписание, которое учитывает различные виды расписания (очную, заочную формы обучения), позволяет выбирать свободный день, время и место консультации, а также выводит график в необходимом формате.

Система должна учитывать, что основное расписание используется как нормативное требование и ограничение при составлении плана консультаций.

Некоторые требования к АС:

Учет различных видов расписания. Система должна поддерживать составление расписания для разных форм обучения, уровней образования (бакалавриат, магистратура).

Отображение расписания в удобном для восприятия пользователю виде.

Возможность выбирать свободный день, время и место консультации. Система должна предоставлять пользователю возможность указывать эти параметры, а также подсказывать возможные варианты расстановки занятий.

Вывод графика консультации в необходимом формате. Например, в виде таблицы с датами, временем и местом консультаций,

с возможностью сохранения графика в форматах *Microsoft Word*, *Excel* или *HTML*.

Процесс работы системы может включать следующие этапы:

- ввод первичной информации: основного расписания преподавателей;
- отображение основного расписания. Система автоматически отображает основные занятия в сетке расписания, тем самым позволяя увидеть оптимальные для консультаций дни;
- выбор дня консультации. Система фиксирует дату, время и место консультаций;
- вывод графика консультации. Система должна выводить график в необходимом формате, например, в виде таблицы с датами, временем и местом консультаций, с возможностью сохранения графика.

Система должна автоматически показывать все коллизии и несоблюдения условий и позволять их быстро отрабатывать.

Основная часть

Сегодня на российском рынке программного обеспечения существует достаточное количество продуктов, которые могут быть использованы для составления расписания учебных занятий, но систем, предназначенных для составления графика консультаций в зависимости от основного расписания, нет [Акрамов 2023]. Таким образом, тематика автоматизации процесса составления графика консультаций преподавателей является актуальной в системе управления учебным процессом вуза.

В большинстве вузов Российской Федерации график консультаций формируется в ручном режиме. Процесс ручного ввода данных является трудоемким, так как требует анализа нескольких видов расписания, ручной фиксации даты и времени консультации, передачи сотруднику для дальнейшего переноса в единый файл в электронном виде [Ткачев, Лаушкин, Саяпин 2011]. Выход из сложившегося положения заключается в том, чтобы создать информационную систему, в которой можно будет в зависимости от исходных файлов расписания выбирать дату и время расписания, получать на выходе файл в электронном виде с графиком консультаций в формате *.doc* или *.json*. Для разработки такой системы необходимо на первом этапе продумать и реализовать алгоритм работы системы, отображающий поэтапный процесс реализации поставленной задачи.

Были проведены аналитические исследования ряда российских и зарубежных источников, посвященных методам и алгоритмам автоматизации построения расписания учебных занятий

[Ульянова 2023]. В анализируемых источниках расписание рассматривается как часть образовательного процесса, обладающая признаками системы, объединяющей в себе различные объекты учебного процесса [Varshney, Gupta, Mapari, Arora, Sathyamoorthy 2024]. Исходя из этого, можно сделать вывод о том, что график консультаций является системой, включающей в себя совокупность таких объектов как преподаватель, дисциплина, аудитория. Применение методов системного анализа позволит выделить основные признаки реализованных систем и на их основе построить оптимальное решение задачи. Задача по составлению графика консультаций представляет собой процесс распределения некоторого конечного набора событий во времени в условиях ресурсных и других ограничений [Гвоздев, Емельянов, Пак 2007]. Пусть требуется составить график консультаций преподавателя с учетом основного расписания различных форм и видов обучения, то есть необходимо достаточным образом распределить имеющиеся ресурсы в соответствии с запросами и выделить время для проведения консультаций.

Для составления контекстной диаграммы процесса в нотации *IDEFO* [Григорьев, Григорьева 2024] были выделены следующие элементы:

- Входы:
файлы расписания (*.xls*, *.xlsx*, *.txt*) – файлы с расписанием преподавателя в формате *.xls*, *.xlsx* или *.txt*;
формат отображения расписания – один из предложенных форматов отображения расписания (недельный или исходный);
параметры экспорта – определяет параметры, связанные с сохранением данных.
- Выходы:
файл расписания/консультаций – файл с расписанием с добавленными в него консультациями в выбранном формате. Расписание представлено в виде таблицы.
- Механизмы:
пользователь – пользователь системы;
алгоритм импорта/визуализации данных – алгоритм, по которому система импортирует или выводит данные.
- Управление:
варианты отображения расписания – варианты, которые можно выбрать для отображения расписания (исходный или недельный);
структура файлов – структура файлов, которая может считываться программой;
нормативные требования и ограничения – рекомендации и предпочтения преподавателя.

С учетом вышеописанных элементов составлена контекстная диаграмма, представленная на рис. 1.

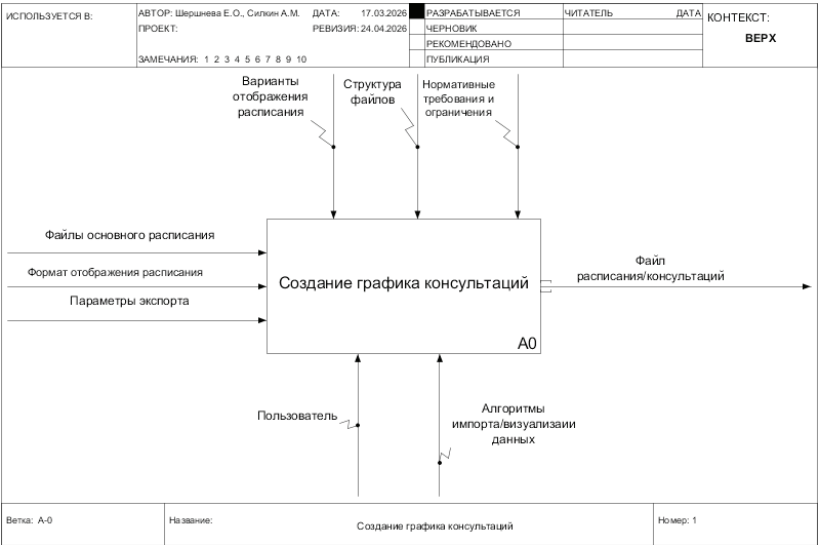


Рис. 1. Контекстная диаграмма процесса создания графика консультаций преподавателей вуза

На следующем шаге были выделены подпроцессы и их элементы:

1. Настройка системы.

- Входы:
формат отображения расписания – один из предложенных форматов отображения расписания (недельный или исходный).
- Выходы:
установленный путь для экспорта файлов – установленный в системе вариант пути для экспорта файлов (экспорт в указанную заранее директорию или выбор директории при экспорте).
- Управление:
варианты отображения расписания – варианты, которые можно выбрать для отображения расписания (исходный или недельный);
нормативные требования и ограничения – рекомендации и предпочтения преподавателя.

- Механизмы:
пользователь – пользователь системы.

2. Импорт расписания.

- Входы:
файлы расписания (*.xls*, *.xlsx*, *.txt*) – файлы с расписанием преподавателя в формате *.xls*, *.xlsx* или *.txt*.
- Выходы:
импортированное расписание – импортированное в систему расписание.
- Управление:
структура файлов – структура файлов, которая может считываться программой.
- Механизмы:
пользователь – пользователь системы;
алгоритм импорта данных – алгоритм, по которому система импортирует данные из файлов.

3. Вывод расписания на экран.

- Входы:
импортированное расписание – импортированное в систему расписание.
- Выходы:
расписание на экране – расписание, выведенное на экран пользователя.
- Управление:
установленный формат отображения расписания – установленный в системе формат отображения расписания на экране пользователя (недельный или исходный).
- Механизмы:
алгоритм визуализации данных – алгоритм, по которому система выводит данные расписания и консультаций на экран.

4. Добавление консультаций.

- Входы:
расписание на экране – расписание, выведенное на экран пользователя;
информация о консультациях – время и аудитория проведения консультации.
- Выходы:
расписание с консультациями – расписание, в которое добавлены консультации.
- Управление:
количество консультаций – количество консультаций, которые необходимо добавить в расписание.

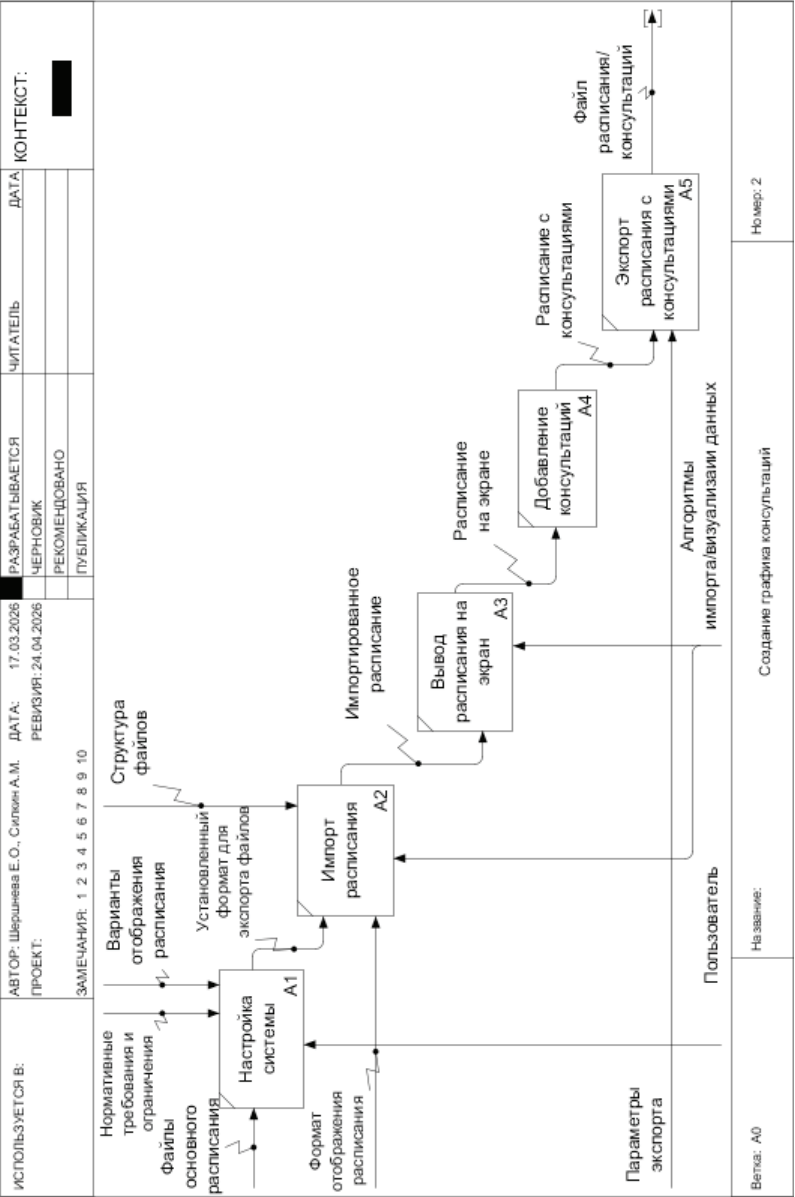


Рис. 2. Декомпозиция процесса создания графика консультаций преподавателей вуза

- Механизмы:
пользователь – пользователь системы.
- 5. Экспорт расписания с консультациями.
 - Входы:
расписание с консультациями – расписание, в которое добавлены консультации;
параметры экспорта – определяет параметры, связанные с сохранением данных.
 - Выходы:
файл с расписанием/консультациями (.doc) – файл с расписанием с добавленными в него консультациями в выбранном формате. Расписание представлено в виде таблицы.
 - Управление:
установленный путь для экспорта файлов – установленный в системе вариант пути для экспорта файлов (экспорт в указанную заранее директорию или выбор директории при экспорте).
 - Механизмы:
пользователь – пользователь системы.

После выделения элементов всех подпроцессов, построена декомпозиция процесса создания расписания, представленная на рис. 2.

Критерии качества графика консультаций преподавателей учитываются и вводятся на уровне внутренних нормативных документов учебного заведения.

На рис. 3 представлена блок-схема архитектуры АССГК.

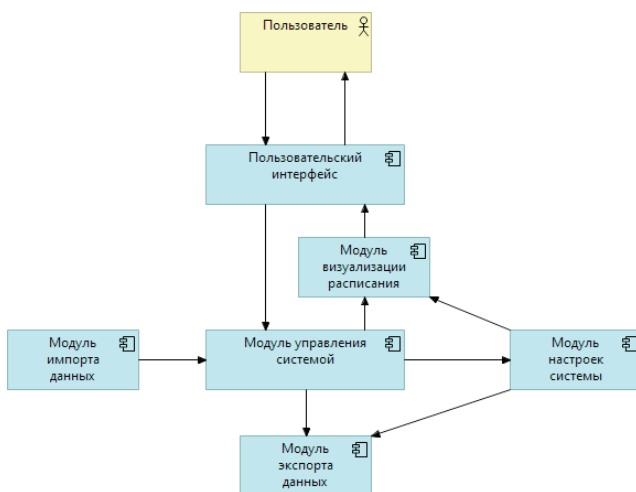


Рис. 3. Блок-схема АССГК

На блок-схеме архитектуры АССГК обозначены блоки:

- пользователь – пользователь системы;
- пользовательский интерфейс – интерфейс системы, с которым взаимодействует пользователь;
- модуль визуализации расписания – модуль, который генерирует структуру расписания (блоки дней, пар и консультаций);
- модуль управления системой – модуль, который управляет системой и хранит всю информацию (расписание и консультации);
- модуль импорта данных – модуль, который импортирует данные из файлов *.xls*, *.xlsx*, *.txt*;
- модуль настроек системы – модуль, который хранит и заменяет настройки системы;
- модуль экспорта данных – модуль, который экспортирует данные из модуля управления системой в файлы *.doc* и *.json*.

При разработке АССГК реализуются следующие методы:

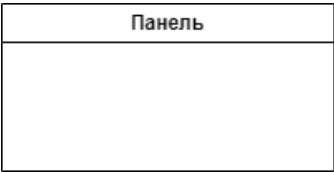
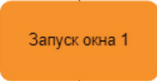
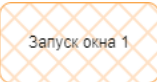
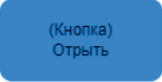
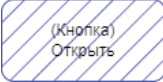
- чтение файла с расширением *.xls*;
- чтение файла в формате *.txt*;
- вывод данных на экран;
- добавление консультаций в расписание;
- сохранение консультаций с расширением *.doc*;
- сохранение консультаций с расширением *.json*;
- сохранение расписания с расширением *.doc*.

При реализации алгоритма, основанного на принципах имитационного моделирования, особое внимание уделялось методам, описываемым в дальнейшем программно [Масляев 2022]. Так как разрабатываемая информационная система предполагает наличие нескольких окон, отвечающих за то или иное действие, была проведена разбивка алгоритма на несколько частей, каждая из которых отвечает за последовательность действий, выполняемых пользователем системы.

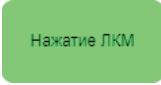
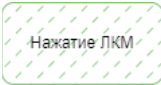
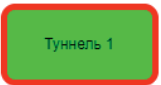
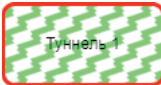
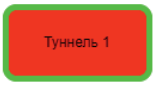
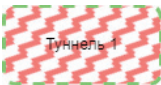
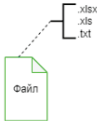
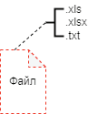
Для описания алгоритма действий пользователя и программы [Song, Liu, Tang, Peng, Chen 2018] была предложена диаграмма, которая позволяет, благодаря вновь введенным обозначениям, определять окна, панели, элементы управления, действия программы и пользователя, сообщения и события. Элементы, представленные в диаграмме, описываются следующими предложенными фигурами (табл. 1).

Таблица 1

Предложенные обозначения элементов,
используемых при описании алгоритма действий
пользователя и программы

Название	Символ		Описание
Точка запуска		или 	С этой точки начинается весь процесс.
Точка завершения работы		или 	Эта точка завершает весь процесс.
Окно			При создании указывается название и номер. В окне размещаются панели.
Панель			В панели размещается основная часть диаграммы.
Переход между окнами			Используется для указания перемещения внимания на другое окно. При создании указывается номер окна для перехода.
Действие программы		или 	Действие, которое в процессе выполняет программа. При создании указывается название действия.
Элемент управления		или 	При создании сверху в круглых скобках указывается тип (кнопка, чекбокс, поле, радио и т. д.), снизу указывается название элемента.

Продолжение табл. 1

Название	Символ	Описание
Действие пользователя	 или 	Действие, которое пользователь выполняет в программе (пример ввода текста).
Туннель (вход)	 или 	Используется для активации параллельного действия программы в другом окне. При создании указывается номер туннеля. Используется в связке с Туннель (выход).
Туннель (выход)	 или 	Туннель (выход) используется для активации параллельного действия программы в другом окне. При создании указывается номер туннеля. Используется в связке с Туннель (вход). После Туннель (выход) указывается действие программы.
Входной файл	 или	Файл, который поступает в программу. При создании указывается название файла, а в сноске указывается расширение (несколько расширений).
Выходной файл	 или 	Файл, который поступает из программы. При создании указывается название файла, а в сноске указывается расширение (несколько расширений).

Окончание табл. 1

Название	Символ	Описание
Поступление данных из базы данных		Данные из базы данных, поступающие в систему. При создании указывается название базы данных, а в сноске таблица или данные.
Выгрузка данных в базу данных		Данные из системы, поступающие в базу данных. При создании указывается название базы данных, а в сноске таблица или данные.
Условие		Условие используется для ветвления процесса. При создании указывается текст условия.
Закрытие окна		Используется, когда работа с окном завершена и его необходимо закрыть.

При запуске программы открывается главное окно, на котором располагается панель управления. Алгоритм работы стартового окна АССГК представлен на рис. 4.

Ниже описаны диаграммы алгоритмов действия пользователя в остальных окнах информационной системы, а именно в окне открытия файла, настройки, выбора варианта сохранения файла и непосредственно самого окна сохранения файла (рис. 5).

В настоящее время в вузе используется информационная система для составления основного расписания преподавателей вуза. После составления файлы расписания каждого преподавателя рассылаются по кафедрам, причем файлов может быть несколько, так как расписание очной формы и заочной формы выгружается отдельно.

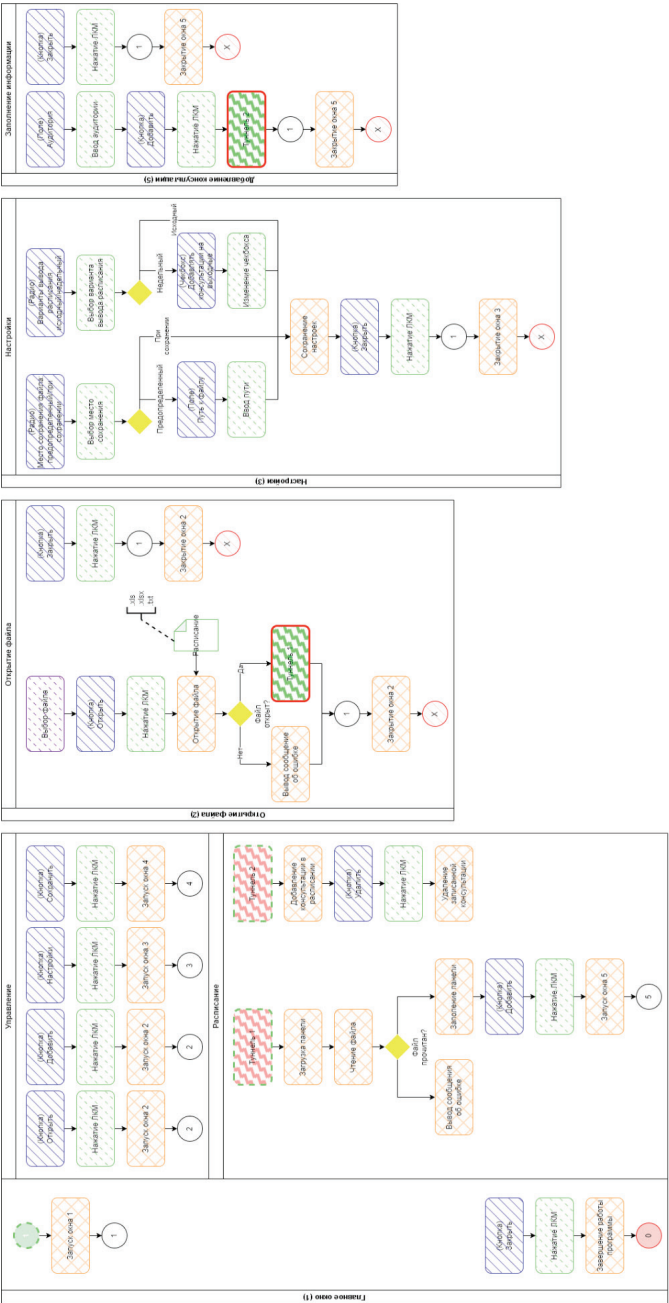


Рис. 5. Алгоритмы работы окон АССГК

Реализуемая система на вход принимает все файлы расписания преподавателя и отображает их в виде, представленном на рис. 6, где зеленым цветом подсвечиваются лекционные занятия, оранжевым – практики, синим – экзамены. Дополнительные цвета, используемые в системе и не представленные на рисунке для лабораторных работ фиолетовый, бирюзовый для зачетов. Желтым отображены уже добавленные посредством окна (рис. 6) консультации. На рис. 7 представлен алгоритм сохранения графика в удобном для пользователя формате.

29.01 пнд	30.01 втр	31.01 сре	01.02 чет	02.02 птн	03.02 сбт	04.02 вск
08:20-09:50	08:20-09:50	08:20-09:50	08:20-09:50	08:20-09:50	08:20-09:50	08:20-09:50
Действие...	Действие...	Действие...	Действие...	Действие...	Действие...	Действие...
10:00-11:30	10:00-11:30	10:00-11:30	10:00-11:30	10:00-11:30	10:00-11:30	10:00-11:30
Действие...	Консультации ФГОУ преподаватель	Действие...	Действие...	Действие...	Действие...	Действие...
11:40-13:10	11:40-13:10	11:40-13:10	11:40-13:10	11:40-13:10	11:40-13:10	11:40-13:10
Действие...	Действие...	Действие...	Действие...	Действие...	Действие...	Действие...
13:40-15:10	13:40-15:10	13:40-15:10	13:40-15:10	13:40-15:10	13:40-15:10	13:40-15:10
Действие...	Действие...	Консультации ФГОУ преподаватель 2-408	Действие...	Действие...	Действие...	Действие...
15:25-16:55	15:25-16:55	15:25-16:55	15:25-16:55	15:25-16:55	15:25-16:55	15:25-16:55
Действие...	Действие...	Действие...	Действие...	Действие...	Действие...	Действие...
17:00-18:25	17:00-18:25	17:00-18:25	17:00-18:25	17:00-18:25	17:00-18:25	17:00-18:25
Л. Препод1 ФГОУ преподаватель Лектор1 Препод1	Л. Препод2 ФГОУ преподаватель Лектор2 Препод2	Действие...	Действие...	Л. Препод3 ФГОУ преподаватель Лектор3 Препод3	Действие...	Действие...
18:40-20:10	18:40-20:10	18:40-20:10	18:40-20:10	18:40-20:10	18:40-20:10	18:40-20:10
Действие...	Действие...	Действие...	Действие...	Действие...	Действие...	Действие...
20:20-21:50	20:20-21:50	20:20-21:50	20:20-21:50	20:20-21:50	20:20-21:50	20:20-21:50
Действие...	Действие...	Действие...	Действие...	Действие...	Действие...	Действие...
05.02 пнд	06.02 пнд	07.02 сре	08.02 чет	09.02 птн	10.02 сбт	11.02 вск

Рис. 6. Главное окно

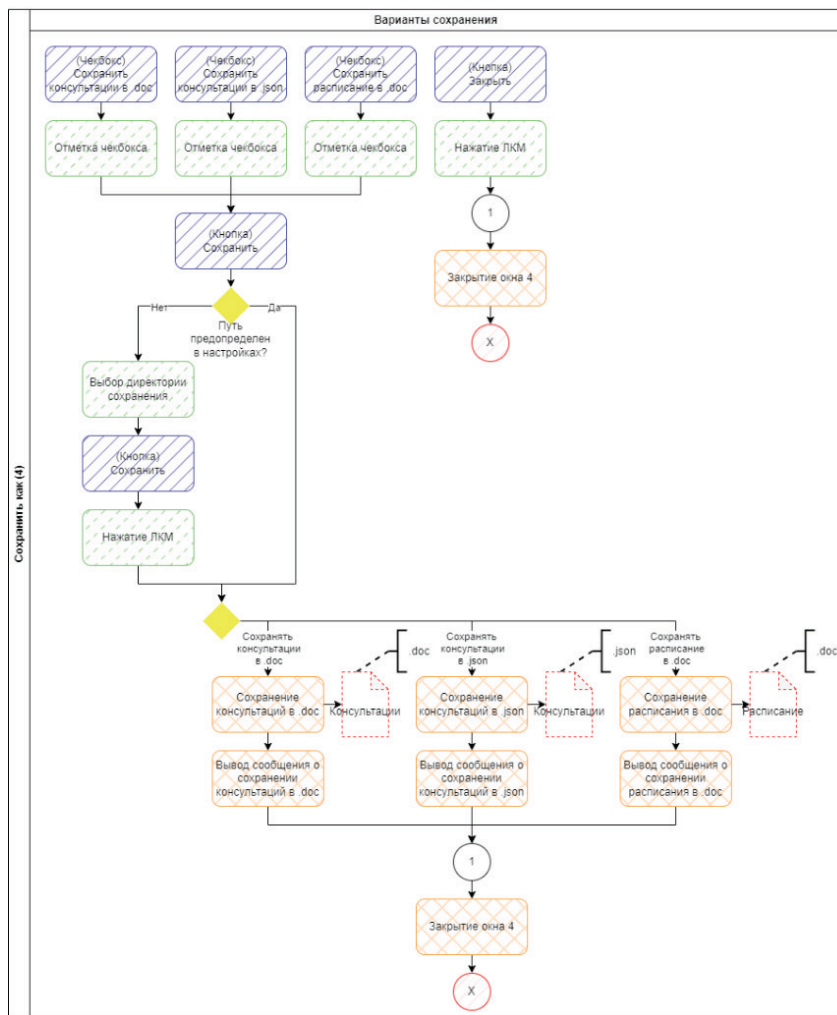
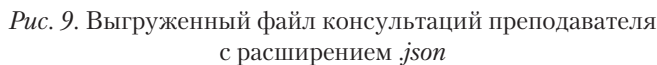


Рис. 7. Алгоритм сохранения графика консультаций в удобном для пользователя формате

Как описывалось выше, система позволяет выгружать консультации в формате *.doc* и *.json*. В качестве примера на рис. 8 и 9 представлены визуализации соответствующих выгрузок.

Рис. 8. Выгруженный файл консультаций преподавателя
с расширением .doc

Рис. 8. Выгруженный файл консультаций преподавателя
с расширением .doc



ISSN 2686-679X • RSUH/RGGU BULLETIN. "Information Science. Information Security.
Mathematics" Series, 2026, no. 3

Заключение

В работе проведен анализ существующих автоматизированных систем составления расписаний вуза для определения функциональных требований к разрабатываемой АССГК преподавателя. С использованием технологии функционального моделирования *IDEFO* построена контекстная диаграмма процесса составления графика консультаций, а также проведена ее декомпозиция. Разработанная АССГК преподавателя вуза при получении на вход различных видов расписания дает возможность:

- отображать его в удобном для восприятия пользователем системы виде;
- реализовывать возможность выбора дня, времени и места консультации;
- выводить график консультации в необходимом формате.

Приведена блок-схема архитектуры АССГК, включающая в себя такие элементы, как пользователь системы, пользовательский интерфейс, модуль визуализации расписания, модуль управления системой, модуль импорта данных, модуль настроек системы, модуль экспорта данных.

Описан алгоритм действия системы и пользователя, включающий отображение окон, панелей, элементов управления, действий программы и пользователя, сообщений и событий. С использованием объектно-ориентированного языка программирования *C#* была произведена реализация основных функций системы.

Внедрение разработанной автоматизированной системы на кафедре «Цифровые технологии» ФГБОУ «СибАДИ» стало важным шагом в цифровизации административных процессов. Ключевым достижением является автоматизация составления графика консультаций с учетом сложного расписания очной и заочной форм обучения студентов.

Основные результаты внедрения:

- оптимизация временных ресурсов: полностью устранены трудоемкие ручные операции по формированию графика;
- повышение точности: система исключает человеческий фактор и автоматически выявляет потенциальные пересечения в расписании, обеспечивая корректность итогового графика;
- гибкость и оперативность: появилась возможность оперативно вносить изменения и генерировать график в различных форматах (электронный, печатный, для публикации на сайте), что улучшило информирование студентов;

- унификация процесса: создан единый цифровой инструмент для планирования консультаций, доступный всем ответственным сотрудникам кафедры.

Итогом стало повышение общей эффективности работы кафедры и снижение административной нагрузки.

Литература

- Акрамов 2023 – *Акрамов Д.Б.* Анализ состояния программных средств для автоматического составления учебных расписаний // Научному прогрессу – творчество молодых. 2023. № 2. С. 270–272. EDN IEBJPB.
- Гвоздев, Емельянов, Пак 2007 – *Гвоздев Б.А., Емельянов П.Г., Пак Е.В.* Составление расписаний в учебных заведениях: требования, проблемы, подходы к решению // Новые технологии в образовании. 2007. № 3. С. 56–76. EDN RXPFMH.
- Григорьев, Григорьева 2024 – *Григорьев М.В., Григорьева И.И.* Проектирование информационных систем: учебное пособие для вузов. М.: Юрайт, 2024. 278 с.
- Масляев 2022 – *Масляев Д.А.* Современное состояние задачи автоматизации составления оптимального учебного расписания в вузе // Вестник Сыктывкарского университета. Серия 1: Математика. Механика. Информатика. 2022. № 1 (42). С. 23–40. DOI: 10.34130/1992-2752_2022_1_23. EDN KGUMLX.
- Ненашев 2023 – *Ненашев С.Д.* Решение проблем составления расписания учебных занятий в высшем учебном учреждении за счет разработки автоматизированной системы // Вестник молодых ученых Санкт-Петербургского государственного университета технологии и дизайна. 2023. № 1. С. 107–112. EDN HGCVBVJ.
- Ткачев, Лаушкин, Саяпин 2011 – *Ткачев Д.В., Лаушкин А.Ф., Саяпин А.В.* Анализ существующих подходов к автоматизации составления расписания учебных занятий в образовательных учреждениях // Решетневские чтения. 2011. Т. 2. С. 590–591. EDN UNKSMV.
- Ульянова 2023 – *Ульянова Е.А.* Методы и алгоритмы для составления расписания учебного заведения // Global and Regional Research. 2023. Т. 5, № 2. С. 70–78. EDN EYJNTT.
- Хасухаджиев, Сибикина 2016 – *Хасухаджиев А.С., Сибикина И.В.* Обобщенный алгоритм составления расписания в вузе с учетом новых требований федеральных государственных образовательных стандартов // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2016. № 3. С. 78–86. EDN WEZGGL.
- Song, Liu, Tang, Peng, Chen 2018 – *Song T., Liu S., Tang X., Peng X., Chen M.* An iterated search algorithm for the University course timetabling problem // Applied Soft Computing 2018. Vol. 68. P. 597–608.

Varshney, Gupta, Mapari, Arora, Sathyamoorthy 2024 – Varshney A., Gupta S., Mapari S., Arora S., Sathyamoorthy M. Optimizing class scheduling: a genetic algorithm approach for smart scheduling and builder system // 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies. Pune: 2024. P. 1–6. DOI: 10.1109/TQCEBT59414.2024.10545033.

References

- Akramov, D.B. (2023), “Analysis of the state of software for automatic compilation of educational schedules”, *Scientific progress – creativity of the young*, no. 2. pp. 270–272.
- Grigoriev, M.V. and Grigorieva, I.I. (2024), *Proyektirovaniye informatsionnykh sistem: uchebnoye posobiye dlya vuzov* [Design of information systems: a textbook for universities], Yurait, Moscow, Russia.
- Gvozdev, B.A., Emelyanov, P.G. and Pak, E.V. (2007), “Scheduling in educational institutions: requirements, problems, approaches to solution”, *New technologies in education*, no. 3, pp. 56–76.
- Khasukhadzhiev, A.S. and Sibikina, I.V. (2016), “General algorithm for scheduling at a university taking into account the new requirements of federal state standards”, *Vestnik ASTU Series “Management, computer technology and information science”*, no. 3, pp. 78–86.
- Maslyayev, D.A. (2022), “Current state of the problem of automating the preparation of an optimal educational schedule at a university”, *Bulletin of Syktyvkar University. Ser. 1: Mathematics. Mechanics. Informatics*. no. 1 (42), pp. 23–40. DOI: 10.34130/1992-2752_2022_1_23.
- Nenashev, S.D. (2023), “Solving the problems of scheduling classes in a higher educational institution through the development of an automated system”, *Bulletin of young scientists of the St. Petersburg State University of Technology and Design*, no. 1. pp. 107–112.
- Song, T., Liu, S., Tang, X., Peng, X., Chen, M. (2018), “An iterated search algorithm for the University Course Timetabling Problem”, *Applied Soft Computing*, vol. 68, pp. 597–608.
- Tkachev, D.V., Laushkin, A.F. and Sayapin, A.V. (2011), “Analysis of existing approaches to automating the scheduling of classes in educational institutions”, *Reshetnevskie readings*, vol. 2, pp. 590–591.
- Ulyanova, E.A. (2023), “Methods and algorithms for creating an educational institution schedule”, *Global and Regional Research*, 2023. vol. 5, no. 2. pp. 70–78.
- Varshney, A., Gupta, S., Mapari, S., Arora, S., Sathyamoorthy, M. (2024), “Optimizing Class Scheduling: A Genetic Algorithm Approach for Smart Scheduling and Builder System”, *International Conference on Trends in Quantum Computing and Emerging Business Technologies*, Pune, India, 2024, pp. 1–6, DOI: 10.1109/TQCEBT59414.2024.10545033.

Информация об авторах

Елена О. Шершнева, кандидат технических наук, Сибирский государственный автомобильно-дорожный университет, Омск, Россия; 644050, Россия, Омск, пр. Мира, д. 5; helen_volf@mail.ru

Артур М. Силкин, студент, Сибирский государственный автомобильно-дорожный университет, Омск, Россия; 644050, Россия, Омск, пр. Мира, д. 5; ams-lc@mail.ru

Information about the authors

Elena O. Shershneva, Cand. of Sci. (Mechanical Engineering), The Siberian State Automobile and Highway University, Omsk, Russia; 5, Mira Av., Omsk, 644050, Russia; helen_volf@mail.ru

Artur M. Silkin, student, The Siberian State Automobile and Highway University, Omsk, Russia; 5, Mira Av., Omsk, 644050, Russia; ams-lc@mail.ru

Сравнительный анализ
алгоритмов обнаружения коллизий
в задачах имитационного моделирования

Андрей П. Титов

МИРЭА – Российский технологический университет,

Москва, Россия;

Институт кибербезопасности и цифровых технологий,

Москва, Россия, titov_and@mail.ru

Наталия В. Гришина

Московский государственный лингвистический университет,

Москва, Россия, grnat@rambler.ru

Дарья Н. Титова

Образовательный центр «Протон», Москва, Россия,

daratitovaa@gmail.com

Аннотация. Обнаружение коллизий является одной из ключевых проблем вычислительной геометрии и критическим фактором, определяющим производительность систем имитационного моделирования, компьютерной графики и разработки компьютерных игр. При увеличении количества взаимодействующих объектов в симуляции вычислительная сложность алгоритмов определения их пересечений может расти экспоненциально, что приводит к снижению частоты кадров и невозможности моделирования сложных систем в реальном времени. В связи с этим выбор оптимального алгоритма обнаружения столкновений становится актуальной задачей при проектировании масштабируемых программных комплексов, работающих с динамическими средами.

Целью статьи является сравнительный анализ вычислительной эффективности алгоритмов определения коллизий применительно к задачам имитационного моделирования с большим количеством агентов. В качестве исследуемых объектов выступают агенты, имеющие форму окружности, что является распространенной моделью для широкого класса задач от моделирования социальных систем до физических симуляторов. В основу обнаружения столкновения положена процедура проверки соотношения суммы радиусов двух окружностей и расстояния между их центрами, вычисляемого с использованием теоремы Пифагора. Рассмат-

риваются два принципиально различных подхода к организации перебора пар объектов. Практическая значимость исследования заключается в возможности использования полученных количественных оценок при проектировании архитектуры программного обеспечения для широкого спектра прикладных задач: от разработки игровых движков и тренажеров до создания научных симуляторов сложных систем, включая моделирование физических процессов, социальных взаимодействий и роевого интеллекта.

Ключевые слова: имитационное моделирование, обнаружение коллизий, алгоритмы, *Brute-force*, *Sweep and prune*, вычислительная геометрия, оптимизация, производительность, *FPS*, временные затраты, компьютерная графика, агентное моделирование

Для цитирования: Титов А.П., Гришина Н.В., Титова Д.Н. Сравнительный анализ алгоритмов обнаружения коллизий в задачах имитационного моделирования // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 51–71. DOI: 10.28995/2686-679X-2026-3-51-71

Comparative analysis of collision detection algorithms in simulation tasks

Andrei P. Titov

*MIREA – Russian Technological University, Moscow, Russia;
Institute of Cybersecurity and Digital Technologies, Moscow, Russia,
titov_and@mail.ru*

Natalia V. Grishina

*Moscow State Linguistic University, Moscow, Russia,
grnat@rambler.ru*

Daria N. Titova

*State Budgetary Educational Institution of the city of Moscow
“Proton Educational Center”, Moscow, Russia,
daratitovaa@gmail.com*

Abstract. Collision detection is a key problem in computational geometry and a critical factor determining the performance of simulation systems, computer graphics, and game development. As the number of interacting objects in a simulation increases, the computational complexity of intersection detection algorithms can grow exponentially, leading to a decrease in frame rate and the

inability to simulate complex systems in real time. Therefore, selecting the optimal collision detection algorithm becomes a pressing issue in the design of scalable software systems operating in dynamic environments. The aim of this article is to comparatively analyze the computational efficiency of collision detection algorithms applied to simulation problems with a large number of agents. The objects under study are agents shaped like circles, a common model for a wide range of problems, from social systems modeling to physical simulations. Collision detection is based on a procedure for checking the ratio of the sum of the radii of two circles to the distance between their centers, calculated using the Pythagorean theorem. Two fundamentally different approaches to organizing the enumeration of pairs of objects are considered. The practical significance of the study lies in the possibility of using the obtained quantitative estimates in designing software architectures for a wide range of applied problems: from the development of game engines and simulators to the creation of scientific simulators of complex systems, including the modeling of physical processes, social interactions, and swarm intelligence.

Keywords: simulation modeling, collision detection, algorithms, Brute-force, sweep and prune, computational geometry, optimization, performance, FPS, time costs, computer graphics, agent-based modeling

For citation: Titov, A.P., Grishina, N.V. and Titova, D.N. (2026), "Comparative analysis of collision detection algorithms in simulation tasks", *RSUH/RGGU Bulletin. "Information Science. Information Security. Mathematics" Series*, no. 3, pp. 51–71, DOI: 10.28995/2686-679X-2026-3-51-71

Введение

Обнаружение столкновений между объектами является одной из ключевых задач вычислительной геометрии и имитационного моделирования. В любой системе, где моделируется физическое взаимодействие множества сущностей будь то компьютерная игра, тренажер виртуальной реальности, симулятор физических процессов или система агентного моделирования возникает необходимость достоверно и своевременно определять факт контакта между объектами. От эффективности решения этой задачи напрямую зависит не только корректность симуляции, но и производительность всей системы в целом, что особенно критично при работе в режиме реального времени.

Особую значимость данная проблема приобретает в контексте разработки универсальных платформ для тестирования алгоритмов поведения искусственного интеллекта. Подобные платформы, предназначенные для исследовательских целей, должны обеспе-

чивать возможность проведения экспериментов с различными сценариями и конфигурациями, варьируя количество агентов, их свойства и правила взаимодействия. В таких условиях модуль обнаружения коллизий может стать узким местом, ограничивающим масштабируемость всей системы.

Область применения алгоритмов обнаружения коллизий чрезвычайно широка. В индустрии компьютерных игр они обеспечивают реалистичность взаимодействия персонажей и объектов игрового мира. В научных симуляторах, моделирующих физические процессы, точность и своевременность обнаружения контактов определяет достоверность получаемых результатов. В робототехнике алгоритмы обнаружения коллизий используются для планирования безопасных траекторий движения. Исследования, направленные на повышение эффективности алгоритмов обнаружения столкновений, имеют как теоретическую, так и практическую ценность.

Определение математического критерия факта коллизии между двумя объектами

В рассматриваемой задаче имитационного моделирования каждый агент представляет собой материальную точку, обладающую некоторым радиусом, то есть геометрически может быть представлен в виде окружности на двумерной плоскости.

Выбор круга в качестве базовой геометрической формы агента обусловлен несколькими факторами. Во-первых, круг является изотропной фигурой, то есть его свойства не зависят от ориентации, что существенно упрощает расчеты. Во-вторых, математический аппарат для работы с кругом хорошо разработан и не требует значительных вычислительных ресурсов. В-третьих, многие реальные объекты при моделировании могут быть аппроксимированы кругом с достаточной степенью точности – например, при моделировании движения пешеходов, клеток микроорганизмов или элементарных частиц [Бурков 2020].

Базовой характеристикой круга является его площадь S , которая вычисляется по классической формуле:

$$S = \pi r^2, \quad (1)$$

где π – константа пи; r – радиус круга.

Однако для целей обнаружения столкновений площадь круга представляет лишь косвенный интерес. Непосредственное

значение имеют координаты центра и его радиус, поскольку именно эти параметры позволяют однозначно определить положение объекта в пространстве и его геометрические размеры. При моделировании каждый агент в каждый момент времени характеризуется тройкой параметров: координатами центра (x, y) и радиусом r .

Коллизия между двумя агентами возникает в тот момент, когда соответствующие им круги начинают пересекаться или, как минимум, касаться друг друга. С геометрической точки зрения круги пересекаются, если расстояние между их центрами меньше суммы радиусов. Если расстояние между центрами в точности равно сумме радиусов, круги касаются, что также должно интерпретироваться как коллизия, поскольку в симуляции агенты не могут занимать одно и то же пространство даже касательным образом [Собинов, Коробицын 2010].

Для определения факта столкновения между двумя агентами с радиусами R_1 и R_2 и центрами в точках (x_1, y_1) и (x_2, y_2) необходимо вычислить евклидово расстояние между их центрами. Это расстояние может быть найдено с использованием теоремы Пифагора, которая устанавливает соотношение между сторонами прямоугольного треугольника.

В координатной форме расстояние c между центрами двух кругов (рис. 1) выражается формулой:

$$c^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2. \quad (2)$$

Выражение (2) представляет собой не что иное, как теорему Пифагора, записанную в декартовых координатах.

Итоговое условие обнаружения коллизии между двумя агентами может быть сформулировано следующим образом: окружности пересекаются или касаются, если квадрат расстояния между их центрами меньше либо равен квадрату суммы их радиусов [Титов 2024]. Математически это условие записывается в виде неравенства:

$$R_1^2 + R_2^2 \leq c^2, \quad (3)$$

где R_1 и R_2 – радиусы двух окружностей; c – расстояние между окружностями.

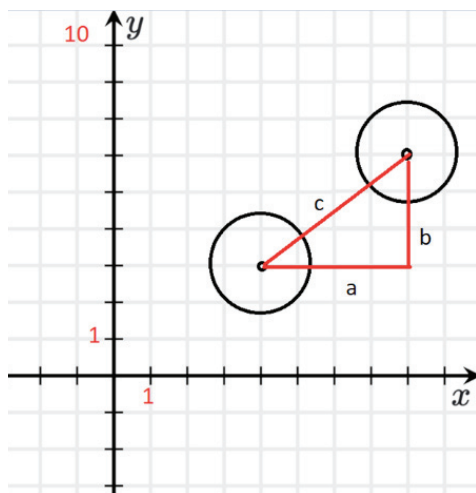


Рис. 1. Расстояние между окружностями

Организация перебора пар объектов для проверки

Однако само по себе наличие математического критерия столкновения не решает проблему обнаружения коллизий в полном объеме. Критически важным является вопрос организации перебора пар объектов для проверки. При наличии n агентов на сцене общее количество возможных пар составляет $n(n - 1)/2$. Наивный подход, предполагающий проверку всех возможных пар, приводит к квадратичной вычислительной сложности, что при увеличении n быстро становится неприемлемым. Так, при $n = 100$ необходимо проверить 4 950 пар, при $n = 1000$ около 500 тысяч пар, а при $n = 10\,000$ примерно 50 миллионов пар за один цикл симуляции.

В системах реального времени такой подход неприменим. Возникает необходимость использования алгоритмов, позволяющих сократить количество проверяемых пар за счет отсева заведомо непересекающихся объектов на основе некоторых эвристик. Основная идея таких алгоритмов заключается в том, чтобы на первом этапе быстро отбросить объекты, которые находятся далеко друг от друга и заведомо не могут столкнуться, и лишь затем выполнять точную геометрическую проверку для оставшихся кандидатов [Эриксон 2022].

Задача обнаружения коллизий распадается на две подзадачи: формирование множества потенциально пересекающихся пар объектов и геометрическая проверка каждой пары из этого множества с использованием критерия. От того, насколько эффективно решается первая подзадача, зависит общая производительность системы моделирования.

Наиболее простым и интуитивно понятным подходом к решению задачи обнаружения коллизий является алгоритм полного перебора, известный в англоязычной литературе как *Brute-force*. Данный алгоритм основывается на прямолинейной идее: для того, чтобы определить все столкновения между агентами, необходимо проверить каждую возможную пару объектов на соответствие критерию пересечения. Несмотря на свою очевидную вычислительную неэффективность при работе с большими наборами данных, алгоритм *Brute-force* представляет интерес как базовая точка отсчета для сравнения и как отправная точка для понимания необходимости оптимизаций [Bergen 2003].

Принцип работы наивного алгоритма полного перебора предельно прост. Алгоритм последовательно перебирает все агенты, присутствующие на сцене, и для каждого из них выполняет проверку со всеми остальными агентами. Если рассматривать множество из n агентов, то общее количество выполняемых проверок составляет n^2 , однако с учетом того, что проверка пары (i, j) эквивалентна проверке пары (j, i) , а проверка агента самим собой лишена смысла, фактическое количество уникальных проверок составляет $n(n - 1)/2$. Даже с учетом этой симметрии вычислительная сложность алгоритма остается квадратичной.

Квадратичная сложность означает, что при увеличении количества агентов вдвое время выполнения алгоритма возрастает примерно в четыре раза. Закономерность делает применение наивного алгоритма проблематичным уже при умеренных количествах объектов. Так, при наличии на сцене 50 агентов алгоритм выполняет 1 225 проверок за кадр, при 100 агентах – 4 950 проверок, при 200 агентах – 19 900 проверок, а при 800 агентах количество проверок достигает 319 600. Выполнение такого объема геометрических вычислений в режиме реального времени может оказаться непосильной задачей для вычислительной системы.

В рамках алгоритма полного перебора существуют возможности для оптимизации, позволяющие сократить количество выполняемых проверок. В рамках статьи рассмотрены три модификации алгоритма *Brute-force*, каждая из которых устраняет определенный класс избыточных вычислений [Hastie et al. 2009].

Первая модификация (*Brute-force* модификация 1) направлена на исключение проверок, в которых агент проверяется сам с собой. В базовой реализации, если не контролировать индексы перебора, алгоритм может выполнять проверку (i, i) , что заведомо не имеет смысла, поскольку объект не может столкнуться сам с собой. Исключение таких проверок позволяет незначительно сократить количество вычислений ровно на n проверок. При малых значениях n этот выигрыш несущественен, однако при очень больших n экономия становится заметной, хотя и не меняет асимптотической сложности алгоритма.

Вторая модификация (*Brute-force* модификация 2) является более содержательной и направлена на устранение дублирующихся пар. Базовая реализация алгоритма, построенная по принципу двух вложенных циклов, где оба счетчика пробегают по всем агентам, неизбежно будет проверять каждую пару дважды: сначала как (i, j) , а затем как (j, i) . Поскольку столкновение является симметричным событием, повторная проверка не несет новой информации, но требует вычислительных затрат. Исключение дубликатов достигается организацией циклов таким образом, чтобы второй счетчик всегда начинал перебор с элемента, следующего за первым. Это сокращает количество проверок примерно вдвое по сравнению с неоптимизированной версией [Бунина и др. 2024].

Третья модификация (*Brute-force* модификация 3) фактически объединяет достоинства первых двух, реализуя оптимальную схему перебора, при которой каждый агент проверяется только с теми агентами, которые имеют больший индекс, и исключается проверка самого себя. Именно эта модификация является наиболее эффективной реализацией идеи полного перебора и обеспечивает минимально возможное количество проверок для данного подхода ровно $n(n - 1)/2$ операций.

Несмотря на возможность оптимизации, алгоритм полного перебора обладает фундаментальным недостатком, так как не использует никакой информации о пространственном расположении объектов. Проверка выполняется для всех пар без исключения, даже для тех, которые находятся на противоположных концах сцены и заведомо не могут столкнуться в данном кадре. Это приводит к тому, что вычислительные ресурсы расходуются на выполнение огромного количества заведомо бесполезных проверок [Ryan, Masaron 2025].

В системах, где количество агентов невелико (до 50...100), алгоритм *Brute-force* в его оптимизированных модификациях может быть вполне приемлемым решением благодаря простоте реализации и отсутствию накладных расходов на поддержание допол-

нительных структур данных. При масштабировании системы на сотни и тысячи объектов необходимость в более интеллектуальных подходах становится очевидной.

В отличие от наивного подхода, алгоритм *Sweep and prune* (также известный как *Sort and sweep*) представляет собой более интеллектуальный метод обнаружения потенциальных коллизий, основанный на использовании пространственной информации о положении объектов. Данный алгоритм относится к классу методов прореживания (*pruning methods*), основная идея которых заключается в быстром отсеивании заведомо непересекающихся объектов на раннем этапе, чтобы минимизировать количество дорогостоящих геометрических проверок. Алгоритм *Sweep and prune* получил широкое распространение в системах физического моделирования, игровых движках и научных симуляторах благодаря оптимальному соотношению эффективности и сложности реализации.

Базовый принцип работы алгоритма основан на наблюдении, что два объекта могут пересекаться только в том случае, если их проекции на координатные оси перекрываются. Иными словами, для того чтобы окружности A и B пересекались, необходимо, чтобы интервалы, занимаемые этими окружностями на оси X , имели общую часть, и аналогично для оси Y . Если проекции объектов на какую-либо ось не пересекаются, то объекты гарантированно не могут находиться в состоянии коллизии. Это условие является необходимым, но не достаточным даже при перекрытии проекций по обеим осям объекты могут не пересекаться, если они расположены по диагонали друг относительно друга. Однако именно это наблюдение позволяет существенно сократить количество проверок [Waskom 2021].

В классической реализации алгоритм *Sweep and prune* работает с осями координат последовательно. На первом этапе выполняется сортировка всех объектов по одной из осей, например, по оси X . Сортировка может выполняться по координате центра объекта или по координате его левой (минимальной) границы. В случае окружностей с одинаковым радиусом выбор конкретного способа не влияет на результат, поскольку смещение всех объектов на постоянную величину сохраняет порядок следования.

Для наглядного представления работы алгоритма рассмотрим исходное состояние системы, в котором агенты имеют случайную нумерацию, соответствующую порядку их создания.

На рис. 2 показано, что объекты расположены в пространстве безотносительно к их идентификаторам, и отсутствует какая-либо пространственная организация, которая могла бы быть использована для оптимизации.

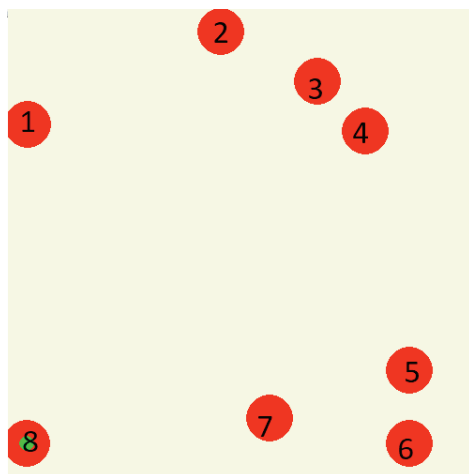


Рис. 2. Случайная нумерация объектов перед запуском алгоритма *Sweep and prune*

После выполнения сортировки по оси X объекты выстраиваются в порядке возрастания координаты их левых границ (или центров).

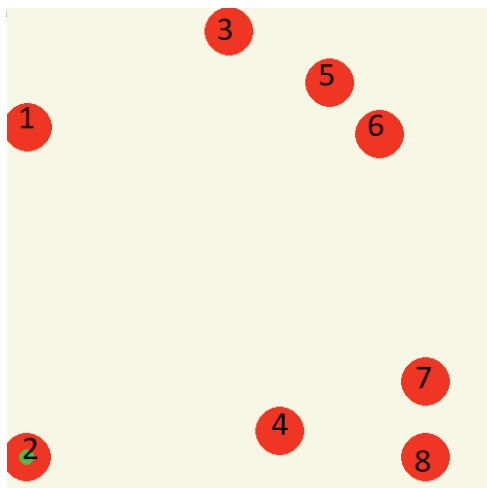


Рис. 3. Нумерация объектов в алгоритме *Sweep and prune* после сортировки

Как видно из рис. 3, после сортировки объекты, расположенные близко друг к другу по оси X , получают близкие номера в отсортированном списке. Это свойство и используется для последующего прореживания. Основной этап алгоритма заключается в проходе по отсортированному списку и выделении множества потенциальных кандидатов на столкновение для каждого объекта. Для фиксированного объекта рассматриваются все последующие объекты в отсортированном списке до тех пор, пока выполняется условие перекрытия проекций на ось X . Условие перекрытия формулируется следующим образом: левая граница последующего объекта должна быть меньше правой границы текущего объекта. Как только встречается объект, левая граница которого находится правее правой границы текущего объекта, дальнейшая проверка для текущего объекта прекращается, поскольку все последующие объекты гарантированно находятся еще правее.

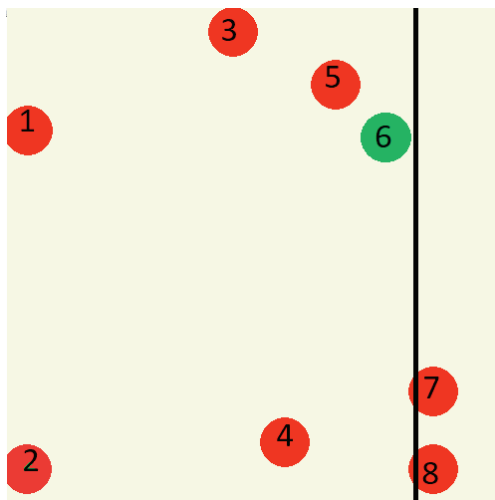


Рис. 4. Проверка потенциальных столкновений
в алгоритме *Sweep and prune*

На рис. 4 проиллюстрирован процесс формирования множества кандидатов. Для выбранного агента (на рисунке это агент номер 6) выделяются только те объекты, которые находятся в пределах его проекции на ось X – в данном случае это агенты 7 и 8. Агент 9 и последующие уже не проверяются, поскольку их левые границы находятся правее правой границы агента 6.

Попадание в множество кандидатов на основе проверки только по оси X не гарантирует реального столкновения. Объекты могут иметь перекрывающиеся проекции на ось X , но при этом находиться далеко друг от друга по оси Y . Именно поэтому сформированное множество называется множеством потенциальных кандидатов: для каждой пары из этого множества в дальнейшем выполняется точная геометрическая проверка с использованием критерия, основанного на теореме Пифагора. Однако количество таких точных проверок оказывается значительно меньше общего числа пар благодаря предварительной фильтрации.

Вычислительная сложность алгоритма *Sweep and prune* определяется двумя основными операциями: сортировкой и проходом с формированием кандидатов. Сортировка выполняется за $O(n \log n)$ операций, где n – количество объектов. Эта операция выполняется один раз за кадр или, в динамических системах, может быть оптимизирована с использованием алгоритмов, учитывающих, что от кадра к кадру положение объектов меняется незначительно. Проход по отсортированному списку и формирование кандидатов выполняется за линейное время $O(n)$ при оптимальных условиях, хотя в худшем случае (когда все объекты выстроены в линию с перекрывающимися проекциями) сложность может возрасти до квадратичной. Однако на практике, благодаря пространственной локализации объектов, алгоритм демонстрирует производительность, близкую к линейной [Wickham 2014].

Преимущества алгоритма *Sweep and prune* перед наивным перебором становятся очевидными при увеличении количества объектов. Вместо проверки всех $n(n-1)/2$ пар алгоритм проверяет только те пары, которые действительно имеют шанс столкнуться. Коэффициент сокращения количества проверок зависит от плотности распределения объектов в пространстве и их размеров. В разреженных сценариях, где объекты распределены равномерно по большой площади, эффективность алгоритма максимальна. В плотных сценариях, где объекты сгруппированы на малой площади, эффективность снижается, но в любом случае алгоритм не выполняет лишних проверок для заведомо удаленных объектов.

К ограничениям алгоритма следует отнести необходимость поддержания отсортированного списка объектов, что требует дополнительных затрат памяти и времени на сортировку. Алгоритм хорошо адаптирован для работы в динамических средах, где объекты перемещаются от кадра к кадру. Существуют оптимизированные версии алгоритма, использующие тот факт, что при малых перемещениях порядок объектов меняется незначительно, и пол-

ная сортировка может быть заменена более лёгкими операциями проталкивания элементов.

Алгоритм *Sweep and prune* может быть распространён на трёхмерный случай, где требуется проверка перекрытия проекций на все три оси. Однако для двумерного случая, рассматриваемого в данной работе, достаточно проверки по одной оси с последующей точной проверкой кандидатов, что и было реализовано в рамках проведенного исследования.

Для получения объективных количественных оценок производительности рассмотренных алгоритмов был проведен вычислительный эксперимент в контролируемых условиях. Основная цель эксперимента заключалась в сравнении эффективности трех модификаций алгоритма *Brute-force* и алгоритма *Sweep and prune* при различном количестве агентов на сцене. Важным требованием к проведению эксперимента являлось обеспечение одинаковых условий для всех тестируемых алгоритмов, чтобы исключить влияние внешних факторов на результаты измерений.

Программная реализация исследуемых алгоритмов

Программная реализация исследуемых алгоритмов выполнена на языке программирования *Python* с использованием библиотеки *Pygame* для графического вывода и организации основного цикла симуляции. Выбор *Python* обусловлен его широким распространением в научной среде, наличием развитых средств для прототипирования и проведения экспериментов, а также возможностью точного измерения временных характеристик работы программного кода. Библиотека *Pygame*, в свою очередь, предоставляет удобные средства для создания двумерной графики и содержит встроенные функции для измерения частоты кадров, что критически важно для целей настоящего исследования.

В качестве модели агентов использовались окружности одинакового радиуса, движущиеся по сцене с постоянной скоростью и меняющие направление при столкновениях. Размер сцены был фиксирован и составлял 1280×720 пикселей. Радиус каждого агента был установлен равным 10 пикселям. Начальные координаты агентов генерировались случайным образом с равномерным распределением по всей площади сцены, начальные направления движения также выбирались случайно. Важно отметить, что для обеспечения воспроизводимости результатов использовалась фиксированная последовательность генерации случайных чисел, что гарантировало одинаковые начальные условия для всех прогонов эксперимента.

Ключевым варьируемым параметром в эксперименте выступало количество одновременно присутствующих на сцене агентов. Были выбраны следующие значения: 50, 100, 200, 400 и 800 агентов. Такой диапазон позволяет проследить динамику изменения производительности алгоритмов от относительно небольших нагрузок, где разница между алгоритмами может быть незначительной, до экстремально высоких нагрузок, где проявляются асимптотические свойства алгоритмов. Дополнительно был проведен контрольный замер при отсутствии агентов (0 агентов) для определения базовой производительности графической подсистемы без нагрузки на алгоритмы обнаружения коллизий.

Для оценки производительности использовались две независимые метрики, позволяющие получить комплексное представление об эффективности алгоритмов.

Частота кадров в секунду *FPS (frames per second)* является общепринятым показателем производительности в системах реального времени и компьютерной графике. *FPS* характеризует, насколько плавно и интерактивно работает симуляция с точки зрения пользователя. Чем выше частота кадров, тем более отзывчивой является система. Измерение *FPS* производилось с использованием встроенных средств библиотеки *Pygame*, усреднение выполнялось на интервале не менее 60 кадров для исключения флуктуаций.

Время выполнения алгоритма обнаружения коллизий в миллисекундах является более объективным показателем, поскольку характеризует непосредственно вычислительные затраты на работу алгоритма, вне зависимости от особенностей графического вывода и других компонентов системы. Измерение времени производилось с использованием модуля *time* языка *Python*: непосредственно перед вызовом алгоритма фиксировалась метка времени, сразу после завершения его работы – вторая метка, разность между ними и составляла искомое значение. Для повышения точности измерение выполнялось многократно с последующим усреднением.

При проведении эксперимента соблюдались следующие условия для обеспечения чистоты измерений:

- все посторонние приложения на компьютере были закрыты, чтобы минимизировать влияние фоновых процессов;
- эксперимент проводился на одном и том же аппаратном обеспечении для всех прогонов;
- для каждого алгоритма и каждого значения количества агентов эксперимент повторялся несколько раз для проверки воспроизводимости результатов;
- перед началом измерений выполнялся прогрев системы для выхода на стационарный режим работы.

В ходе вычислительного эксперимента были получены количественные оценки производительности четырех реализаций алгоритмов обнаружения коллизий: трех модификаций алгоритма *Brute-force* и алгоритма *Sweep and prune*. Результаты измерений по двум метрикам: частоте кадров в секунду (*FPS*) и времени выполнения алгоритма в миллисекундах, были систематизированы и представлены в виде сводных таблиц (табл. 1, табл. 2). Для наглядного восприятия динамики изменения производительности при росте количества агентов по каждой таблице были построены графики.

Частота кадров в секунду является интегральным показателем, отражающим не только скорость работы алгоритма обнаружения коллизий, но и влияние других компонентов системы: отрисовки графики, обработки пользовательского ввода, обновления позиций агентов. Тем не менее при прочих равных условиях именно производительность алгоритма обнаружения коллизий становится определяющим фактором, особенно при больших количествах агентов.

Таблица 1

Производительность 4 алгоритмов в *FPS*

Количество клеток	Тип алгоритма			
	<i>Brute force</i>	<i>Brute force</i> модификация 1	<i>Brute force</i> модификация 2	<i>Sweep and prune</i>
0	60	60	60	60
50	58	58	59	59
100	42	42	51	55
200	17	18	24	29
400	6	6	9.5	13
800	1.8	1.7	3.1	5.3

График (рис. 5) наглядно демонстрирует, что кривые, соответствующие алгоритмам семейства *Brute-force*, имеют крутой спад уже при относительно небольших значениях n . Кривая алгоритма *Sweep and prune* спадает более полого, что подтверждает его лучшую масштабируемость. Важно отметить, что даже при максимальной нагрузке *Sweep and prune* сохраняет работоспособность, тогда как алгоритмы полного перебора фактически перестают выполнять свою функцию.

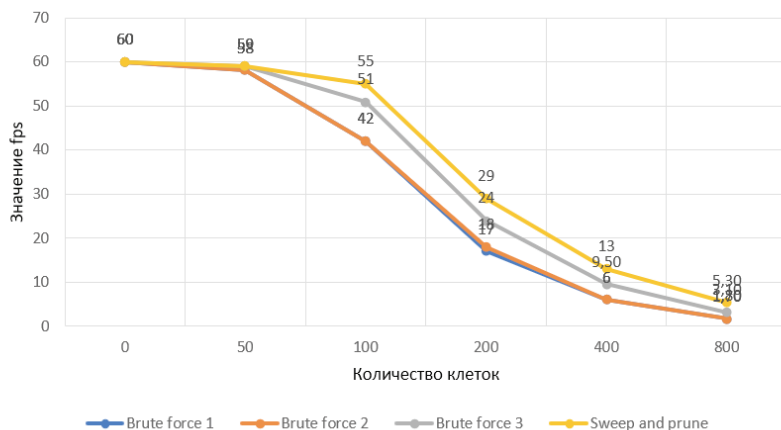


Рис. 5. Производительность алгоритмов в FPS

Вторая метрика – время выполнения алгоритма в миллисекундах, представляет более объективную характеристику, поскольку измеряет непосредственно вычислительные затраты на обнаружение коллизий, исключая влияние графической подсистемы и других компонентов.

Таблица 2

Производительность 4 алгоритмов
в миллисекундах (мс)

Количество клеток	Тип алгоритма			
	<i>Brute force</i> модификация 1	<i>Brute force</i> модификация 2	<i>Brute force</i> модификация 3	<i>Sweep and prune</i>
50	2,02	2,02	1,99	0,98
100	7,01	6,97	3,98	1,97
200	29,88	30,93	15,94	6,02
400	112,70	126,66	57,87	24,92
800	506,17	492,68	230,38	90,72

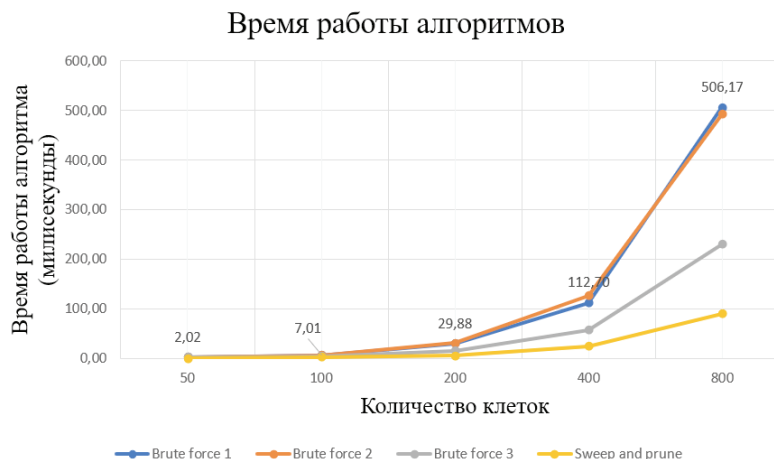


Рис. 6. Производительность алгоритмов в миллисекундах

График временных затрат (рис. 6) наглядно демонстрирует различный характер роста сложности алгоритмов. Кривые *Brute-force* имеют ярко выраженный квадратичный характер: они сильно поднимаются вверх при увеличении n . Кривая *Sweep and prune* растет значительно медленнее, подтверждая его близкую к линейной сложность на практике.

Полученные экспериментальные данные полностью подтверждают теоретические оценки вычислительной сложности алгоритмов. Алгоритм *Brute-force* демонстрирует квадратичную зависимость времени работы от количества агентов: при увеличении n вдвое время работы возрастает примерно в 4 раза. Это хорошо видно при переходе от 200 к 400 агентам: время работы модификации 3 увеличивается с 15,94 мс до 57,87 мс, то есть в 3,63 раза, что близко к теоретической оценке.

Алгоритм *Sweep and prune* показывает существенно более благоприятную зависимость. При переходе от 200 к 400 агентам время работы увеличивается с 6,02 мс до 24,92 мс в 4,14 раза, что на данной выборке даже несколько хуже теоретической оценки. Однако при переходе от 400 к 800 агентам рост составляет с 24,92 мс до 90,72 мс в 3,64 раза, что уже ближе к ожидаемому поведению. В целом на всем диапазоне измерений *Sweep and prune* работает стабильно быстрее.

Особого внимания заслуживает сравнение модификаций алгоритма *Brute-force* между собой. Модификация 1, исключая

проверку агента с самим собой, дает незначительный выигрыш, практически неразличимый на фоне погрешности измерений. Модификация 2, устраняющая дублирующиеся пары, обеспечивает заметное улучшение примерно в 1,5...2 раза по сравнению с базовой версией. Модификация 3, объединяющая оба улучшения, по сути, эквивалентна модификации 2 с точки зрения количества проверок, что и подтверждается близкими значениями времени работы.

Сравнение лучшей модификации *Brute-force* (модификация 3) с алгоритмом *Sweep and prune* показывает устойчивое преимущество последнего на всех тестируемых значениях. При 50 агентах *Sweep and prune* быстрее в 2 раза, при 100 в 2 раза, при 200 в 2,6 раза, при 400 – в 2,3 раза, при 800 в 2,5 раза. В среднем можно говорить о двукратном и более превосходстве алгоритма прореживания над оптимальной реализацией полного перебора.

Важно также отметить, что измеренные значения времени работы позволяют оценить предельные возможности системы при использовании различных алгоритмов. Если принять за приемлемое время работы модуля обнаружения коллизий 10 мс (что оставляет около 6...7 мс на графику и прочие операции при целевых 60 FPS), то алгоритмы *Brute-force* обеспечивают такой уровень производительности лишь до 100...150 агентов. Алгоритм *Sweep and prune* сохраняет приемлемое время работы вплоть до 300...400 агентов, а при компромиссных требованиях (например, 30 FPS) может работать и с 600...800 агентами.

Полученные результаты имеют четкую практическую интерпретацию. При разработке систем имитационного моделирования, где количество одновременно взаимодействующих объектов не превышает 100...150, можно использовать оптимизированные версии алгоритма полного перебора – это позволит получить простое и надежное решение без усложнения архитектуры. Однако при проектировании масштабируемых систем, рассчитанных на работу с сотнями и тысячами агентов, применение алгоритмов типа *Sweep and prune* становится не просто желательным, а необходимым условием обеспечения приемлемой производительности.

Заключение

Экспериментальные данные подтвердили теоретические предположения о вычислительной сложности алгоритмов. Алгоритмы семейства *Brute-force* продемонстрировали квадратичное ухудшение производительности: при 800 агентах время работы базовых модификаций превысило 500 мс, что соответствует 1,8 FPS. Даже

оптимальная модификация, устраняющая дублирующиеся пары, при максимальной нагрузке работала более 230 мс (3,1 FPS).

Алгоритм *Sweep and prune* показал существенно лучшие результаты: при 800 агентах время работы составило 90,7 мс, что обеспечило 5,3 FPS – более чем в 2,5 раза быстрее лучшей модификации *Brute-force*. На всем диапазоне измерений алгоритм прореживания демонстрировал стабильное преимущество, сохраняя приемлемую производительность при нагрузках, где полный перебор уже становится неработоспособным.

Практическая значимость результатов заключается в возможности их использования при проектировании систем имитационного моделирования. Для систем с количеством объектов до 150 допустимо применение оптимизированного полного перебора. Для масштабируемых систем, рассчитанных на сотни и тысячи агентов, использование алгоритмов типа *Sweep and prune* является необходимым условием обеспечения производительности. Полученные количественные оценки позволяют разработчикам принимать обоснованные решения о выборе алгоритмической основы модуля обнаружения коллизий на ранних этапах проектирования.

Литература

- Бунина и др. 2024 – Бунина Л.В., Титов А.П., Лихачев М.А. Разработка модуля сохранения датасета для обнаружения столкновений с использованием полигональной сетки и нейронных сетей // Инженерный вестник Дона. 2024. № 8 (116). С. 178–185. EDN CWAJXR.
- Бурков 2020 – Бурков А. Краткая книга по машинному обучению. М.: ДМК Пресс, 2020. 160 с.
- Собинов, Коробицын 2010 – Собинов Д.И., Коробицын В.В. Алгоритмы обнаружения столкновений // Математические структуры и моделирование. 2010. № 21. С. 82–95. EDN QCRPFT.
- Титов 2024 – Титов А.П. Анализ моделей адаптивных нейро-нечетких систем // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2024. № 1. С. 21–35. DOI: 10.28995/2686-679X-2024-1-21-35. EDN IHFUNN.
- Эрикссон 2022 – Эрикссон К. Алгоритмы обнаружения столкновений в реальном времени. М.: ДМК Пресс, 2022. 632 с.
- Bergen 2003 – van den Bergen G. Collision Detection in Interactive 3D Environments. Burlington, MA: Morgan Kaufmann Publishers, 2003. 290 p.
- Hastie et al. 2009 – Hastie T., Tibshirani R., Friedman J. The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York: Springer, 2009. 745 p.

- Ryan, Massaron 2025 – *Ryan M., Massaron L. Machine Learning for Tabular Data: XGBoost, Deep Learning, and AI*. Shelter Island: Manning Publications, 2025. 504 p.
- Waskom 2021 – *Waskom M. Seaborn: Statistical data visualization* // Journal of Open Source Software. 2021. Vol. 6, no. 60. P. 3021. DOI: 10.21105/joss.03021. EDN YWXRY.
- Wickham 2014 – *Wickham H. Tidy data* // Journal of Statistical Software. 2014. Vol. 59 (10). P. 1–23. DOI: 10.18637/jss.v059.i10.

References

- Bergen van den, G. (2003), *Collision Detection in Interactive 3D Environments*, Morgan Kaufmann Publishers, Burlington, MA, USA.
- Bunina, L.V., Titov, A.P. and Likhachev, M.A. (2024), “Development of a Dataset Saving Module for Collision Detection Using a Polygonal Mesh and Neural Networks”, *Engineering Herald of the Don*, no. 8 (116), pp. 178–185.
- Burkov, A.A. (2010), *Kratkaya kniga po mashinnomu obucheniyu* [Brief book on machine learning], DMK Press, Moscow, Russia.
- Erickson, K. (2022), *Algoritmy obnaruzheniya stolknoveniy v real'nom vremeni* [Real-time collision detection algorithms], DMK Press, Moscow, Russia.
- Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Springer, New York, USA.
- Ryan, M. and Massaron, L. (2025), *Machine learning for tabular data: XGBoost, deep learning, and AI*, Manning Publications, Shelter Island, NY, USA.
- Sobinov, D.I. and Korobitsyn, V.V. (2010), “Collision detection algorithms”, *Mathematical Structures and Modeling*, issue 21, pp. 82–95.
- Titov, A.P. (2024), “Analysis of models of adaptive neurofuzzy systems”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, vol. 1, pp. 21–35, DOI: 10.28995/2686-679X-2024-1-21-35. EDN IHFUNN.
- Waskom, M. (2021), “Seaborn: statistical data visualization”, *Journal of Open Source Software*, vol. 6, no. 60, p. 3021.
- Wickham, H. (2014), “Tidy data”, *Journal of Statistical Software*, vol. 59 (10), pp. 1–23, DOI: 10.18637/jss.v059.i10.

Информация об авторах

Андрей П. Титов, кандидат технических наук, доцент, МИРЭА – Российский технологический университет, Москва, Россия; 119454, Россия, Москва, проспект Вернадского, д. 78;

Институт кибербезопасности и цифровых технологий, Москва, Россия; 107076, Россия, Москва, ул. Строммынка, д. 20; titov_and@mail.ru

Наталья В. Гришина, кандидат технических наук, доцент, Московский государственный лингвистический университет, Москва, Россия; 119034, Россия, Москва, ул. Остоженка, д. 38 стр. 1; grnat@rambler.ru

Дарья Н. Титова, Образовательный центр «Протон», Москва, Россия; 121309, Россия, Москва, ул. Баркляя, д. 15, корп. 3; daratitovaa@gmail.com

Information about the authors

Andrei P. Titov, Cand. of Sci. (Computer Science), associate professor, MIREA – Russian Technological University, Moscow, Russia; 78, Vernadskii Lane, Moscow, 119454, Russia;

Institute of Cybersecurity and Digital Technologies, Moscow, Russia; 20, Stromynka Str., Moscow, 107076, Russia; titov_and@mail.ru

Natalia V. Grishina, Cand. of Sci. (Computer Science), associate professor, Moscow State Linguistic University, Moscow, Russia; 38-1, Ostozhenka Str., Moscow, 119034, Russia; grnat@rambler.ru

Daria N. Titova, Proton Educational Center, Moscow, Russia; 15-3, Barklaya Str., Moscow, 121309, Russia; daratitovaa@gmail.com

Система управления памятью на видеоускорителе с отложенным повторным использованием буферов для WebGPU

Андрей Т. Солдатов

*Московский авиационный институт,
Москва, Россия, mastyfar@yandex.ru*

Наиль З. Султанов

*Российский государственный гуманитарный университет,
Москва, Россия, sultanovnz@mail.ru*

Аннотация. В статье рассматривается система управления памятью видеоускорителя, разработанная для экспериментального фреймворка тензорных вычислений на базе *WebGPU*. Актуальность работы обусловлена ростом интереса к выполнению задач машинного обучения за пределами *CUDA*-экосистемы, в том числе в браузерных и кроссплатформенных средах, где *WebGPU* предоставляет единый доступ к вычислительным возможностям *GPU* через современные нативные графические *API*. Показано, что эффективность тензорных вычислений в такой среде определяется не только производительностью вычислительных шейдеров, но и организацией жизненного цикла *GPU*-буферов, их повторным использованием, синхронизацией между *CPU* и *GPU*, а также обработкой служебных метаданных. В работе предложена архитектура, включающая отдельные аллокаторы для *storage*-буферов и *readback*-буферов, механизм отложенного возврата буферов в пул свободных ресурсов после синхронизации, стратегию повторного использования по принципу *best-fit*, вытеснение неиспользуемых буферов по *LRU*-подходу и отдельную арену метаданных для компактной передачи параметров вычислительных ядер. Особое внимание уделено адаптации известных подходов к управлению памятью, применяемых в *CUDA*-ориентированных *ML*-фреймворках, к ограничениям *WebGPU* и его асинхронной модели исполнения команд. Экспериментальная проверка выполнена на задаче обучения *MLP*-модели на наборе данных *MNIST* с использованием *GPU NVIDIA GeForce RTX 5070*. Полученные результаты показывают, что применение разработанной системы позволяет повысить среднюю производительность на 20,72% по сравнению с наивной реализацией без кэширования буферов. Дополнительно проанализировано влияние параметра *max_ttl* на число новых

аллокаций и повторных использований памяти. Результаты демонстрируют, что при корректной настройке система формирует устойчивый «горячий кэш» и существенно снижает количество дорогостоящих операций создания GPU-буферов. Предложенный подход может быть использован при разработке портативных фреймворков машинного обучения на базе WebGPU, особенно для устройств с ограниченными ресурсами.

Ключевые слова: машинное обучение, WebGPU, GPU, управление памятью GPU, тензорные вычисления, *best-fit*, LRU, вычислительные шейдеры

Для цитирования: Солдатов А.Т., Султанов Н.З. Система управления памятью на видеоускорителе с отложенным повторным использованием буферов для WebGPU // Вестник РГТУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 72–93. DOI: 10.28995/2686-679X-2026-3-72-93

GPU memory management system with deferred buffer reuse for WebGPU

Andrei T. Soldatov

*Moscow Aviation Institute,
Moscow, Russia, mastyfar@yandex.ru*

Nail Z. Sultanov

*Russian State University for the Humanities,
Moscow, Russia, sultanovnz@mail.ru*

Abstract. This paper considers a GPU memory management system developed for an experimental tensor-computation framework based on WebGPU. The relevance of the work is driven by the growing interest in running machine-learning workloads outside the CUDA ecosystem, including in browser-based and cross-platform environments, where WebGPU provides unified access to GPU computing capabilities through modern native graphics APIs. It is shown that the efficiency of tensor computations in such an environment is determined not only by the performance of compute shaders, but also by the organization of the GPU buffer lifecycle, buffer reuse, CPU–GPU synchronization, and the handling of auxiliary metadata. The paper proposes an architecture that includes separate allocators for storage buffers and readback buffers, a mechanism for deferred return of buffers to the free-resource pool after synchronization, a best-fit reuse strategy, eviction of unused buffers based on an LRU approach, and a separate metadata arena for compact transfer of compute-kernel parameters. Particular attention is paid to adapting known memory-management approaches used in CUDA-oriented machine-learning frameworks to the

constraints of WebGPU and its asynchronous command-execution model. The experimental evaluation was performed on the task of training an MLP model on the MNIST dataset using an NVIDIA GeForce RTX 5070 GPU. The obtained results show that the proposed system improves average performance by 20.72 % compared with a naive implementation without buffer caching. In addition, the influence of the `max_ttl` parameter on the number of new allocations and memory reuse operations is analyzed. The results demonstrate that, when configured properly, the system forms a stable “hot cache” and substantially reduces the number of costly GPU-buffer creation operations. The proposed approach can be used in the development of portable machine-learning frameworks based on WebGPU, especially for devices with limited resources.

Keywords: machine learning, WebGPU, GPU, GPU memory management, tensor computations, best-fit, LRU, compute shaders

For citation: Soldatov, A.T. and Sultanov, N.Z. (2026), “GPU memory management system with deferred buffer reuse for WebGPU”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, no. 3, pp. 72–93, DOI: 10.28995/2686-679X-2026-3-72-93

Введение

В последние годы возрастает интерес к выполнению задач машинного обучения за пределами классической *CUDA*-экосистемы, в том числе в браузерных и кроссплатформенных средах. Одним из перспективных направлений является использование *WebGPU* – современного *API*, предоставляющего доступ к вычислительным возможностям *GPU* через вычислительные шейдеры и отображающегося на нативные графические *API*, такие как *Vulkan*, *Metal* и *Direct3D 12* [Kenwright 2022]. Это делает *WebGPU* потенциально применимым не только для графики, но и для тензорных вычислений, тренировки и инференса нейронных сетей и других задач, основанных на массивном параллелизме [Kenwright 2022]. Для задач прикладного моделирования все более значимыми становятся методы машинного обучения и *soft computing*, позволяющие анализировать сложные технические процессы на основе экспериментальных данных [Dayev et al. 2022]. Это важно для настоящей работы, поскольку такие методы часто опираются на тензорные вычисления и могут выигрывать от переноса вычислительной нагрузки на *GPU*. Однако практическое использование *GPU* в исследовательских задачах нередко оказывается связано с зависимостью от *CUDA*-экосистемы и конкретного класса оборудования. Поэтому переход к платформонезависимым решениям, таким как

WebGPU, может расширить применимость тензорных вычислений для моделирования подобных процессов в разных программных и аппаратных средах. Практический интерес к этому направлению подтверждается появлением *WebGPU*-бэкендов в *ML*-инструментах [Ruan et al. 2026], например, *tinygrad* с *WebGPU*-бэкендом и более новый фреймворк *Burn*, в котором *WebGPU* является основным поддерживаемым средством работы с видеоускорителем. Однако эффективность такого подхода определяется не только производительностью отдельных вычислительных шейдеров, но и организацией среды: созданием, переиспользованием, копированием и синхронизацией *GPU*-буферов, то есть буферов, размещаемых в памяти графического ускорителя. Для тензорного фреймворка, где операции порождают большое число промежуточных данных и служебных структур, система управления памятью становится одним из ключевых компонентов производительности и корректности исполнения.

Развитие вычислительных систем, ориентированных на прикладное моделирование и обработку данных, показывает, что эффективность программного комплекса определяется не только выбранным математическим методом, но и архитектурой его реализации. В работах, посвященных автоматизированному проектированию технологических процессов, подчеркивается роль специализированных программных средств, математического аппарата и экспериментальных *CAD*-систем при решении инженерных задач [Chernikova et al. 2018; Zubkova, Tokareva, Sultanov 2018].

Наивное создание *GPU*-буферов для каждой операции приводит к росту накладных расходов и затрудняет контроль жизненного цикла памяти. В то же время немедленное повторное использование освобожденных буферов может быть некорректным из-за асинхронной модели выполнения команд на *WebGPU*. С точки зрения *CPU* буфер уже может считаться свободным, тогда как ранее отправленные команды все еще используют его на *GPU*. Это делает актуальной специализированную систему управления памятью, учитывающую типы буферов, характер их использования и точки синхронизации между *CPU* и *GPU*.

Целью данной статьи является описание и анализ системы управления памятью, разработанной для экспериментального фреймворка тензорных вычислений на *WebGPU*. Рассматриваемая система предназначена для организации жизненного цикла *GPU*-буферов и служебных метаданных, используемых при выполнении тензорных операций. В рамках статьи рассматриваются отдельные аллокаторы для *storage*-буферов и *readback*-буферов, механизм отложенного повторного использования памяти после

синхронизации CPU и GPU, вытеснение неиспользуемых буферов, а также арена метаданных, применяемая для компактной передачи параметров вычислительных ядер.

Объектом исследования в данной статье является процесс управления памятью на видеоускорителе. Предметом исследования выступают алгоритмы и методы оптимизации работы с памятью на видеоускорителе и адаптация методов под ограничения *WebGPU*.

Научная новизна работы заключается в адаптации подходов к кэшированию и повторному использованию GPU-буферов к ограничениям *WebGPU* и его асинхронной модели выполнения команд. В отличие от *CUDA*-ориентированных решений, предложенная система учитывает модель явно создаваемых *GPUBuffer*, ограничения на режимы их использования и необходимость отложенного возврата буферов в пул после синхронизации CPU и GPU.

Практическая ценность работы состоит в возможности применения разработанной системы в портативных фреймворках машинного обучения на базе *WebGPU*. Предложенный подход может снизить число дорогостоящих операций создания буферов, уменьшить накладные расходы на управление памятью и повысить производительность на устройствах с ограниченными ресурсами, включая ноутбуки, мобильные устройства и тонкие клиенты.

Существующие решения

Поскольку для *WebGPU* пока не сформировался набор устоявшихся решений по управлению памятью в тензорных рантаймах, архитектура рассматриваемой системы была спроектирована с учетом подходов, применяемых в фреймворках машинного обучения, использующих *CUDA*. В качестве ориентиров рассматривались *PyTorch CUDA Caching Allocator* и *TensorFlow BFC Allocator*. Из *BFC Allocator* была заимствована идея выбора наиболее подходящего свободного блока для повторного использования по стратегии *best-fit* [Meng et al. 2017]. Из *caching allocator* была использована общая идея сохранения ранее выделенных участков памяти в пуле вместо их немедленного освобождения, что позволяет снижать накладные расходы на создание новых GPU-ресурсов. При этом предложенная система адаптирована к особенностям *WebGPU*, где работа с памятью строится вокруг явно создаваемых буферов, а исполнение команд происходит асинхронно относительно CPU [Kenwright 2022].

В отличие от универсальных аллокаторов, рассчитанных на произвольные последовательности выделений и освобождений

памяти, рассматриваемая система ориентирована на типичный сценарий машинного обучения: многократное выполнение одного и того же графа операций в цикле обучения или инференса [Domíngos 2012]. В таком режиме размеры и порядок временных аллокаций часто повторяются между итерациями. Поэтому в текущей реализации было принято решение отказаться от динамического расщепления и слияния блоков памяти. Это упрощает реализацию, уменьшает накладные расходы аллокатора и делает поведение системы более предсказуемым. Поскольку *WebGPU* не предоставляет механизма представления нескольких независимых *GPUBuffer* как одного непрерывного *storage-buffer binding*, объединение свободных блоков на уровне пула буферов невозможно без создания нового буфера и копирования данных. В связи с этим система использует стратегию повторного использования целых буферов, а не динамическое слияние и расщепление блоков.

Архитектура решения

Для определения архитектуры системы необходимо рассмотреть типовой сценарий выполнения вычислительной операции с использованием *WebGPU*, например, матричного умножения. На первом этапе создаются буферы, содержащие входные матрицы, а также буфер, предназначенный для хранения результата вычисления. Дополнительно формируется буфер с параметрами операции, через который в вычислительное ядро передается метаданные о входных данных, поскольку буферы *WebGPU* представляют собой линейные области памяти и сами по себе не содержат сведений о структуре размещенных в них данных.

Затем указанные буферы объединяются в структуру *GPUBindGroup*, задающую набор ресурсов, доступных вычислительному шейдеру, после этого создается *GPUComputePipeline*, включающий описание вычислительного пайплайна и скомпилированный код шейдера. Команды выполнения, содержащие ссылку на *GPUComputePipeline* и соответствующую *GPUBindGroup*, передаются через *GPUQueue*, которая ставит задачу на исполнение графическому ускорителю. После завершения вычислений результат матричного умножения находится в выходном буфере, размещенном в памяти *GPU*. *WebGPU* не предоставляет гарантий, что после вызова метода отправки команд *GPUQueue* на обработку, все операции будут выполнены. В этом заключается асинхронная модель вычислений в *WebGPU*. Для того чтобы удостовериться, что операция была корректно выполнена, необходимо вызвать команду ожидания,

например, `device.poll()`, которая приостановит выполнение команд потока до тех пор, пока все операции на *GPU* не будут выполнены.

Непосредственное чтение результирующего буфера со стороны *CPU* невозможно, поэтому для получения результата необходимо создать промежуточный буфер, доступный для отображения в адресное пространство *CPU*. Такой буфер выполняет роль прокси между памятью графического ускорителя и оперативной памятью, позволяя считать данные, полученные в результате вычислений. При этом использовать подобные буферы непосредственно в процессе вычислений нельзя, поскольку отображаемая в память *CPU* область не обладает необходимыми гарантиями безопасности, требуемыми для использования их в шейдерах.

То, какими свойствами обладает буфер в *WebGPU*, определяется флагом *GPUBufferUsage*, который может принимать любую комбинацию значений, определенных в табл. 1.

Таблица 1

Флаги *GPUBufferUsage*

COPY_SRC	Буфер можно использовать как источник операции копирования, например как аргумент <i>source</i> при вызове <i>copyBufferToBuffer()</i> .
COPY_DST	Буфер можно использовать как назначение операции копирования/записи, например как аргумент <i>destination</i> при вызове <i>copyTextureToBuffer()</i> .
INDEX	Буфер можно использовать как индексный буфер, например как аргумент <i>buffer</i> , передаваемый в <i>setIndexBuffer()</i> .
INDIRECT	Буфер можно использовать для хранения аргументов не прямых команд, например как аргумент <i>indirectBuffer</i> при вызове <i>drawIndirect()</i> или <i>dispatchWorkgroupsIndirect()</i> .
MAP_READ	Буфер можно отображать в память для чтения, например при вызове <i>mapAsync()</i> с режимом <i>GPUMapMode.READ</i> . Этот флаг можно комбинировать только с <i>GPUBufferUsage.COPY_DST</i> .
MAP_WRITE	Буфер можно отображать в память для записи, например при вызове <i>mapAsync()</i> с режимом <i>GPUMapMode.WRITE</i> . Этот флаг можно комбинировать только с <i>GPUBufferUsage.COPY_SRC</i> .

Окончание табл. 1

QUERY_RESOLVE	Буфер можно использовать для сохранения результатов запросов, например как аргумент <i>destination</i> при вызове <i>resolveQuerySet()</i> .
STORAGE	Буфер можно использовать как <i>storage</i> -буфер, например как ресурс в записи <i>bind group</i> при создании <i>GPUBindGroup</i> через <i>createBindGroup()</i> , соответствующей записи <i>GPUBindGroupLayout</i> с типом <i>layout</i> -привязки буфера <i>storage</i> или <i>read-only-storage</i> .
UNIFORM	Буфер можно использовать как <i>uniform buffer</i> , например как ресурс в записи <i>bind group</i> при создании <i>GPUBindGroup</i> через <i>createBindGroup()</i> , соответствующей записи <i>GPUBindGroupLayout</i> с типом <i>layout</i> -привязки буфера <i>uniform</i> .
VERTEX	Буфер можно использовать как вершинный буфер, например как аргумент <i>buffer</i> , передаваемый в <i>setVertexBuffer()</i> .

В контексте данной работы нас интересуют только флаги *COPY_SRC*, *COPY_DST*, *MAP_READ* и *STORAGE*, так как именно буферы с этими флагами будут использоваться в разрабатываемой системе. Остальные типы буферов используются в основном, в задачах компьютерной графики и мало релевантны для этой работы.

Исходя из анализа типового сценария вычислений в *WebGPU*, систему управления памятью целесообразно разделить на три основных компонента:

- аллокатор *Storage*-буферов, помеченных флагом *STORAGE*, используемых в качестве входных и выходных данных вычислительных операций;
- аллокатор *Readback*-буферов, помеченных флагом *MAP_READ*, предназначенных для передачи результатов вычислений из памяти *GPU* в память *CPU*;
- арену метаданных, через которую в шейдеры передаются параметры операций.

Такое разделение отражает различие в назначении и жизненном цикле соответствующих типов данных. *Storage*-буферы участвуют непосредственно в вычислительных задачах, *Readback*-буферы используются только на этапе извлечения результата, а метадан-

ные представляют собой небольшие временные блоки параметров, необходимые для настройки некоторых типов операций. Поскольку создание GPU-буферов является относительно дорогостоящей операцией, ключевой целью системы становится повторное использование уже выделенных ресурсов и сокращение числа обращений к механизму аллокации памяти видеоускорителя.

Аллокатор Storage- и Readback-буферов

Так как логика работы Storage и Readback аллокаторов во многом совпадает, их целесообразно реализовать на основе единой абстракции. Основные различия между ними сводятся к флагам использования создаваемых буферов, а также к набору операций, допустимых над буферами, выдаваемыми аллокатором. Поэтому в системе был выделен общий базовый класс BufferAllocator, поверх которого инкапсулируются специализированные шаблонные классы-обертки.

Архитектура базового аллокатора строится вокруг трех взаимосвязанных компонентов: глобального хранилища store, промежуточного множества pending и пула свободных буферов free pool, организованного на основе B-дерева. Эти структуры отражают полный жизненный цикл буфера и весь его набор состояний: InUse, Pending и Free. Обзор архитектуры BufferAllocator представлен на рис. 1.

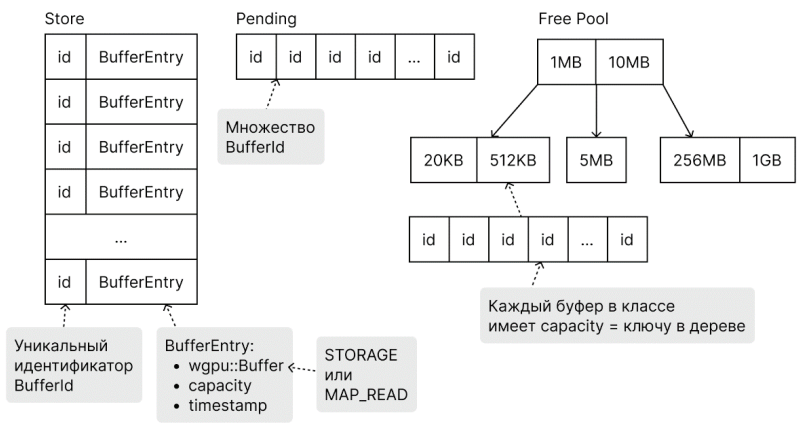


Рис. 1. Архитектура BufferAllocator

Основным хранилищем буферов является *store* – объект типа *SlotMap*. По своей роли он близок к ассоциативному контейнеру ключ-значение (*HashMap*), однако в отличие от обычного *HashMap*, самостоятельно выдает уникальный идентификатор каждому добавленному объекту и поддерживает повторное использование освободившихся ключей. Добавление, удаление и индексирование в *SlotMap* имеют среднюю временную сложность $O(1)$, что подходит для организации глобального хранилища. *Store* является локальным для конкретного аллокатора и содержит записи обо всех буферах, созданных данным аллокатором, независимо от их текущего состояния. Каждая запись представляет собой объект типа *BufferEntry* и включает в себя ссылку на *wgpu::Buffer*, его емкость и временную метку последнего использования. При добавлении нового буфера *store* выдает ему идентификатор *BufferId*, который далее используется во всех остальных структурах аллокатора. Благодаря этому *pending* и *free_pool* не владеют буферами напрямую, а хранят только их идентификаторы. Такой подход централизует информацию о буфере в одном месте и упрощает переходы между состояниями.

Выдача буфера выполняется методом *request*. Сначала аллокатор пытается найти в *free_pool* уже существующий буфер подходящего размера. Поскольку *free_pool* реализован как отображение на основе *B*-дерева, где ключом выступает класс емкости буфера, поиск начинается с минимального класса емкости, не меньшего запрошенного размера. Временная сложность данной операции $O(\log_B N)$, где *B* – фактор ветвления дерева, а *N* – общее количество классов емкости. По этой причине и было решено использовать контейнер на основе *B*-дерева, так как он позволяет найти нужный класс емкости за менее чем линейное время.

Найденный кандидат используется только в том случае, если его емкость не приводит к чрезмерному перерасходу памяти; это проверяется отдельной функцией *validate_size*. Если подходящий буфер найден, он извлекается из пула, его временная метка обновляется, и он возвращается вызывающему коду. Таким образом реализуется стратегия *best-fit*. Если подходящего буфера нет, создается новый GPU-буфер с флагами использования, соответствующими типу аллокатора (*STORAGE*, *MAP_READ*). Результатом выполнения метода *request* является RAII структура *BufferLease*, которая держит соответствующий ей буфер в состоянии *InUse* на протяжении своего существования.

Необходимость отдельного промежуточного состояния *Pending* связана с асинхронной природой *WebGPU*. После отправки вычислительной операции в *GPUQueue* нельзя считать, что ис-

пользованные в ней буферы сразу становятся доступными для повторного применения. В большинстве случаев они продолжают участвовать в выполнении соответствующего шейдера на стороне *GPU*. После логического освобождения буфера кодом, который его использовал, буфер не помещается непосредственно в пул свободных буферов. Вместо этого он добавляется во множество *pending*.

Для перевода таких буферов в состояние *Free* используется специальный метод *reclaim*. Этот метод синхронизируется с выполнением ранее отправленных *GPU*-команд, дожидается их завершения и только после этого переносит буферы из *pending* в *free_pool*. Таким образом, *Pending* выполняет роль шлюза между состояниями *InUse* и *Free*.

После завершения синхронизации буферы попадают в *free_pool*. Эта структура представляет собой контейнер на основе *B*-дерева, где ключом выступает класс емкости буфера, а значением – массив свободных буферов соответствующего размера. При возврате буфера в пул его емкость используется для определения подходящего класса, после чего буфер добавляется в соответствующий массив.

Для предотвращения неограниченного роста пула свободных буферов в аллокаторе реализован механизм вытеснения (*eviction*) по стратегии *LRU* (*least recently used*), то есть память вытесняется на основе значения внутреннего счетчика [Сибиряков 2017]. Внутри каждого аллокатора создается монотонный счетчик *clock*, увеличиваемый при каждом запросе буфера. Каждый буфер в *store* содержит временную метку, отражающую момент его последнего использования. Значение этой временной метки задается текущим значением *clock* при выдаче буфера из аллокатора: как при создании нового буфера, так и при повторном использовании буфера из *free_pool*. Таким образом, чем меньше значение временной метки, тем дольше буфер не использовался и тем выше его приоритет для удаления из кэша.

Вытеснение выполняется методом *evict* и состоит из двух этапов. На первом этапе, если вытеснение не является принудительным, из *free_pool* удаляются буферы, время жизни которых в кэше превысило заданный параметр *max_ttl*. Этот проход позволяет освободить давно неиспользуемые буферы независимо от текущего давления на память. На втором этапе вычисляется допустимый размер кэша свободных буферов. Он определяется как доля от объема памяти, занятого активными буферами, и затем ограничивается снизу и сверху параметрами *soft_cache_limit* и *hard_cache_limit*.

Если фактический объём *free_pool* превышает рассчитанный предел, аллокатор формирует список свободных буферов, сортирует их по временной метке и удаляет наиболее старые элементы до тех пор, пока не будет освобожден требуемый объем памяти. Так как формирование списка кандидатов на удаление предполагает сортировку массива буферов по их временной метке, временная сложность операции *evict* оценивается как $O(n \log n)$ в среднем случае [Сибиряков 2017]. Тем самым *free_pool* сохраняет недавно использованные буферы, которые с большей вероятностью будут востребованы повторно, и освобождает буферы, давно не участвовавшие в вычислениях.

Чтобы вытеснение не выполнялось слишком часто и не превращалось в дополнительный источник накладных расходов, используется параметр *eviction_debounce*. Метод *try_evict* запускает мягкое вытеснение только в том случае, если с момента предыдущего прохода прошло достаточно запросов к аллокатору ($clock - last_eviction \geq eviction_debounce$). Помимо этого, предусмотрен режим принудительного вытеснения: при неудачной попытке создать новый буфер вызывается *hard_evict*, который пропускает *TTL*-проход и удаляет свободные буферы по *LRU*-порядку до достижения заданного целевого объёма. Такой режим используется как аварийный механизм освобождения памяти при ошибке аллокации.

MetadataArena

Помимо аллокаторов для *Storage* и *Readback* буферов, в системе используется отдельная структура *MetadataArena*, используемая для размещения метаданных, передаваемых в шейдеры.

Наивный подход состоял бы в создании отдельного *GPU*-буфера для каждого такого блока метаданных. Однако метаданные обычно имеют малый размер (< 1 Кбайт) и создаются часто, поэтому подобная схема приводила бы к большому количеству мелких *GPU*-аллокаций. Это увеличивает накладные расходы на создание буферов и усложняет управление их временем жизни. *MetadataArena* решает эту проблему за счет выделения крупных страниц фиксированного размера, внутри которых небольшие блоки метаданных размещаются последовательно с выравниванием, определенным стандартом *WebGPU*. Таким образом, множество небольших логических выделений памяти сводится к небольшому числу физических *GPU*-буферов. Схема работы арены метаданных представлена на рис. 2.

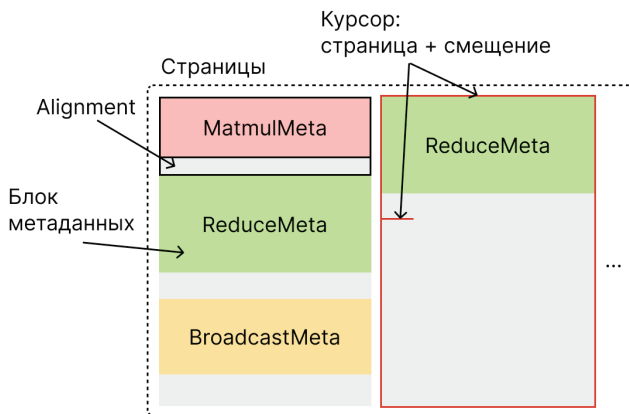


Рис. 2. Схема работы *Metadarena*

Архитектурно *Metadarena* представляет собой линейный *page-based allocator*. Внутри структуры находятся массив страниц *pages*, текущий курсор записи, емкость одной страницы и значение выравнивания. Курсор состоит из двух полей: номера текущей страницы и смещения внутри нее. Каждая страница является объектом *wgpu::Buffer*, так как *WebGPU* позволяет передавать в *BindGroup* не целые буферы, а набор размер + смещение, что позволяет использовать один большой блок как контейнер для множества небольших логических выделений памяти.

При создании *Metadarena* запрошенная емкость страницы сначала выравнивается вверх относительно ограничения устройства. После этого создается первая страница арены, а курсор устанавливается в начало: нулевая страница и нулевое смещение. Остальные страницы создаются по требованию, только если текущей страницы оказывается недостаточно для очередной аллокации.

Основная операция арены – метод *allocate*, принимающий байтовое представление метаданных. Сначала проверяется, помещается ли запрошенный блок в одну страницу с учетом выравнивания. Если размер блока превышает емкость страницы, метод возвращает ошибку, поскольку одна логическая запись метаданных не может быть распределена между несколькими страницами.

Важной особенностью *Metadarena* является отсутствие индивидуального освобождения блоков. В отличие от *BufferAllocator*, где каждый буфер проходит состояния *InUse*, *Pending* и *Free*, метаданные выделяются линейно и освобождаются сразу всей ареной. Для этого используется метод *reset*, который просто возвращает

курсор в начало первой страницы, не удаляя уже созданные страницы. После сброса последующие аллокации начинают переиспользовать существующие буферы с начала. Если ранее арена выросла до нескольких страниц, эти страницы сохраняются и могут быть повторно использованы в следующих вычислительных проходах. Такое решение оправдано для задач машинного обучения, где нагрузка обычно имеет циклический характер: одни и те же операции многократно выполняются в цикле обучения или инференса, а значит, объем требуемых метаданных от итерации к итерации остается близким к уже достигнутому пиковому значению.

Проверка результата

Экспериментальная проверка разработанной системы аллокации памяти проводилась на задаче обучения нейросетевой модели *MLP* на наборе данных *MNIST*. Общее число параметров модели составило 235146.

Тестирование проводилось на графическом ускорителе *NVIDIA GeForce RTX 5070*.

На первом этапе было выполнено сравнение производительности разработанной системы управления памятью с наивной реализацией. В наивной реализации выделение памяти под буферы и метаданные выполнялось по требованию, без использования кэширования.

Результаты тестирования представлены на рисунке 3.

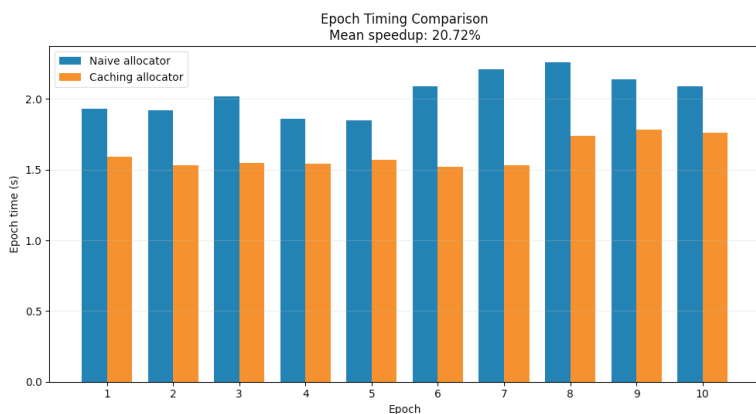


Рис. 3. Сравнение производительности наивного аллокатора с разработанной системой

Как видно из приведенных данных, применение разработанной системы управления памятью при оптимальных настройках позволило повысить среднюю производительность на 20,72% по сравнению с базовой реализацией. Следует учитывать, что работа с буферами – это лишь часть функционала фреймворка машинного обучения, и многие особенности реализации могут влиять на производительность, однако тестирование показывает эффективность метода в типовом сценарии использования.

Во втором эксперименте исследовалась зависимость количества новых аллокаций и повторных использований буферов от значения параметра *max_ttl*. Данный параметр задает время жизни буфера в кэше относительно внутренних часов аллокатора. Результаты эксперимента представлены в табл. 2 и 3, а также на рис. 4...6.

Таблица 2

Статистика *Storage* аллокатора

max_ttl	Новые аллокации	Повторно использованные	Удаленные буферы
1	300603	2499	299973
128	46553	256549	45835
256	739	302363	0

Таблица 3

Статистика *Readback* аллокатора

max_ttl	Новые аллокации	Повторно использованные	Удаленные буферы
0	10002	4998	10000
128	5	14995	0
256	5	14995	0

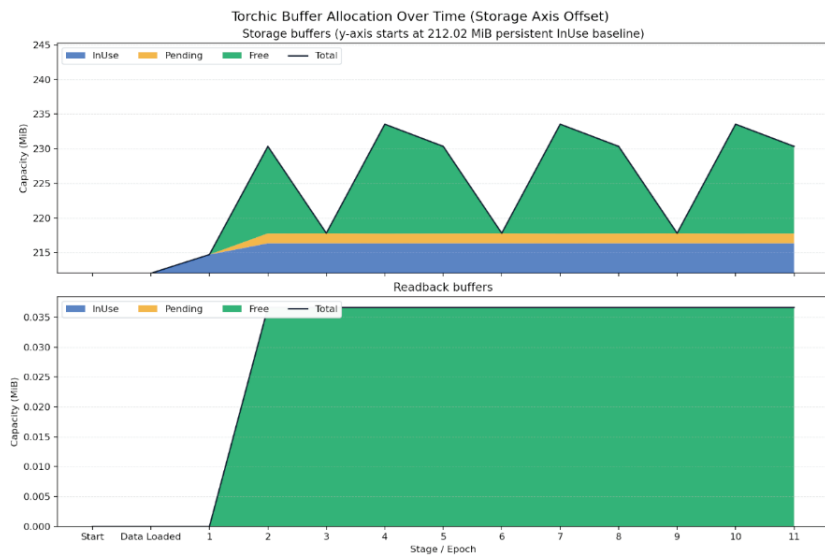


Рис. 4. График состояния *Storage* и *Readback* аллокаторов при $max_ttl = 1$

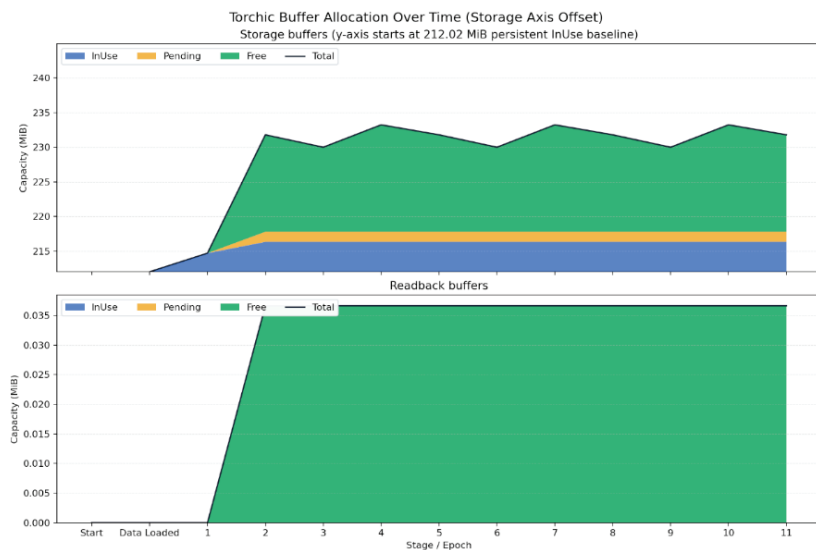


Рис. 5. График состояния *Storage* и *Readback* аллокаторов при $max_ttl = 128$

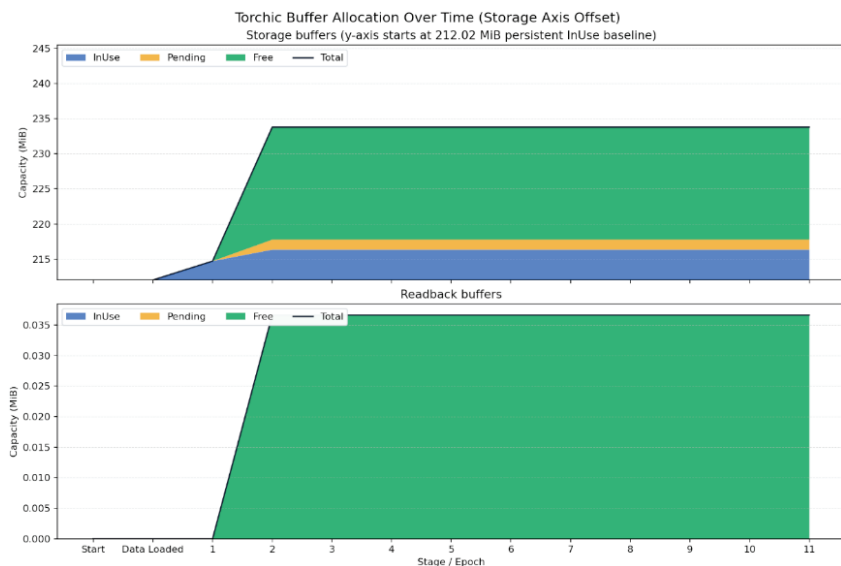


Рис. 6. График состояния *Storage* и *Readback* аллокаторов при $max_ttl = 256$

Полученные данные показывают, что при корректном выборе значения max_ttl аллокатор выходит в устойчивый режим работы: количество новых аллокаций ограничивается числом буферов, необходимых для хранения параметров модели, данных и временных тензоров вычислительного графа, тогда как последующие запросы на выделение памяти обслуживаются за счет повторного использования ранее выделенных буферов, что приводит к формированию стабильного «горячего кэша». При слишком малых значениях буферы преждевременно удаляются из *free_pool*, что приводит к росту числа новых аллокаций и снижает эффект от кэширования. Тем самым настройка max_ttl фактически становится одним из элементов управления балансом между расходом памяти и производительностью. В более общем виде подобная задача может рассматриваться как задача выбора параметров вычислительной системы при наличии нескольких критериев, что сближает ее с подходами оптимизации и поддержки принятия решений, применяемыми в технических системах управления [Dayev, Kairakbayev, Sultanov 2021; Тугов, Орлова, Султанов 2025]. Задача оптимизации данного параметра не ставится в данной статье и рассматривается в разделе «Ограничения текущей реализации».

Ограничения текущей реализации

Текущая реализация имеет ряд ограничений.

Во-первых, система имеет значительное число вручную задаваемых параметров конфигурации, что ограничивает потенциал ее использования в автоматизированных системах. Настройки политики утилизации буферов, настройки размера страниц для арены метаданных, ручное планирование вызовов методов *reclaim* и *evict* и т. д. Все эти настройки могут быть оптимально подобраны для обеспечения максимальной производительности, как показано в эксперименте, однако для реального использования система должна уметь подбирать параметры автоматически или по крайней мере предоставлять оптимальные значения по умолчанию.

Еще одним недостатком является невозможность централизованного управления аллокаторами. Например, метод *hard_evict* будет вытеснять буферы только в своем локальном аллокаторе, даже если при этом виновником переполнения является другой аллокатор.

Также одно из архитектурных ограничений системы – это наличие точек глобальной синхронизации, то есть точек, в которых система останавливает выполнение и ожидает завершения всех операций на видеоускорителе для того, чтобы гарантировать безопасность манипуляций с буферами. В текущей реализации синхронизация происходит в методах *reclaim* (перенос буферов из *pending* в *free_pool*) и *download* (скачивание *readback* буфера на *CPU*). В идеальной реализации такие методы должны ограничивать выполнение программы только на то время, которое необходимо для освобождения запрошенных буферов, а не всех сразу. Потенциальным решением этой проблемы видится присвоение *BufferEntry* соответствующих *SubmissionIndex*, относящихся к операциям над этими буферами, но это требует дополнительного исследования.

Также следует отметить ограниченность тестирования. Разработанная система проверялась только на одном аппаратном устройстве и на одном типе задачи машинного обучения, что существенно ограничивает возможность оценки пригодности решения для других типов задач.

Компромиссом в данной системе является отсутствие явной системы субаллокации и расщепления и слияния блоков. Такой дизайн был выбран осознанно и его обоснование приведено в начале работы.

Заключение

В работе была представлена система управления памятью на видеоускорителе для тензорных вычислений на *WebGPU*. Ее основная задача заключается в снижении накладных расходов, возникающих при частом создании временных буферов, а также в обеспечении корректного повторного использования памяти с учетом асинхронного выполнения команд на *GPU*.

Предложенный подход основан на разделении буферов по назначению, отложенном возврате освобожденных ресурсов в пул повторного использования и отдельной арене для хранения метаданных вычислительных операций. Такое разделение позволяет согласовать модель управления памятью с характером тензорных вычислений, где значительная часть ресурсов имеет временный жизненный цикл, но при этом не может быть безопасно повторно использована сразу после логического освобождения на стороне *CPU*.

Экспериментальная проверка системы на задаче обучения *MLP*-модели на наборе *MNIST* демонстрирует повышение средней производительности на 20,72% по сравнению с базовой реализацией без кэширования буферов. Дополнительный анализ параметра *max ttl* показал, что при корректной настройке аллокатор выходит в устойчивый режим, в котором большинство запросов обслуживается за счет повторного использования уже созданных *GPU*-буферов.

Таким образом, результаты работы показывают, что предложенная система позволяет сократить число дорогостоящих аллокаций, повысить производительность и сохранить корректность исполнения в условиях асинхронной работы *CPU* и *GPU*. Дальнейшее развитие может быть связано с автоматическим подбором параметров кэширования и проверкой подхода на более крупных моделях и вычислительных графах, а также с расширением функционала системы для работы с несколькими физическими *GPU*.

Литература

- Сибиряков 2017 – Сибиряков М.А. Разработка алгоритмов ускоренного вытеснения информации в больших хранилищах данных // Кибернетика и программирование. 2017. № 4. С. 75–83. EDN ZFIDKX.
- Тугов, Орлова, Султанов 2025 – Тугов В.Б., Орлова Е.В., Султанов Н.З. Система поддержки принятия решений по замене средств физической защиты на основе метода анализа иерархий // Современные наукоемкие технологии. 2025. № 8. С. 147–153. DOI: 10.17513/snt.40477. EDN FHHVKA.

- Chernikova et al. 2018 – *Chernikova N.N., Sultanov N.Z., Zubkova T.M., Sherstobitova V.N.* Preproduction Using Automated Process Design // *Russian Engineering Research*. 2018. Vol. 38, No. 5. P. 384–386. DOI: 10.3103/S1068798X18050052. EDN YBTQTB.
- Dayev, Kairakbayev, Sultanov 2021 – *Dayev Z.A., Kairakbayev A.K., Sultanov N.Z.* Development of humidity of natural gas control system based on linear programming methods // *Energy Systems*. 2021. Vol. 12, No. 1. P. 233–245. DOI: 10.1007/s12667-019-00353-z. EDN PRTTSF.
- Dayev et al. 2022 – *Dayev Z., Shopanova G., Toksanbaeva B., Sultanov N., Yetilmezsoy K., Kiyan E., Sihag P., Bahramian M.* Modeling the flow rate of dry part in the wet gas mixture using decision tree/kernel/non-parametric regression-based soft-computing techniques // *Flow Measurement and Instrumentation*. 2022. Vol. 86. P. 102195. DOI: 10.1016/j.flowmeasinst.2022.102195. EDN VBFWVGZ.
- Domingos 2012 – *Domingos P.* A few useful things to know about machine learning // *Communications of the ACM*. 2012. Vol. 55. No. 10. P. 78–87.
- Kenwright 2022 – *Kenwright B.* Introduction to the WebGPU API // *ACM SIGGRAPH 2022 Courses*. 2022. SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference. Vancouver, Canada. New York, NY: ACM, 2022. P. 1–184.
- Meng et al. 2017 – *Meng C., Sun M., Yang J., Qiu M., Gu Y.* Training deeper models by GPU memory optimization on TensorFlow // *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA. 2017. San Diego, CA, 2017. URL: http://learningsys.org/nips17/assets/papers/paper_18.pdf (дата обращения: 15.05.2026).
- Ruan et al. 2026 – *Ruan C.F., Qin Y., Parthasarathy A.R., Zhou X., Lai R., Jin H., Dong Y., Hou B., Yu M.-S., Zhai Y., Agarwal S., Cao H., Feng S., Chen T.* WebLLM: A high-performance in-browser LLM inference engine, 2026. URL: <https://arxiv.org/html/2412.15803> (дата обращения: 15.05.2026).
- Zubkova, Tokareva, Sultanov 2018 – *Zubkova T.M., Tokareva M.A., Sultanov N.Z.* Creation of system of computer-aided design for technological objects // *Journal of Physics: Conference Series*, Tomsk, 17–20 Jan. 2018. Vol. 1015 5. Bristol: Institute of Physics Publishing, 2018. P. 052031. DOI: 10.1088/1742-6596/1015/5/052031. EDN XXYRSH.

References

- Chernikova, N.N., Sultanov, N.Z., Zubkova, T.M. and Sherstobitova, V.N. (2018), “Preproduction using automated process design”, *Russian Engineering Research*, vol. 38, no. 5, pp. 384–386, DOI: 10.3103/S1068798X18050052. EDN YBTQTB.
- Dayev, Z.A., Kairakbayev, A.K. and Sultanov, N.Z. (2021), “Development of humidity of natural gas control system based on linear programming methods”, *Energy Systems*, vol. 12, no. 1, pp. 233–245, DOI: 10.1007/s12667-019-00353-z.

- Dayev, Z., Shopanova, G., Toksanbaeva, B., Sultanov, N., Yetilmesoy, K., Kiyan, E., Sihag, P. and Bahramian, M. (2022), "Modeling the flow rate of dry part in the wet gas mixture using decision tree/kernel/non-parametric regression-based soft-computing techniques", *Flow Measurement and Instrumentation*, vol. 86, p. 102195, DOI: 10.1016/j.flowmeasinst.2022.102195.
- Domingos, P. (2012), "A few useful things to know about machine learning", *Communications of the ACM*, vol. 55, no. 10, pp. 78–87.
- Kenwright, B. (2022), "Introduction to the WebGPU API", ACM SIGGRAPH 2022 Courses. 2022. *SIGGRAPH '22: Special Interest Group on Computer Graphics and Interactive Techniques Conference*. Vancouver, Canada, ACM, New York, NY, USA, pp. 1–184.
- Meng, C., Sun, M., Yang, J., Qiu, M. and Gu, Y. (2017), "Training deeper models by GPU memory optimization on TensorFlow", *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017, San Diego, CA, USA, available at: http://learningsys.org/nips17/assets/papers/paper_18.pdf (Accessed 15 May 2026).
- Ruan, C.F., Qin, Y., Parthasarathy, A.R., Zhou, X., Lai, R., Jin, H., Dong, Y., Hou, B., Yu, M.-S., Zhai, Y., Agarwal, S., Cao, H., Feng, S. and Chen, T. (2024), "WebLLM: A high-performance in-browser LLM inference engine", available at: <https://arxiv.org/html/2412.15803> (Accessed 15 May 2026).
- Sibiryakov, M.A. (2017), "Development of algorithms for accelerated eviction of information in large data warehouses", *Cybernetics and Programming*, no. 4, pp. 75–83, EDN ZFIDKX.
- Tugov, V.B., Orlova, E.V. and Sultanov, N.Z. (2025), "A decision support system for replacing physical protection means based on the analytic hierarchy process", *Modern High Technologies*, no. 8, pp. 147–153, DOI: 10.17513/snt.40477, EDN FHHVKA.
- Zubkova, T.M., Tokareva, M.A. and Sultanov, N.Z. (2018), "Creation of system of computer-aided design for technological objects", *Journal of Physics: Conference Series*, Tomsk, 17–20 January 2018, vol. 1015, article 052031, Institute of Physics Publishing, Bristol, UK, DOI: 10.1088/1742-6596/1015/5/052031.

Информация об авторах

Андрей Т. Солдатов, студент, Московский авиационный институт, Москва, Россия; 125993, Россия, Москва, Волоколамское шоссе, д. 4; mas-tyfar@yandex.ru

Наиль З. Султанов, доктор технических наук, профессор, Российский государственный гуманитарный университет, Москва, Россия; 125047, Россия, Москва, Миусская пл., д. 6, стр. 6; sultanovnz@mail.ru

Information about the authors

Andrei T. Soldatov, student, Moscow Aviation Institute, Moscow, Russia; bld. 4, Volokolamskoye Av., Moscow, 125993, Russia; mastyfar@yandex.ru

Nail Z. Sultanov, Dr. of Sci. (Mechanical Engineering), professor, Russian State University for the Humanities, Moscow, Russia; 6-6, Miusskaya Sq., Moscow, 125047, Russia; sultanovnz@mail.ru

Информационная безопасность

УДК 004.056

DOI: 10.28995/2686-679X-2026-3-94-113

Направления и методы обнаружения и противодействия АРТ-атакам на объектах информатизации

Сергей И. Козьминых

МИРЭА – Российский технологический университет, Москва, Россия;

Российский экономический университет им. Г.В. Плеханова,

Москва, Россия;

Финансовый университет при Правительстве

Российской Федерации, Москва, Россия, kozminykh@mirea.ru

Александр Ю. Дубинин

Финансовый университет при Правительстве

Российской Федерации, Москва, Россия, 249426@edu.fa.ru

Аннотация. В последнее время СМИ все чаще сообщают об успешных кибератаках на крупные компании, как зарубежные, так и отечественные, включая организации, относящиеся к объектам критической информационной инфраструктуры. Целью статьи является исследование термина *advanced persistent threat* (АРТ-атак) в нормативно-правовой документации в области информационной безопасности, анализ направлений и методов обнаружения этих атак, а также разработка рекомендаций по их противодействию. В статье приведены примеры успешных АРТ-атак на крупные зарубежные и российские компании, в том числе являющиеся субъектами критической информационной инфраструктуры. Проведен анализ зарубежных и отечественных стандартов, публикаций профильных организаций, а также научных статей, который показывает разнообразие трактовок термина *advanced persistent threat*. Проведенный анализ позволил выделить признаки, присущие АРТ-атакам, отличающие их от обычных кибератак, а также уточняет цели АРТ-атак. Определено, что к признакам таких атак относится неоднократное преследование своих целей в течение длительного периода времени, адаптируемость к усилиям обороняющихся сторон, стремление поддерживать наличие скрытых каналов (C2). Выявлено, что целями АРТ-атак являются отслеживание различных активностей в сетях и информационных системах организации, кража конфиденциальной информации, нарушение нормального

© Козьминых С.И., Дубинин А.Ю., 2026

функционирования организации. В статье проведен анализ направлений и методов обнаружения C2-каналов, а также разработаны рекомендации по противодействию *advanced persistent threat* атакам на объектах информатизации.

Ключевые слова: АРТ-атака, признаки АРТ-атак, C2-каналы, метод противодействия АРТ-атакам, рекомендации по противодействию АРТ-атакам

Для цитирования: Козьминых С.И., Дубинин А.Ю. Направления и методы обнаружения и противодействия АРТ-атакам на объектах информатизации // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 94–113. DOI: 10.28995/2686-679X-2026-3-94-113

Directions and methods for detecting and countering APT attacks on information systems

Sergei I. Kozminykh

MIREA – Russian Technological University, Moscow, Russia;

Plekhanov Russian University of Economics, Moscow, Russia;

*Financial University under the Government of the Russian Federation,
Moscow, Russia, kozminykh@mirea.ru*

Aleksandr Yu. Dubinin

*Financial University under the Government of the Russian Federation,
Moscow, Russia, 249426@edu.fa.ru*

Abstract. Recently, the media has increasingly reported on successful cyberattacks on large companies, both foreign and domestic, including organizations related to critical information infrastructure. The purpose of this article is to study the term advanced persistent threat (APT attacks) in regulatory documentation in the field of information security, analyze the directions and methods of detecting these attacks, and develop recommendations for countering them. The article provides examples of successful APT attacks on large foreign and Russian companies, including those that are subjects of critical information infrastructure. An analysis of foreign and domestic standards, publications of specialized organizations, and scientific articles is conducted, which shows the variety of interpretations of the term advanced persistent threat. The conducted analysis allowed us to identify the characteristics inherent in APT attacks, distinguishing them from conventional cyberattacks, and also clarifies the goals of APT attacks. It was determined that the characteristics of such attacks include repeated pursuit of their goals over a long period of time,

adaptability to the efforts of the defending parties, and the desire to maintain the presence of covert channels (C2). It has been established that the goals of APT attacks include monitoring various activities in an organization's networks and information systems, stealing confidential information, and disrupting the organization's normal operations. This article analyzes the methods and approaches for detecting C2 channels and develops recommendations for countering advanced persistent threat attacks on IT facilities.

Keywords: APT attack, signs of APT attacks, C2 channels, methods for countering APT attacks, recommendations for countering APT attacks

For citation: Kozminykh, S.I. and Dubinin, A.Yu. (2026), "Directions and methods for detecting and countering APT attacks on information systems", *RSUH/RGGU Bulletin, "Information Science. Information Security. Mathematics" Series*, no. 3, pp. 94–113, DOI: 10.28995/2686-679X-2026-3-94-113

Введение

В последнее время все чаще в новостной повестке появляется информация об успешных атаках на крупные как зарубежные, так и российские компании, в том числе являющиеся субъектами критической информационной инфраструктуры (КИИ). В качестве самых свежих примеров можно привести атаку на Аэрофлот, о которой стало известно 28 июля 2025 года¹, или атаку на *Jaguar Land Rover (JLR)* 2 сентября 2025 года². Если Аэрофлот смог восстановить пострадавшую инфраструктуру на третий день, то *JLR* пришлось на месяц останавливать свое производство. Это только две атаки, которые на слуху у обывателя, никогда не сталкивающегося со сферой информационной безопасности. Обе эти атаки связаны с так называемыми *APT*-группировками. В первом случае злоумышленники сами заявили о своем противоправном деянии и целях, во втором случае – об этом сообщают эксперты в области информационной безопасности (ИБ). Ю.Ю. Шитова, анализируя динамику и основные тенденции кибератак в России за период с 2020 по 2025 год, подчеркивает, что целями сложных целевых атак (*APT*) становятся шпионаж, кража данных или

¹ Хакерская атака на «Аэрофлот» // Вести. URL: <https://www.vesti.ru/article/4618222> (дата обращения: 17.10.2025).

² *Jaguar Land Rover* оправилась после августовской хакерской атаки и наконец-то возобновляет работу // 3ДНьюс. URL: <https://3dnews.ru/1130423/jaguar-land-rover-opravilas-posle-avgustovskoy-hakerskoy-ataki-i-nakoneto-vozobnovlyaet-rabotu> (дата обращения: 17.10.2025).

диверсии, а последние годы «все чаще становятся элементом геополитического противостояния и промышленного шпионажа» [Шитова 2025, с. 98].

По данным Лаборатории Касперского в 2024 году 25% организаций во всем мире фиксировали атаки, которые управлялись непосредственно людьми (АРТ)³. На фоне понижения до 5% инцидентов высокой критичности от общего количества (специалисты связывают данное снижение с совершенствованием средств защиты, выполняющими функции без участия человека) доля АРТ-атак побила исторический максимум 2021 года в 41% и достигла 44% общего количества инцидентов высокой критичности.

По сообщению глобальной команды по реагированию на компьютерные инциденты (*Global Emergency Response Team, GERT*) АО «Лаборатория Касперского», 35% зафиксированных ими атак длились около одного месяца, а в среднем 254 дня. К сожалению, специалисты официально не классифицируют их как АРТ-атаки, но, опираясь на факт продолжительности как один из основных признаков, можно сделать вывод о том, что это были именно АРТ-атаки⁴.

Термин АРТ, ранее знакомый профессионалам, уже на слуху и у неприсяжных к сферам безопасности людей. Для проведения комплексного и всестороннего исследования в области терминологии ИБ мы не можем обойти стороной контекст (среду и условия) появления термина. Нам необходимо ответить на вопросы, что стало движителем и в чем была необходимость его внедрения, уточнить, является ли термин новым или заимствованным из смежной области. Формирование понятийного аппарата требует всесторонней оценки контекста использования данного термина.

Появление в профессиональной терминологии сферы информационной безопасности понятия *advanced persistent threat (APT)*, согласно исследователям *Zongyuan Xiang, Dong Guo, Qiang Li*, относится к 2006 году [Xiang et al. 2020]. Выделение АРТ среди общего количества атакующих по определенным признакам впервые мы наблюдаем в военно-научных кругах США. В своем отчете

³ Каждая четвертая компания в мире столкнулась с АРТ-угрозами в 2024 году // Kaspersky. URL: <https://www.kaspersky.ru/about/press-releases/kazhdaya-chetvyortaya-kompaniya-v-mire-stolknulas-s-apt-ugrozami-v-2024-godu> (дата обращения: 19.03.2025).

⁴ Каждая третья кибератака длилась в 2024 году больше месяца // Kaspersky. URL: <https://www.kaspersky.ru/about/press-releases/kazhdaya-tretya-kiberataka-dlilas-v-2024-godu-bolshe-mesyaca> (дата обращения: 19.03.2025).

«Победа в киберпространстве»⁵ для конференции ВВС США с одноименным названием⁶ доктор Ребекка Грант ссылается на выступление генерал-лейтенанта Роберта Дж. Элдера, командующего 8-й воздушной армией, на круглом столе группы военных писателей 13 июня 2007 года, где он приводит классификацию угроз Стратегического командования Вооруженных сил США, разделяя их на три уровня. К первой угрозе были причислены так называемые *kiddy hackers* – взламывающие сети, талантливые, но в основном аполитичные личности. Ко второму уровню отнесены угрозы от людей с более продвинутыми навыками, но зачастую с возможностями, не сопоставимыми с возможностями и ресурсами государств. И наконец, третий уровень включал в себя кибер-противников, равных Агентству Национальной Безопасности, с сопоставимыми возможностями и ресурсами государства за ними. Именно третий уровень угроз был назван *advanced persistent threat*.

Авторы статьи «*What is APT? Продвинутое атаки, хитрости и методы защиты*» указывают даже на более раннее появление термина *APT*. Уже в 2004 году *LM-CIRT (Lockheed Martin Critical Incident Response Team)* в лице команды по реагированию на компьютерные инциденты (*CSIRT*) в своих исследованиях ссылаются на данное явление⁷. В другой статье зарубежные исследователи связывают эту угрозу с наличием хорошо подготовленных и оснащенных противников, проводящих многолетние кампании по вторжению внутрь информационного периметра организаций, с целью добыть конфиденциальную экономическую, коммерческую информацию или иные данные, касающиеся национальной безопасности⁸. На их вооружении передовые инструменты и методы, используемые для обхода традиционных СЗИ.

⁵ Victory in cyberspace // Air & Space Forces Association. URL: <https://web.archive.org/web/20071011044002/https://www.afa.org/media/reports/victorycyberspace.pdf> (дата обращения: 15.03.2025).

⁶ Air Force Cyberspace Report // TaoSecurity. URL: <https://taosecurity.blogspot.com/2007/10/air-force-cyberspace-report.html> (дата обращения: 15.03.2025).

⁷ What is APT? Продвинутое атаки, хитрости и методы защиты // Хакер. URL: <https://xaker.ru/2018/07/20/wtf-is-apt/> (дата обращения: 15.03.2025).

⁸ Hutchins E.M., Cloppert M.J., Amin R.M. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. URL: <https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf> (дата обращения: 15.03.2025).

Сам термин *APT* (*advanced persistent threat*) широко используется в профессиональной⁹, научной¹⁰ и даже медийной среде¹¹, но при этом отсутствует строгое, единое и общепринятое определение в российском сегменте. Организации вкладывают в данный термин вариативный смысл: где-то определяют как тип атаки или характер угрозы, где-то классифицируют как модель поведения атакующих или деятельность конкретной группы. Данный момент обусловлен различием целей применения понятия, контекстов его использования и исследовательских подходов. Однако отсутствие единства в терминологии может приводить к путанице, осложняя восприятие литературы данной тематики и затруднять унификацию методов защиты. Более того, насущность вопроса уточнения терминологии сферы ИБ отмечена в установочных документах Совета Безопасности Российской Федерации. Среди основных направлений научных исследований в области обеспечения ИБ РФ Совет указывает на проблемы формирования понятийного (терминологического) аппарата в области ИБ¹².

Таким образом, цель данного исследования заключается в уточнении термина *APT* и производных от него понятий.

Задачи исследования:

- 1) проанализировать существующие трактовки понятий *APT*, *APT*-атака, *APT*-группировка;
- 2) определить критерии корректного употребления данных терминов;
- 3) предложить уточненные дефиниции, соответствующие современным реалиям сферы информационной безопасности;
- 4) определить существенные признаки *APT*-атак;
- 5) уточнить цели *APT*-атак;

⁹ Современные хакеры все чаще выбирают медиа как цель // Cnews. URL: https://safe.cnews.ru/news/line/2025-08-20_sovremennye_hakery_vse_chashche (дата обращения: 17.10.2025).

¹⁰ Целевые атаки // Сбербанк. URL: <https://www.sberbank.ru/ru/person/kibrary/vocabulary/celeveye-ataki> (дата обращения: 17.10.2025).

¹¹ Что такое АРТ-атака (*Advanced Persistent Threat* – продолжительная атака повышенной сложности)? // Kaspersky. URL: <https://www.kaspersky.ru/resource-center/definitions/advanced-persistent-threats> (дата обращения: 17.10.2025).

¹² Выписка из Основных направлений научных исследований в области обеспечения информационной безопасности Российской Федерации // Совет Безопасности РФ. URL: <http://www.scrf.gov.ru/security/information/document155/> (дата обращения: 10.02.2026).

- 6) проанализировать существующие направления и методы обнаружения *C2*-каналов, и разработать рекомендации по противодействию *advanced persistent threat* атакам на объектах информатизации.

Новизна исследования состоит в многоаспектном рассмотрении понятия *APT*-атака, обобщении существующих формулировок, а также в анализе направлений и методов обнаружения *APT*-атак и разработке рекомендаций по противодействию *APT*-атакам на объектах информатизации.

Определения и признаки APT-атак

Нормативная документация. Термин *APT* либо его полное наименование *advanced persistent threat* присутствует в небольшом количестве нормативно-правовых актов (НПА). Прежде всего это НПА страны происхождения термина (США). В стандартах он появляется в 2006 году в глоссарии используемых терминов Комитета по системам национальной безопасности, не подвергаясь изменениям ни в 2011 году, ни в актуальной версии от 2015 года¹³.

В 2011 году в *NIST SP 800-39* используется определение глоссария, согласно которому *APT (advanced persistent threat)* – это «противник, обладающий высоким уровнем знаний и значительными ресурсами»¹⁴. Стоит упомянуть, что целями такого противника, как правило, являются создание и расширение плацдармов в инфраструктуре информационных технологий намеченных организаций с целью хищения информации, подрыва или воспрепятствования нормальному функционированию критических программ или подразделений; или создание скрытых плацдармов, чтобы выполнить эти цели в будущем.

К признакам такого противника стандарт относит:

- 1) неоднократное преследование своих целей в течение длительного периода времени;
- 2) адаптируемость к усилиям обороняющихся сторон по противодействию ему;

¹³ National Information Assurance (IA) Glossary. URL: https://www.dni.gov/files/NCSC/documents/nittf/CNSSI-4009_National_Information_Assurance.pdf. (дата обращения: 02.11.2025).

¹⁴ Ross R. Managing Information Security Risk: Organization, Mission, and Information System View, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=908030 (дата обращения: 10.11.2025).

3) стремление поддерживать наличие скрытых каналов (C2), необходимых для выполнения своих целей [Дубинин 2025].

В Великобритании Национальный центр кибербезопасности (*National Cyber Security Centre, NCSC*) дает следующее определение: *advanced persistent threat* – актор, представляющий интересы государства, или преступник, обладающий высокой квалификацией, получающий несанкционированный доступ к системе и остающийся незамеченным в течение длительного времени¹⁵.

Канадский Центр Кибербезопасности¹⁶ (*Canadian Centre for Cyber Security, CCCS*) под *advanced persistent threats* понимает злоумышленников, обладающих высоким мастерством, которые способны задействовать передовые методы для проведения сложных и длительных кампаний для достижения своих целей. Это понятие обычно используется в контексте групп, представляющих интересы государства или высококвалифицированных организованных преступных группировок¹⁷.

Аналитические организации. В средствах массовой информации, коммерческих материалах, компаниях, профилирующихся в сфере информационной безопасности, понимание АРТ диверсифицированы.

Лаборатория Касперского дает следующее определение: АРТ – это сокращение от *advanced persistent threat*, то есть сложная постоянная угроза. Раскрывая понятие АРТ далее, специалисты высказывают предположение, что рассматривать АРТ можно с двух сторон: это и сама атака, и это люди, стоящие за ней. С одной стороны, сложная постоянная угроза (АРТ) является высокоточной кибератакой. С другой стороны, АРТ можно назвать группу, спонсируемую государством либо иным покровителем, оплачивающим целевую атаку¹⁸. Целью такой атаки называется машина (персональный компьютер), на которой присутствует действительно ценная для злоумышленников информация. Кроме того, в созданной Лабораторией Касперского энциклопедии термин АРТ «применяется к согласованным, скрытным, постоянным кибератакам на конкретные организации, в отличие от спекулятивных, единичных или продиктованных

¹⁵ Glossary // National Cyber Security Centre. URL: <https://www.ncsc.gov.uk/section/advice-guidance/glossary> (дата обращения: 04.11.2025).

¹⁶ An introduction to the cyber threat environment // Government of Canada. URL: <https://www.cyber.gc.ca/en/guidance/introduction-cyber-threat-environment> (дата обращения: 04.11.2025).

¹⁷ An introduction to the cyber threat environment // Government of Canada...

¹⁸ Что такое АРТ-атака...

случаем инцидентов, которые составляют основную долю криминальной активности»¹⁹. Отдельно специалистами выделяется понятие *APT*-атаки как «целевой продолжительной атаки повышенной сложности, задачей которой является обнаружение на устройстве пользователя секретной, конфиденциальной или любой ценной информации и использование ее в интересах киберпреступников»²⁰. В другой статье от Лаборатории Касперского понятие *APT*-атака раскрывается как «длительная, целенаправленная и скрытая кибератака, при которой злоумышленники получают доступ к информационным системам и остаются незамеченными, чтобы собрать конфиденциальные данные или нанести другой ущерб»²¹. Более того, отмечается, что *APT* является частным случаем целевых атак, которые в свою очередь «направлены на компрометацию определённой системы или объекта (этим они отличаются от массовых атак, также известных как нецелевые и «ковровые»)»²².

Словарь терминов *Positive Technologies* также не обошел стороной область *APT*. Специалистами выделяются понятия *APT*-атака и *APT*-группировка. Под *APT*-атакой они подразумевают «хорошо организованную, тщательно спланированную многоэтапную целевую кибератаку», а под *APT*-группировками понимаются киберпреступники, «обладающие значительными финансовыми ресурсами и техническими возможностями» и «отличающиеся высоким уровнем квалификации»²³.

Представитель кредитно-финансовой сферы Сбера акцентирует внимание на термине *APT*-атака: «Целевые атаки – это кибератаки, которые направлены на конкретную ИТ-систему или организацию. Их еще называют таргетированными, или *APT*-атаками»²⁴. В данной статье атаки рассматриваются как массовые и целевые.

¹⁹ Продолжительные атаки повышенной сложности (advanced persistent threats, APT). URL: <https://encyclopedia.kaspersky.ru/glossary/apt-advanced-persistent-threats/> (дата обращения: 16.11.2025).

²⁰ 5 признаков *APT*-атаки и советы по ее предотвращению. URL: <https://www.kaspersky.ru/resource-center/threats/advanced-persistent-threat> (дата обращения: 16.11.2025).

²¹ Что такое *APT*-атака (Advanced Persistent Threat – продолжительная атака повышенной сложности)?..

²² Целевые (таргетированные) атаки. URL: <https://encyclopedia.kaspersky.ru/glossary/targeted-attack/> (дата обращения: 16.11.2025).

²³ Словарь терминов ИБ. URL: <https://www.ptsecurity.com/research/glossary/?letter=a> (дата обращения: 22.11.2025).

²⁴ Целевые атаки // Сбербанк. URL: <https://www.sberbank.ru/ru/person/kibrary/vocabulary/celevye-ataki> (дата обращения: 17.10.2025).

К целевым относятся те, что направлены злоумышленниками на конкретные информационные инфраструктуры или компании.

Американская компания *Microsoft* определяет АРТ как «долгосрочную целенаправленную кибератаку, призванную проникнуть в организацию и оставаться незамеченной как можно дольше»²⁵. При осуществлении таких кибератак злоумышленники ориентируются не на массовость, а на конкретные высокодоходные организации, такие как государственные учреждения, субъекты критической инфраструктуры или крупные предприятия. Целями, преследуемыми злоумышленниками, являются:

- 1) отслеживание различных активностей в сетях и информационных системах организации;
- 2) кража конфиденциальной информации;
- 3) нарушение нормального функционирования организации.

Зачастую, согласно *Microsoft*, такие атаки злоумышленники проводят в течение нескольких месяцев или даже лет.

Специалисты *PaloAlto* определяют АРТ как сложную долгосрочную кибератаку, обычно осуществляемую высококвалифицированными киберпреступниками, часто при поддержке государств или организованных киберпреступных групп. В противовес обычным компьютерным атакам, целью которых зачастую является немедленный сбой в работе или получение финансовой выгоды, АРТ предназначены для длительного и скрытого нахождения в целевой сети или организации²⁶.

Отличительными характеристиками, по мнению специалистов, являются:

1. *Продвинутый уровень*. Злоумышленниками используются сложные методы проведения компьютерных атак, используются уязвимости нулевого дня, собственное разработанное вредоносное программное обеспечение (ВПО), различные тактики уклонения, чтобы обойти традиционные средства защиты информации (СЗИ).

2. *Постоянность*. Злоумышленники стараются сохранить присутствие в организации как можно дольше, соответственно и сама компьютерная атака продолжается в течение длительного времени.

²⁵ What is an advanced persistent threat (APT)? // Microsoft. URL: <https://www.microsoft.com/en-us/security/business/security-101/what-is-advanced-persistent-threat-apt> (дата обращения: 09.11.2025).

²⁶ What is an advanced persistent threat? // Paloaltonetworks URL: <https://www.paloaltonetworks.com/cyberpedia/what-is-advanced-persistent-threat-apt> (дата обращения: 10.11.2025).

3. *Целенаправленность*. В первую очередь целью АРТ-атак являются организации или отрасли с ценной информацией. Например, правительственные организации, объекты КИИ.

4. *Господдержка* либо собственная высокая организация.

Известный в российском медиапространстве интернет-портал *SecurityLab.ru* дает следующее определение АРТ в своем глоссарии: усовершенствованная постоянная угроза (*advanced persistent threat, АРТ*) – это скрытая угроза, за которой обычно стоит национальное государство или спонсируемая государством группировка, которая получает несанкционированный доступ к компьютерной сети и остается незамеченной в течение длительного периода времени»²⁷. В другой своей статье авторы под АРТ уже подразумевают «устойчивые целевые угрозы», а под АРТ-группировками понимаются «сплоченные команды специалистов, зачастую работающие при поддержке правительств или тесно связанных с ними структур»²⁸.

Научные публикации

Согласно исследователям *Nanda Rani, Bikash Saha, Sandeep Kumar Shukla*, АРТ-атаки – это серия атак, высокоорганизованных и часто спонсируемых государством групп, действующих скрытно и методично в течение длительного периода времени [Rani et al. 2025]. Данное определение находит поддержку в другой статье, в которой исследователи также отмечают схожие признаки акторов и их целей, а именно, относят их к организованным группам опытных злоумышленников, финансируемых какой-либо организацией или правительством, в целях получения важной информации; описывают АРТ как атаку, не похожую на обычную или атаку, совершаемую обычным злоумышленником. Кроме того, авторы утверждают, что АРТ – военный термин, адаптированный к сфере информационной безопасности [Alshamrani et al. 2019]. В то же время, по словам исследователей *Duraïd Thamer Salim, Manmeet Mahinderjit Singh* и *Pantea Keikhosrokiani*, АРТ – это актор, реализующий скрытые угрозы, путем получения нелегитимного доступа к сетевому устройству [Salim et al. 2023].

Отечественные исследователи также имеют различный взгляд как на понимание самого термина АРТ, так и его перевод на русский

²⁷ Глоссарий // Securitylab. URL: https://www.securitylab.ru/glossary/1642/page1_14.php (дата обращения: 22.11.2025).

²⁸ АРТ-группировки и глобальная кибервойна – кто против кого, как и почему // Securitylab. URL: <https://www.securitylab.ru/blog/personal/xiaomite-journal/356240.php> (дата обращения: 16.11.2025).

язык. Например, П.В. Закалкин указывает, что АРТ – это «тип сложных атак преимущественно на инфраструктуру государственных и военных объектов, целых отраслей и конкретных компаний» [Закалкин 2020, с. 306]. И.В. Савин в своей статье соглашается с переводом Лаборатории Касперского и утверждает, что АРТ – сложная постоянная угроза и «представляет собой нацеленную атаку, на какой-либо конкретный объект» [Савин 2017, с. 100]. А.О. Ермаков дает перевод АРТ как «развитая устойчивая угроза», а в качестве определения указывает следующее: «спланированные атаки, направленные против государственных структур с целью подрыва их авторитета как для своих граждан внутри страны, так и с целью формирования негативного политического имиджа на международной арене» [Ермаков и др. 2017, с. 25]. По мнению исследователя Е.А. Логинова, «одной из наиболее опасных угроз являются АРТ-группировки (*advanced persistent threats*), которые используют долгосрочную стратегию для получения несанкционированного доступа к важной информации или инфраструктуре и поддержания своего присутствия в системе» [Логинов 2025, с. 160]. Таким образом, автор приравнивает понятие угрозы (в контексте АРТ) к АРТ-группировкам, не конкретизируя сам термин АРТ, а также упуская возможную аффилированность таких группировок с государственными структурами.

А.Е. Грязнов и Ю.Д. Фот к важным признакам АРТ-атак относят многоэтапность, скрытность, использование уязвимостей [Грязнов, Фот 2025].

Проведя анализ зафиксированных государственными структурами определений и существующих взглядов на АРТ в исследовательской среде, нами составлена таблица, которая позволяет выделить основные признаки, указывающие на принадлежность компьютерной атаки к АРТ (табл. 1).

Таблица 1

Признаки целевых атак

Признак	Описание	Чем отличается от «обычных» атак
Целевая направленность	Атака направлена на конкретную организацию, отрасль или даже конкретных людей	Массовые атаки пытаются охватить как можно больший объем
Продолжительность	Кампания может длиться месяцами или годами	Типичные атаки – «ударил и ушел»

Окончание табл. 1

Признак	Описание	Чем отличается от «обычных» атак
Подготовка	Глубокая разведка: структура сети, сотрудники, подрядчики, технологии	В обычных атаках разведка минимальна либо автоматизирована
Сложная цепочка атаки	Используется <i>kill chain</i> из нескольких этапов	Простые атаки часто ограничены 1–2 этапами
Маскирование	Минимизация «шума», обход <i>SIEM/EDR</i> , использование легитимных инструментов в инфраструктуре цели	Обычные атаки легко поддаются обнаружению системами безопасности, основанными на сигнатурном либо поведенческом подходе
Уникальные или редкие инструменты	Используются адаптированные бэкдоры и модифицированные эксплойты под конкретную жертву	Зачастую используются распространенные и известные инструменты проведения атаки
Устойчивость в системе	Несколько точек закрепления в системе	В обычных атаках присутствует единственный механизм закрепления в системе
Ручное управление	Часть действий выполняется операторами вручную	В массовых атаках все автоматизировано
Конкретная цель	Шпионаж, саботаж, кража данных, диверсия	Чаще всего деньги или быстрый результат
Связь с организованным актором	Государства или хорошо финансируемые группы	Киберпреступники и скрипт-кидди (<i>kiddy hackers</i>)

Направления, методы обнаружения и рекомендации по противодействию advanced persistent threat атакам

Несмотря на различие как в русскоязычных переводах, так и в трактовках термина АРТ и его производных зарубежными коллегами, акцентируются прежде всего возможности злоумышленников незаметно и продолжительное время находиться в инфраструктуре атакуемых организаций. Данное свойство АРТ-атак прежде всего зависит от способности группировок выстраивать устойчивые каналы управления в целевой инфраструктуре (*command and control*, *C&C*, *C2*). Именно *C2*-каналы обеспечивают длительное нахождение

ние в инфраструктуре и возможность управления, эксфильтрацию данных, адаптацию атакующих к мерам защиты. В связи с этим выявление таких каналов в инфраструктуре является одним из приоритетов команды информационной безопасности организаций.

На современном этапе информационного противоборства атакующей и защищающейся сторон существующие методы и средства обнаружения, особенно в разрезе АРТ-группировок, являются явно недостаточными. Помимо техники маскирования под легитимные протоколы для построения С2-каналов актуальной тенденцией является и использование методов криптографической защиты информации [Дубинин 2025].

Анализ существующих распространенных методов по обнаружению, в том числе АРТ-атак, представлен в табл. 2.

Анализ наиболее часто используемых методов противодействия службами информационной безопасности организаций представлен в табл. 3.

Таблица 2

Анализ методов обнаружения АРТ-атак

Метод обнаружения	Описание	Преимущества	Ограничения
Сигнатурные методы	Использование сигнатур вредоносного ПО и известных С2-индикаторов (IP, домены, хэши)	Простота реализации, высокая точность для известных угроз	Низкая эффективность против полиморфизма, уникальных сборок и быстро меняющейся инфраструктуры
Поведенческий анализ	Анализ аномалий: сетевые соединения, процессы, <i>lateral movement</i> , использование EDR и NDR	Устойчивость к новым и неизвестным угрозам, выявление скрытой активности	Требует настройки <i>baseline</i> , возможны ложные срабатывания
Анализ сетевого трафика и С2-паттернов	Выявление <i>beaconing</i> , малых периодических соединений, шифрования, маскировки, анализ DGA и туннелирования	Эффективен для обнаружения С2-коммуникаций	Сложность анализа зашифрованного трафика, необходимость глубокой аналитики

Окончание табл. 2

Метод обнаружения	Описание	Преимущества	Ограничения
<i>Threat Intelligence (IOC/IOA)</i>	Использование внешних данных: <i>IOC</i> (IP, домены, хэши) и <i>IOA</i> (поведение)	Позволяет выявлять известные кампании и техники на ранних стадиях	Зависимость от актуальности и качества источников
Машинное обучение и корреляционный анализ	Кластеризация, выявление отклонений от <i>baseline</i> , корреляция событий в <i>SIEM/XDR</i>	Эффективен против <i>zero-day</i> атак и сложных <i>APT</i>	Высокая сложность внедрения, требования к данным и ресурсам

Таблица 3

Анализ методов противодействия *APT*-атакам

Метод	Описание	Основные меры	Эффект
Архитектурные меры	Построение защищенной архитектуры инфраструктуры	Сегментация сети, принцип наименьших привилегий, архитектура нулевого доверия	Ограничение распространения атакующего, усложнение создания устойчивого <i>C2</i>
Защита и разрушение <i>C2</i> -каналов	Противодействие каналам управления и контроля атакующего	Блокировка <i>C2</i> , фильтрация <i>DNS</i> , <i>TLS</i> -инспекция, обнаружение <i>beaconing</i> и установленных туннелей, контроль исходящего трафика	Нарушение управления атакой, снижение эффективности <i>APT</i>
Active Defense и Threat Hunting	Активный поиск скрытых угроз в инфраструктуре	Ручной анализ системных журналов, гипотезно-ориентированный анализ	Раннее выявление <i>APT</i> до нанесения ущерба
Реагирование и изоляция	Меры при обнаружении атаки	Изоляция хоста, блокировка учетных записей, блокировка <i>C2</i> , анализ артефактов	Быстрое сдерживание атаки и ее деградация

При существующих методах обнаружения и противодействия С2-каналам в инфраструктуре управления и контроля защита сталкивается с рядом проблем. Во-первых, использование легитимных сервисов (например, облаков) в качестве С2-серверов управления²⁹.

Во-вторых, шифрование трафика. Более того, чтобы средства обнаружения вторжений могли расшифровывать сетевой трафик, необходимо реализовать инфраструктуру публичных ключей, распространить на рабочие станции политики доверия к развернутым центрам сертификации, возникают не всеми производителями решенные проблемы с расшифровкой *TLS1.3* трафика в части получения значения из поля *SNI*. В-третьих, бесфайловое вредоносное ПО (*fileless malware*) – ВПО, не сохраняющее исполняемые файлы на жестком диске ПК и выполняющее свои действия в оперативной памяти легитимными средствами операционной системы (ОС)³⁰. И наконец, низкий уровень «шума» АРТ-атак, так называемые *low-and-slow* атаки.

На основе проведенного анализа можно дать следующие рекомендации по противодействию АРТ на объектах информатизации. Прежде всего необходимо определить основные направления выявления и противодействия этим атакам, которые должны включать следующие взаимосвязанные компоненты:

Создание модели угроз, учитывающей специфику АРТ, в том числе длительность присутствия нарушителя в информационном контуре организации, использование легитимных инструментов операционных систем и построение скрытых каналов связи в инфраструктуре управления и контроля.

Определение перечня отслеживаемых событий и источников телеметрии (сетевой трафик, журналы безопасности, события ОС, активность учетных записей, в том числе и привилегированных, активность приложений).

Внедрение механизмов комплексного обнаружения, включающих сигнатурный анализ, поведенческую аналитику, выявление аномалий.

Обнаружение С2-коммуникаций.

Использование данных *threat intelligence* для актуализации индикаторов компрометации и индикаторов атак, используемых

²⁹ GitHub вместо С2-сервера. Одна небрежная LNK-метка помогла связать серию кибератак на дипломатов в России // Securitylab URL: <https://www.securitylab.ru/news/565942.php> (дата обращения: 09.04.2026).

³⁰ Бесфайловое вредоносное ПО (Fileless malware) // Сбербанк. URL: <https://www.sberbank.ru/ru/person/kibrary/vocabulary/fileless-malware> (дата обращения: 09.04.2026).

APT-группировками для построения инфраструктуры управления и контроля.

Реализация мер, направленных на ограничение и контроль исходящих соединений, включая отслеживание *DNS*-запросов, использование шифрованных протоколов и выявление попыток обхода средств защиты информации.

Представленные рекомендации, на наш взгляд, позволят достаточно эффективно противостоять *APT*-атакам на объектах информатизации.

Заключение

Согласно проведенному анализу, термин *APT* имеет многообразную трактовку. С одной стороны, под *APT* понимают атаку, характеризующуюся определенными этапами и признаками, с другой стороны, актора, воплощенного или в криминальных группировках, или структурах, аффилированных с определенным государством. Однако существующие интерпретации термина *APT* требуют критической оценки из-за многоаспектности данного явления. На наш взгляд, термин *APT* требует унификации, и под ним следует понимать именно группу лиц, обладающую навыками и нацеленную на коммерческие организации, государственные учреждения в целях извлечения финансовой прибыли или нанесения серьезного ущерба. Под *APT*-атакой можно понимать целенаправленную, продолжительную во времени компьютерную атаку, осуществляемую высококвалифицированным и высокомотивированным противником (актором). К основным признакам *APT*-атак, что и является их отличительной чертой в сравнении с обычными атаками, мы выделяем целевую направленность, продолжительность во времени, наличие этапа подготовки и сложной цепочки атаки, маскирование, уникальные или редкие инструменты, ручное управление, конкретная цель и связь с организованным актором. Представленные рекомендации, предложенные направления и методы обнаружения и противодействия *APT*-атакам включают в себя создание модели угроз, формирование необходимого перечня отслеживаемых событий информационной безопасности, внедрение механизмов комплексного обнаружения. Отдельно необходимо сосредоточиться на обнаружении *C2*-каналов, так как они являются ключевым инструментом атакующей стороны по контролю над заражёнными системами и позволяют им извлекать информацию.

Литература

- Грязнов, Фот 2025 – *Грязнов А.Е., Фот Ю.Д.* Методы защиты объектов КИИ от АРТ-атак // Актуальные вопросы обеспечения комплексной безопасности: материалы национальной научно-практической конференции с международным участием, посвященной 35-летию МЧС России и 95-летию Оренбургского ГАУ, Оренбург, 23 мая 2025 года. Оренбург: Оренбургский государственный аграрный университет, 2025. С. 110–114. EDN RSFFKB.
- Дубинин 2025 – *Дубинин А.Ю.* Современные тенденции построения каналов инфраструктуры управления и контроля (С2) // Цифровые инструменты обеспечения устойчивого развития экономики и образования: новые подходы и актуальные проблемы: сб. науч. трудов IV-й Национальной научно-практической конференции (с международным участием), Орёл, 23 апреля 2025 года. Орел: Российская академия народного хозяйства и государственной службы при Президенте РФ, 2025. С. 190–195.
- Ермаков и др. 2017 – *Ермаков А.О., Кавешников М.Б., Клянчина Е.В.* Вредоносное программное обеспечение АРТ-групп и его характеристики // Вопросы кибербезопасности. 2017. № S2 (20). С. 24–29. EDN ZVZGHP.
- Закалкин 2020 – *Закалкин П.В.* АРТ-атаки как средство воздействия на критическую инфраструктуру // Актуальные проблемы защиты и безопасности: Труды XXIII Всероссийской научно-практической конференции РАРАН, Санкт-Петербург, 1–4 апреля 2020 года. Т. 1. СПб.: Российская академия ракетных и артиллерийских наук, 2020. С. 306–309. EDN ANPJZH.
- Логинов 2025 – *Логинов Е.А.* Применение модели Марковских цепей для предсказания движений АРТ-группировок на основе Threat Intelligence-данных // Международный журнал информационных технологий и энергоэффективности. 2025. Т. 10, № 5 (55). С. 160–164. EDN UQYWEO.
- Савин 2017 – *Савин И.В.* АРТ-атаки. Анализ и защита // Вестник научных конференций. 2017. № 10-1 (26). С. 100–102. EDN YMVUZT.
- Шитова 2025 – *Шитова Ю.Ю.* Анализ динамики киберугроз в России // Развитие интеллектуальной экономики и промышленности на основе искусственного интеллекта: Монография. СПб.: ПОЛИТЕХ-ПРЕСС, 2025. С. 98–122. DOI: 10.18720/IEP/2025.3/5. EDN VVQERK.
- Alshamrani et al. 2019 – *Alshamrani A., Myneni S., Chowdhary A., Huang D.* A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities // IEEE Communications Surveys & Tutorials. 2019. Vol. 21, no. 2. P. 1851–1877. DOI: 10.1109/COMST.2019.2891891.
- Rani et al. 2025 – *Rani N., Saha B., Shukla S.K.* A comprehensive survey of automated advanced persistent threat attribution: taxonomy, methods, challenges and open research problems // Journal of Information Security and Applications. 2025. Vol. 92. P. 104076. DOI: 10.1016/j.jisa.2025.104076.
- Salim et al. 2023 – *Salim D., Mahinderjit Singh M., Keikhosrokiani P.* A systematic literature review for APT detection and Effective Cyber Situational Awareness (ECSA)

conceptual model // *Heliyon*. 2023. Vol. 9, issue 7. P. e17156, DOI: 10.1016/j.heliyon.2023.e17156.

Xiang et al. 2020 – *Xiang Z., Guo D., Li Q.* Detecting mobile advanced persistent threats based on large-scale DNS logs. *Computers & Security*. 2020. Vol. 96. P. 101933. DOI: <https://doi.org/10.1016/j.cose.2020.101933>.

References

- Alshamrani, A., Myneni, S., Chowdhary, A. and Huang, D. (2019), “A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851–1877, 2019, DOI: 10.1109/COMST.2019.2891891.
- Dubinin, A.Yu. (2025), “Current trends in building command and control (C2) infrastructure channels”, *Tsifrovyye instrumenty obespecheniya ustoychivogo razvitiya ekonomiki i obrazovaniya: novyye podkhody i aktual'nyye problemy* [Digital tools for ensuring sustainable development of the economy and education: new approaches and current issues], *Proc. of the 4th Nat. Sci. and Pract. Conf. (with International Participation)*, Orel, 23 April 2025, Russian Presidential Academy of National Economy and Public Administration, Orel, Russia, pp. 190–195.
- Ermakov, A.O., Kaveshnikov, M.B. and Klyanchina, E.V. (2017), “Malware of APT groups and its characteristics”, *Cybersecurity Issues*, vol. S2 (20). pp. 24–29.
- Gryaznov, A.E. and Fot, Yu.D. (2025), “Methods of protecting critical infrastructure facilities from APT attacks”, *Current issues in ensuring integrated security. Proceedings of the national scientific and practical conference with international participation, dedicated to the 35th anniversary of the Russian Ministry of emergency situations and the 95th anniversary of the Orenburg State Agrarian University*, Orenburg, 23 May 2025, Orenburg State Agrarian University, Orenburg Russia, pp. 110–114.
- Loginov, E.A. (2025), “Application of Markov chain models to predict APT group movements based on threat intelligence”, *International Journal of Information Technology and Energy Efficiency*, vol. 10, no. 5 (55). pp. 160–164.
- Rani, N., Saha, B. and Shukla, S.K. (2025), “A comprehensive survey of automated advanced persistent threat attribution: taxonomy, methods, challenges and open research problems”, *Journal of Information Security and Applications*, vol. 92, p. 104076, DOI: 10.1016/j.jisa.2025.104076.
- Salim, D., Mahinderjit Singh, M. and Keikhosrokiani, P. (2023), “A systematic literature review for APT detection and Effective Cyber Situational Awareness (ECSA) conceptual model”, *Heliyon*, vol. 9, issue 7, p. e17156, DOI: 10.1016/j.heliyon.2023.e17156.
- Savin, I.V. (2017), “APT attacks. Analysis and defense”, *Bulletin of Scientific Conferences*, vol. 10-1 (26), pp. 100–102.
- Shitova, Yu.Yu. (2025), “Development of the intellectual economy and industry based on artificial intelligence”, *Analiz dinamiki kiberugroz v Rossii* [Analysis of cyber

threat dynamics in Russia], POLYTECH-PRESS St. Petersburg, Russia, pp. 98–122. DOI: 10.18720/IEP/2025.3/5.

Xiang, Z., Guo, D. and Li, Q. (2020), “Detecting mobile advanced persistent threats based on large-scale DNS logs”, *Computers & Security*, vol. 96, p. 101933, DOI: <https://doi.org/10.1016/j.cose.2020.101933>.

Zakalkin, P.V. (2020), “APT Attacks as a Means of Influencing Critical Infrastructure”, *Current Issues of Defense and Security. Proceedings of the 23rd All-Russian scientific and practical conference of the Russian academy of missile and artillery sciences*, St. Petersburg, 1–4 April 2020, Russian Academy of Missile and Artillery Sciences, St. Petersburg, Russia, vol. 1, pp. 306–309.

Информация об авторах

Сергей И. Козьминых, доктор технических наук, доцент, МИРЭА–Российский технологический университет, Москва, Россия; 119454, Россия, Москва, пр-кт Вернадского, д. 78;

Российский экономический университет им. Г.В. Плеханова, Москва, Россия; 115054, Россия, Москва, Стремянный пер., д. 36;

Финансовый университет при Правительстве Российской Федерации, Москва, Россия; 125167, Россия, Москва, Ленинградский пр-кт, д. 49/2; kozminykh@mirea.ru

Александр Ю. Дубинин, аспирант, Финансовый университет при Правительстве Российской Федерации, Москва, Россия; 125167, Россия, Москва, Ленинградский пр-кт, д. 49/2; 249426@edu.fa.ru

Information about the authors

Sergei I. Kozminykh, Dr. of Sci. (Mechanical Engineering), associate professor, MIREA-Russian Technological University, Moscow, Russia; 78, Vernadsky Av., Moscow, 119454, Russia;

Plekhanov Russian University of Economics, Moscow, Russia; 36, Stremyanny Lane, Moscow, 115054, Russia;

Financial University under the Government of the Russian Federation, Moscow, Russia; 49/2, Leningradsky Av., Moscow, 125167, Russia; kozminykh@mirea.ru

Aleksandr Yu. Dubinin, postgraduate student, Financial University under the Government of the Russian Federation, Moscow, Russia; 49/2, Leningradsky Av., Moscow, 125167, Russia; 249426@edu.fa.ru

Архитектура отказоустойчивой аутентификации через внешних поставщиков идентификации в критичных приложениях

Джахед Рахмани

Московский технический университет
связи и информатики (МТУСИ), Россия, Москва, *jahed@mail.ru*

Анна В. Сугрובה

Московский технический университет
связи и информатики (МТУСИ), Россия, Москва,
a.v.sugrobova@edu.mtuci.ru

Аннотация. В статье исследуются особенности применения внешних поставщиков идентификации (*Identity Providers, IdP*) в системах критической информационной инфраструктуры (КИИ), где отказ или компрометация механизма аутентификации напрямую влияет на доступность и безопасность технологических процессов. Целью исследования является разработка и экспериментальная валидация рискориентированной архитектуры аутентификации на основе протоколов *OAuth 2.0* и *OpenID Connect*, обеспечивающей устойчивость к отказам внешнего поставщика идентификации, безопасность обработки токенов, соблюдение требований приватности и непрерывность функционирования критичного приложения. Методология исследования включает моделирование угроз, анализ протокольных уязвимостей, а также разработку и экспериментальную валидацию архитектурного решения на лабораторном стенде. Стенд включает клиентское приложение, шлюз аутентификации, локальный *IAM*-контур и эмулятор внешнего *IdP* с управляемыми сценариями отказа (*S1–S7*). В ходе экспериментов моделировались сценарии деградации, полной недоступности и компрометации *IdP*, а также оценивались показатели доступности, времени восстановления (*RTO, Recovery Time Objective*) и задержек обработки запросов. Предложена архитектура, основанная на принципе недоверия к токену, централизованной валидации и использовании управляемых резервных режимов (*fallback, step-up, break-glass, BCP*). Разработана верификационная модель угроз и система количественных метрик. Результаты показывают возможность сохранения доступа при деградации *IdP* и снижение риска компрометации токенов. Практическая значимость заключается в формировании воспроизводимого подхода к проектированию аутентификации в критичных системах.

Ключевые слова: единый вход, SSO, OAuth2, OpenID Connect, поставщик идентификации, отказоустойчивость, резервный режим, модель угроз, приватность, критическая информационная инфраструктура

Для цитирования: Рахмани Д., Сугрובה А.В. Архитектура отказоустойчивой аутентификации через внешних поставщиков идентификации в критичных приложениях // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 114–140. DOI: 10.28995/2686-679X-2026-3-114-140

Fault-tolerant authentication architecture through external identity providers in critical applications

Dzhakhed Rakhmani

*Moscow Technical University of Communications
and Informatics (MTUCI), Moscow, Russia, jahed@mail.ru*

Anna V. Sugrobova

*Moscow Technical University of Communications
and Informatics (MTUCI), Moscow, Russia,
a.v.sugrobova@edu.mtuci.ru*

Abstract. This paper examines the use of external identity providers (IdPs) in critical information infrastructure systems (CII), where failures or compromises of authentication mechanisms directly affect system availability and security. The objective of the study is to develop and experimentally validate a risk-based authentication architecture using OAuth 2.0 and OpenID Connect protocols that ensures resilience to external identity provider failures, secure token processing, compliance with privacy requirements, and continuity of a mission-critical application. The research methodology includes threat modeling, protocol vulnerability analysis, and experimental validation of the proposed architecture on a laboratory testbed. The test environment consists of a client application, an authentication gateway, a local IAM system, and an external IdP emulator with controlled failure scenarios (S1-S7). The experiments simulate degradation, complete unavailability, and compromise of the IdP, while evaluating key metrics such as availability, recovery time (RTO), and request latency. The proposed architecture is based on the principle of zero trust towards tokens, centralized validation, and controlled fallback mechanisms (fallback, step-up, break-glass, BCP). A threat verification model and quantitative evaluation metrics are introduced. The results demonstrate the feasibility of maintaining system access during IdP degradation and reducing token-related risks. The practical contribution lies in providing a reproducible approach to authentication design in mission-critical systems.

Keywords: single sign-on, SSO, OAuth2, OpenID Connect, identity provider, fault tolerance, fallback mechanisms, threat model, privacy, critical information infrastructure

For citation: Rahmani, D. and Sugrobova, A.V. (2026), "Fault-tolerant authentication architecture through external identity providers in critical applications", *RSUH/RGGU Bulletin. "Information Science. Information Security. Mathematics" Series*, no. 3, pp. 114–140, DOI: 10.28995/2686-679X-2026-3-114-140

Введение

Механизмы единого входа (*SSO – Single Sign-On*) стали стандартом для современных информационных систем, обеспечивая удобство пользователей и снижение нагрузки на службы поддержки [Демидова, Жуков 2020; Старосельский, Жиренкин 2023]. Однако при их применении в критической информационной инфраструктуре возникает зависимость от внешнего поставщика идентификации [Рахмани 2022; Докучаев и др. 2025].

В этих условиях внешний *IdP* становится частью критического контура: его отказ или компрометация напрямую влияет на доступность системы. Следовательно, архитектура аутентификации должна обеспечивать не только безопасность, но и непрерывность функционирования.

1. Постановка задачи и метод исследования

Объектом исследования является подсистема аутентификации и авторизации критической информационной системы, интегрированной с внешним поставщиком идентификации (*IdP*). Предмет исследования – принципы построения риск-ориентированной архитектуры *SSO* на основе протоколов *OAuth 2.0* и *OpenID Connect*¹ (далее *OIDC*), включая управление жизненным циклом токенов, обеспечение приватности и реализацию механизмов отказоустойчивости (*RFC 6749*²).

¹ OpenID Connect Core 1.0 incorporating errata set 2. URL: https://openid.net/specs/openid-connect-core-1_0.html (дата обращения: 11.05.2026).

² RFC 6749. The OAuth 2.0 Authorization Framework. URL: <https://www.rfc-editor.org/info/rfc6749> (дата обращения: 11.05.2026).

Целью исследования является разработка и экспериментальная валидация рискориентированной архитектуры аутентификации на основе протоколов *OAuth 2.0* и *OpenID Connect*, обеспечивающей устойчивость к отказам внешнего поставщика идентификации, безопасность обработки токенов, соблюдение требований приватности и непрерывность функционирования критичного приложения.

Для достижения цели решаются следующие задачи: формирование модели угроз, анализ потоков токенов и пользовательских сессий, идентификация сценариев отказа *IdP*, разработка резервных механизмов, определение метрик оценки и проведение экспериментальной валидации.

1.1. Классификация операций и ролей

Ролевая модель и классификация операций, используемые в работе, представлены в табл. 1 и 2.

Таблица 1

Роли пользователей и критичность операций

Роль	Тип субъекта	Пример операций	Уровень критичности	Требования к непрерывности
Оператор	Интер-активный пользователь	Мониторинг системы, просмотр данных, запуск стандартных процедур	Уровень 1	Допускается работа через резервные механизмы
Администратор	Интер-активный пользователь	Изменение конфигурации, управление учетными записями, настройка политик безопасности	Уровень 3	Требуется усиленная аутентификация
Сервисная учетная запись	Автоматический сервис	Межсервисное взаимодействие, интеграционные API-запросы	Уровень 2	Только через внешний провайдер

Таблица 2

Классификация операций

Уровень	Тип операции	Примеры
Уровень 1	Наблюдение и чтение	Мониторинг, просмотр журналов
Уровень 2	Штатные изменения	Обновления параметров
Уровень 3	Критические операции	Изменение конфигурации

2. Моделирование угроз и анализ рисков

Разработка модели угроз является основой построения защищенной архитектуры [Иванов, Капгер, Шабуров 2023; Валеев, Кондратьева 2023а]. В условиях использования внешнего *IdP* необходимо учитывать не только классические атаки на протоколы, но и риски недоступности провайдера и угрозы цепочки поставок (*RFC 9700*³) [Докучаев и др. 2025].

2.1. Атакующие и их возможности

В модели угроз учитываются внешние атакующие, злоумышленники, использующие социальную инженерию, компрометированные клиентские окружения, инсайдеры, а также сценарии компрометации или ошибочной конфигурации *IdP*. Основные возможности нарушителей включают перехват *authorization code*, *MITM*-атаки⁴, *DNS*-подмену⁵, кражу *refresh*-токенов, эксплуатацию ошибок *redirect URI*, *state/nonce*, *audience* и *issuer*, а также злоупотребление легитимными правами доступа.

2.2. Угрозы по цепочке SSO

Детальный анализ протоколов *OAuth2* и *OIDC* в контексте их применения в критичных системах позволяет выделить следующие критические точки, требующие особого внимания при проек-

³ RFC 9700. Best Current Practice for OAuth 2.0 Security. URL: <https://www.rfc-editor.org/info/rfc9700> (дата обращения: 11.05.2026).

⁴ *MITM* – *Man-in-the-Middle*, атака посредника или атака «человек посередине».

⁵ *DNS* – *Domain Name System*, система доменных имен.

тировании защищенной архитектуры (RFC 6749, *OpenID Connect*, *OpenID Connect Discovery*⁶):

- Компрометацию токенов: перехват и повторное использование (*replay*), особенно для *refresh*-токенов.
- Ошибки валидации *JWT*⁷: отсутствие проверки подписи, *aud* и *iss*.
- Невалидированные параметры протокола: игнорирование *state* и *nonce* (*CSRF*⁸, *replay*).
- Утечки через логи и телеметрию: запись токенов и персональных данных.
- Уязвимости *redirect URI*: перехват *authorization code* через *open redirect*.

2.3. Сценарии сбоя поставщика идентификации

Ключевым условием обеспечения отказоустойчивости выступает формализация аварийных сценариев, связанных с недоступностью или компрометацией внешнего *IdP*. В рамках исследования идентифицированы следующие критические сценарии (обозначенные как *S1-S7*), каждый из которых требует специфических механизмов реагирования:

- *S1*: полная недоступность *IdP* – невозможность новых сессий при сохранении активных.
- *S2*: деградация – рост задержек и риск каскадного отказа приложения.
- *S3*: сбой эмиссии токенов – невозможность обновления сессий.
- *S4*: недоступность *JWKS*⁹ – невозможность проверки подписи токенов.
- *S5*: компрометация *IdP* – выпуск валидных, но недоверенных токенов.
- *S6*: изменение структуры *claims* – нарушение маппинга ролей.
- *S7*: блокировка пользователей – потеря доступа к системе.

⁶ OpenID Connect Discovery 1.0 incorporating errata set 2. URL: https://openid.net/specs/openid-connect-discovery-1_0.html (дата обращения: 11.05.2026).

⁷ JWT – *JSON Web Token*, JSON-веб-токен.

⁸ *CSRF* – *Cross-Site Request Forgery*, межсайтовая подделка запроса.

⁹ *JWKS* – *JSON Web Key Set*, набор JSON-веб-ключей.

2.4. Сводная верификационная таблица угроз

Для обеспечения проверяемости модели угроз и возможности аудита защитных механизмов все идентифицированные угрозы сведены в единую верификационную таблицу (табл. 3). Для каждой угрозы определены: поражаемый актив, потенциальные последствия, конкретный метод контроля и способ верификации (как именно проверяется наличие и работоспособность защиты).

Таблица 3

Верификационная матрица угроз и контрмер

Угроза	Актив	Последствия	Контроль	Проверка
Перехват <i>authorization code</i>	Код авторизации	Захват пользовательской сессии	<i>PKCE</i>	Тест <i>replay</i> -атаки
Подмена <i>redirect URI</i>	Код авторизации	Утечка токена	<i>Allowlist redirect URI</i>	Тест <i>Open Redirect</i>
Подмена токена	Пользовательская сессия	Несанкционированный доступ	Проверка подписи <i>JWT</i>	Тест невалидной подписи
Несовпадение <i>audience</i>	Токен доступа	Использование токена другим сервисом	Проверка <i>aud</i>	Интеграционный тест
Утечка токена через логи	Токен доступа	Компрометация аккаунтов	Маскирование логов	Аудит логов

3. Архитектура SSO для критичного приложения

Предлагаемая архитектура (рис. 1) основана на принципе недоверия к токену и использует многоуровневую защиту, обеспечивающую безопасность и отказоустойчивость [Кирильчук, Аблитаров 2024; Валеев, Кондратьева 2023а; Валеев, Кондратьева 2023б]. Внешний *IdP* рассматривается как потенциально ненадежный компонент, что требует внедрения механизмов контроля и резервирования.

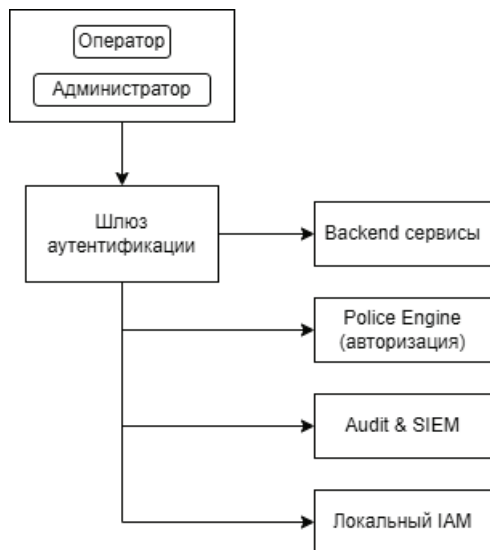


Рис. 1. Архитектура контура аутентификации

Архитектура включает шлюз аутентификации для централизованной проверки токенов и локальный *IAM*-контур¹⁰, используемый в резервных режимах [Барабанов, Макрушин 2020; Зимина, Лапоница 2023].

3.1. Компоненты контура

Архитектура включает клиентское приложение, шлюз аутентификации, *backend*-сервисы, модуль авторизации, систему аудита, внешний *IdP* и локальный *IAM*-контур. Шлюз выполняет централизованную проверку токенов, *backend*-сервисы используют результаты валидации, а локальный *IAM* применяется в резервных режимах.

3.2. Принципы

Архитектура основана на принципах нулевого доверия к токеном, минимизации времени его действия, разделения аутентификации

¹⁰ *Identity and Access Management* – управление идентификацией и доступом.

ции и авторизации, а также полного аудита действий. Внешний *IdP* рассматривается как потенциально ненадежный компонент.

3.3. Токены и сессии

Последовательность взаимодействия компонентов при использовании *Authorization Code Flow* с *PKCE* показана на рис. 2.

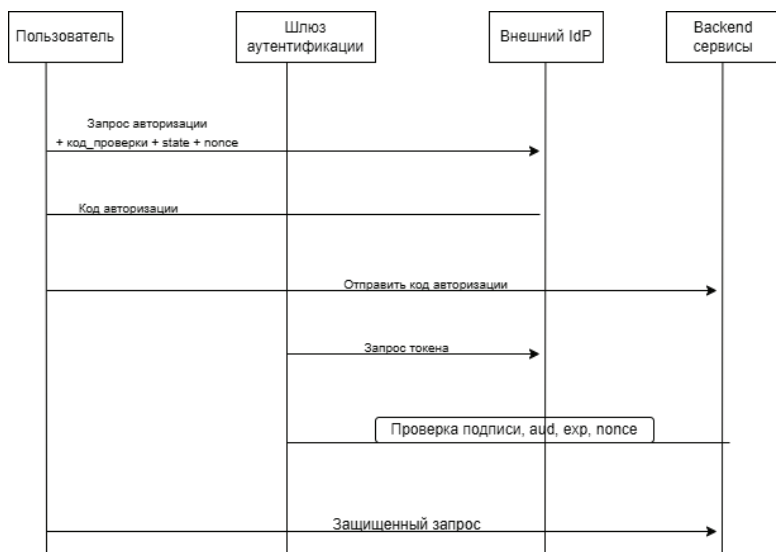


Рис. 2. Диаграмма последовательности
OAuth2 Authorization Code Flow с *PKCE*

Реализация архитектуры требует формализации политики управления токенами и пользовательскими сессиями. Ключевые требования включают ограничение времени жизни токена доступа до минимально достаточного уровня (для интерактивных сессий – 15...30 минут), что снижает риск его компрометации без ухудшения пользовательского опыта.

Для сервисных взаимодействий допускается увеличение времени жизни токена при условии применения дополнительных ограничений: привязки к сетевому контуру, использования взаимной *TLS*-аутентификации и контроля частоты запросов. Использование *refresh*-токенов минимизируется; при необходимости их применения они хранятся исключительно на сервере.

ной стороне, используются с ротацией и привязкой к клиенту и устройству.

Валидация *JWT* включает обязательную проверку подписи, *issuer*, *audience*, срока действия и временных ограничений (*exp*, *nbf*, *iat*), а также корректности идентификатора субъекта (*sub*). Для публичных клиентов применяется *PKCE*, а параметры *state* и *nonce* используются для защиты от атак повторного воспроизведения и *CSRF*.

В рамках предложенной архитектуры исключаются небезопасные практики: передача токенов через *URL*, их логирование в открытом виде, хранение в небезопасных клиентских хранилищах и использование без проверки аудитории.

Для обеспечения отказоустойчивости реализуется политика работы с *JWKS* (*RFC 7517*¹¹, *OpenID Connect Discovery*): ключи кэшируются с контролируемым *TTL*, при недоступности *endpoint* используется кэш, а при отсутствии необходимого ключа применяется стратегия *fail-closed*.

3.4. Разделение интерактивных пользователей и сервисных учетных записей

В КИИ необходимо разграничение интерактивных пользователей и сервисных учетных записей. Для сервисных учетных записей применяется *Client Credentials Flow* с взаимной *TLS*-аутентификацией, ограниченным временем жизни токенов, их ротацией и обязательным аудитом. В аварийных режимах допускаются только ограниченные операции.

4. Резервный режим и аварийный доступ

Одним из ключевых результатов работы является формализация управляемого резервного режима при недоступности или компрометации внешнего *IdP*. Резервный режим рассматривается не как ослабленный путь входа, а как заранее спроектированная, ограниченная и верифицируемая процедура восстановления доступа.

¹¹ RFC 7517. JSON Web Key (JWK). URL: <https://www.rfc-editor.org/rfc/rfc7517.html> (дата обращения: 11.05.2026).

4.1. Идея управляемого резервного режима

Резервный режим не должен снижать общий уровень защиты. Его реализация должна основываться на функциональной ограниченности, контролируемой активации по формальным триггерам, усиленной верификации и полном аудите всех действий.

4.2. Варианты резервного режима

- Для разных сценариев отказа предлагается иерархия из четырех механизмов:
- Локальная учетная запись с минимальными привилегиями – используется при недоступности внешнего провайдера.
- Усиленная аутентификация (*step-up*) – применяется для критических операций при штатной работе *IdP* [Саломатин, Исаков 2020; Шарипов, Олин, Габдрахманов 2025] (NIST SP 800-63B-4¹²).
- Аварийный доступ (*break-glass*) – регламентированная процедура для экстренных случаев.
- *BCP*¹³-режим – ограниченный набор функций, доступных при длительном сбое *IdP*.

4.3. Критерии приемлемости резервного режима

Формальные критерии приемлемости резервных механизмов представлены в табл. 4.

¹² NIST Special Publication 800-63B. URL: <https://pages.nist.gov/800-63-4/sp800-63b.html> (дата обращения 11.05.2026).

¹³ *BCP* (*Business Continuity Plan*) – план обеспечения непрерывности бизнеса.

Таблица 4

Формальные критерии
приемлемости резервных механизмов

Параметр	Локальная учетная запись	Усиленная аутентификация	Аварийный доступ	BCP-режим
Триггеры активации	Недоступность внешнего провайдера в течение $T > 5 \text{ мин}$, $success_rate < 95\%$ за 10 мин	Критичность операции / обнаружение аномалий	Экстраординарная ситуация, подтвержденная двумя администраторами	Затяжной сбой внешнего провайдера $> 30 \text{ мин}$, официальное решение руководителя смены
Субъекты, имеющие право активации	Все пользователи	Все пользователи	Только уполномоченные лица из списка «аварийные администраторы»	Все пользователи, включенные в BCP-список
Необходимые подтверждения	Автоматическая проверка системой	Дополнительный фактор	Два подтверждения от разных администраторов	Письменное распоряжение
Перечень доступных операций	Только уровень 1	Полный перечень, но каждое критическое действие требует отдельного подтверждения	Расширенный, но с аудитом каждого действия	Только критически важные операции, утвержденные в BCP-плане
Предельная длительность сессии	До восстановления внешнего провайдера, но не более 8 ч	Время выполнения операции	Строго ограничено: 30...60 мин, автоматическое завершение по таймеру	На весь период сбоя, но с переутверждением каждые 24 ч

Окончание табл. 4

Параметр	Локальная учетная запись	Усиленная аутентификация	Аварийный доступ	BCP-режим
Требования к хранению учетных данных	Пароли хранятся в защищенном хранилище, выдаются заранее в запечатанном конверте	Факторы привязываются к учетной записи локально, не зависят от <i>IdP</i>	Ключи доступа хранятся в сейфе, вскрытие фиксируется	Учетные данные для <i>BCP</i> хранятся отдельно от штатных
Уровень детализации аудита	Стандартный + фиксация факта аварийного входа и всех действий	Стандартный + фиксация факта усиленной аутентификации и операций с ним	Максимальный (каждое действие, каждое нажатие)	Повышенный (все операции, все изменения данных)
Процедура деактивации	Автоматическая при восстановлении <i>IdP</i> + принудительная смена паролей	Автоматическая после завершения операции или истечения таймера	Принудительное завершение по таймеру, повторный вход только через новую процедуру	По решению руководства после восстановления <i>IdP</i> , принудительный выход всех пользователей
Пост-инцидентные действия	Обязательная смена паролей, анализ причин сбоя	Анализ необходимости усиленной аутентификации, корректировка политик	Обязательное служебное расследование, пересмотр прав доступа	Полный аудит всех действий, коррекция планов <i>BCP</i>

5. Приватность и минимизация данных

В критичных приложениях класса КИИ вопросы приватности приобретают особое значение не только с точки зрения выпол-

нения регуляторных требований (152-ФЗ¹⁴, GDPR¹⁵, отраслевые приказы), но и как фактор снижения рисков компрометации. избыточное хранение и передача данных увеличивают поверхность атаки, поэтому архитектура должна базироваться на принципах минимизации, псевдонимизации и контроля потоков данных.

5.1. Минимизация атрибутов

Для работы системы достаточно ограниченного набора данных [Абселямов, Лагуткина 2024; Ляшов 2024]:

- стабильный идентификатор пользователя (*sub*);
- атрибуты для маппинга ролей (например, *groups*);
- при необходимости – контактные данные (например, *email*).

Прочие персональные данные не должны запрашиваться или сохраняться.

Политика минимизации должна включать:

- перечень запрашиваемых атрибутов с обоснованием;
- перечень разрешенных к хранению атрибутов;
- перечень запрещенных к хранению данных.

Для всех данных устанавливаются минимальные сроки хранения с последующим удалением или обезличиванием. Политика обработки *claims* приведена в табл. 5.

Таблица 5

Политика обработки *claims*

<i>Claim</i>	Зачем нужен	Хранение	Срок хранения	Маскирование в логах
<i>sub</i>	Идентификатор пользователя	Да	Срок жизни аккаунта	Хеш
<i>email</i>	Уведомления	Нет	Не хранится	Полностью
<i>groups</i>	Маппинг ролей	Да	Срок сессии	Частичное
<i>name</i>	Отображение интерфейса	Нет	Не хранится	Полностью

¹⁴ Федеральный закон «О персональных данных» от 27.07.2006 № 152-ФЗ // Информационно-правовой портал ГАРАНТ.РУ. URL: <https://base.garant.ru/12148567/?ysclid=msepcxd39h371904742> (дата обращения 11.05.2026).

¹⁵ General Data Protection Regulation – Общий регламент по защите данных ЕС.

5.2. Разделение идентификаторов

Глобальные идентификаторы *IdP* создают риск межсистемной корреляции данных.

Для его снижения:

- используется механизм *pairwise identifiers* (уникальный идентификатор на приложение);
- вводится локальная псевдонимизация (внутренний *UUID* вместо внешнего идентификатора).

Это исключает связывание данных между системами и снижает последствия компрометации.

5.3. Логи и телеметрия

Логи являются частым источником утечек чувствительных данных [Бетелин и др. 2021].

Запрещено:

- логирование токенов и кодов авторизации;
- запись персональных данных без необходимости;
- использование чувствительных данных в метриках и телеметрии.

Требования:

- маскирование идентификаторов (хеширование);
- агрегирование данных;
- удаление персональных данных при экспорте.

Доступ к неанонимизированным логам должен быть строго ограничен и аудитируем.

6. Метрики

Для оценки эффективности предложенной архитектуры определены метрики безопасности, доступности и приватности. Для каждой метрики заданы единица измерения, способ сбора, целевое и критическое значения. Пороговые значения используются как критерии приемки архитектуры и как основание для запуска процедур реагирования при деградации внешнего *IdP*.

6.1. Метрики безопасности

Метрики безопасности, их целевые и критические значения представлены в табл. 6.

Таблица 6

Метрики безопасности

Метрика	Определение	Единица измерения	Способ сбора	Целевое значение	Критическое значение
Доля отклоненных токенов доступа	Отношение количества запросов, отклоненных на этапе проверки токена	%	Логи шлюза аутентификации, счетчики за интервал (1 ч)	< 0,1% в штатном режиме	> 1% в течение 15 мин
Частота невалидных <i>state/nonce</i>	Количество запросов к <i>endpoint</i> -ам аутентификации, в которых параметр <i>state/nonce</i> отсутствует, не совпадает с ожидаемым или имеет неверный формат	событий/ч	Логи шлюза аутентификации	< 5 событий/ч	> 20 событий/ч
Время отзыва доступа (MTTR)	Интервал от момента обнаружения события, требующего принудительного завершения сессий, до момента, когда все активные сессии пользователя завершены, а токены добавлены в черный список	с	Ручное тестирование / логи	< 60 с (автоматический отзыв)	> 300 с

Окончание табл. 6

Метрика	Определение	Единица измерения	Способ сбора	Целевое значение	Критическое значение
Количество аварийных сессий	Абсолютное количество успешно активированных аварийных сессий за отчетный период	<i>сессий/мес</i>	Журнал аудита	<i>< 2 сессий/мес</i> (нормальная эксплуатация)	<i>> 5 сессий/мес</i> (требуется анализ)
Доля сервисных учеток с устаревшими секретами	Процент сервисных учетных записей, у которых <i>client_secret</i> или сертификат не обновлялся более установленного срока (например, 90 дней)	%	Инвентаризация учетных записей	0% (все секреты актуальны)	> 5%

6.2. Метрики доступности

Метрики доступности и их пороговые значения представлены в табл. 7.

Таблица 7

Метрики доступности

Метрика	Определение	Единица измерения	Способ сбора	Целевое значение	Критическое значение
Штатный режим	Доля успешно завершенных попыток аутентификации (получен валидный токен, создана сессия) от общего количества попыток при работающем <i>IdP</i>	%	Метрики приложения, логи	> 99,9% (три девятки)	< 99%

Окончание табл. 7

Метрика	Определение	Единица измерения	Способ сбора	Целевое значение	Критическое значение
Дегра- дация <i>IdP</i>	Доля успешных попыток входа (через резервные механизмы) от общего количества попыток в период, когда система зафиксировала недоступность <i>IdP</i>	%	Метрики приложения, логи (с меткой «резервный режим»)	> 95% для операций уровня 1	< 80%
<i>RTO</i>	Время, в течение которого функция аутентификации может быть полностью недоступна для пользователей	мин	Тестирование отказоустойчивости	< 5 мин (автоматическое переключение)	> 15 мин
Время переключения на резервный режим	Интервал от момента наступления условия для активации резервного режима до момента, когда резервный режим фактически вступил в действие	с	Логи + метрики	< 30 с (автоматическое) < 300 с (с утверждением)	> 60 с / > 600 с
P95 время ответа (штатное)	95-й процентиль времени выполнения полного цикла аутентификации (от запроса до получения токена)	мс	Метрики приложения	< 500 мс	> 2000 мс
P95 время ответа (деградация)	95-й процентиль времени выполнения аутентификации при работе через резервные механизмы	мс	Метрики приложения	< 2000 мс	> 5000 мс

6.3. Метрики приватности и соответствия

Состав метрик приватности и соответствия представлен в табл. 8.

Таблица 8

Метрики приватности

Метрика	Определение	Единица измерения	Способ сбора	Целевое значение	Критическое значение
Объем хранимых атрибутов	Количество типов атрибутов пользователей, фактически хранящихся в базах данных (с классификацией по чувствительности)	шт.	Инвентаризация БД	Соответствует утвержденному перечню	Превышение перечня на 1 и более
Срок хранения персональных данных	Максимальный фактический срок хранения персональных данных от момента получения до удаления	дней	Анализ БД, аудит	≤ установленного политикой (например, 90 дней)	Превышение > 10% от нормы
Доля записей с маскированием	Процент записей аудита, в которых применено маскирование персональных данных (для систем, где маскирование предусмотрено)	%	Выборка логов	100% для логов разработчиков	< 100%
Утечки в логи	Наличие токенов или персональных данных в логах в открытом виде	бинарно (есть/нет)	Регулярный аудит логов (автоматический поиск паттернов)	Отсутствие утечек	Любое обнаружение
Соответствие <i>pairwise</i> -идентификаторов	Доля приложений, использующих <i>pairwise</i> -идентификаторы вместо глобальных	%	Аудит конфигурации <i>IdP</i>	100%	< 100%

Представленные метрики позволяют количественно оценить соответствие системы требованиям приватности и своевременно выявлять отклонения от установленных политик.

7. Тест-план проверки

Для валидации требований безопасности, отказоустойчивости и приватности разработан воспроизводимый тест-план с формальными критериями *Pass/Fail*. Тестирование проводится при вводе системы в эксплуатацию и периодически (не реже одного раза в квартал).

7.1. Экспериментальные условия

Экспериментальная валидация проводилась на лабораторном стенде, включающем клиентское приложение, шлюз аутентификации, локальный *IAM* и эмулятор внешнего *IdP*. Для каждого сценария *S1-S7* выполнялось не менее 30 прогонов. Нагрузка моделировалась в виде параллельных пользовательских сессий с контролируемым уровнем интенсивности запросов. Метрики доступности (*success rate*), задержек (*p95*) и времени восстановления (*RTO*) рассчитывались на основе агрегированных результатов испытаний. Воспроизводимость обеспечивалась фиксированной конфигурацией стенда и сценариев отказа.

Методика проведения эксперимента. Для каждого сценария сбоя выполнялись:

1. Измерение базовых показателей в штатном режиме (10 минут).
2. Внесение возмущения (отключение эндпоинта, внесение задержек).
3. Фиксация времени переключения на резервный режим и показателей доступности.
4. Восстановление штатного режима и фиксация возврата.

Результаты тестирования отказоустойчивости представлены в табл. 9.

Таблица 9

Результаты тестирования сценариев сбоя

Сценарий	<i>Success rate</i> (до сбоя)	<i>Success rate</i> (во время сбоя)	p95 время ответа (до)	p95 время ответа (во время)	Время переключения на резервный режим	RTO (фактическое)
<i>S1</i> : Полная недоступность <i>IdP</i>	99,95%	97,2%	320 <i>мс</i>	890 <i>мс</i>	12 <i>с</i>	12 <i>с</i>
<i>S2</i> : Дegrаdация (задержки 2000 <i>мс</i>)	99,92%	94,5%	340 <i>мс</i>	2450 <i>мс</i>	25 <i>с</i>	25 <i>с</i>
<i>S3</i> : Сбой эмиссии токенов	99,94%	89,3%	330 <i>мс</i>	520 <i>мс</i>	18 <i>с</i>	18 <i>с</i>
<i>S4</i> : Недоступность <i>JWKS</i> (с кэшем)	99,93%	98,1%	340 <i>мс</i>	410 <i>мс</i>	не требовалось	0 <i>с</i>
<i>S4</i> : Недоступность <i>JWKS</i> (без кэша)	99,91%	0% (<i>fail-closed</i>)	330 <i>мс</i>	50 <i>мс</i> (отказ)	35 <i>с</i>	35 <i>с</i>

Success rate – доля успешных попыток аутентификации или продления пользовательской сессии от общего числа попыток за интервал измерения. P95 время ответа – 95-й процентиль времени выполнения операции аутентификации (от отправки запроса до создания пользовательской сессии).

Важно, что в сценарии *S3* измеряется не первичная аутентификация пользователя, а успешность обновления токена доступа, поэтому показатель *success rate* имеет иную природу и не должен напрямую сравниваться с метрикой доступности входа.

Результат тестирования безопасности приведен в табл. 10.

Таблица 10

Результат тестирования безопасности

ID теста	Сценарий	Метод проверки	Ожидаемый результат	Результат на стенде	Статус
SEC-01	Replay-атака authorization code	Повторная отправка ранее перехваченного code на token endpoint в течение 5 мин	Отказ с ошибкой invalid_grant	Отказ получен, код invalid_grant	PASS
SEC-02	Подмена параметра state (CSRF)	Изменение значения state в ответе от IdP перед отправкой клиенту	Отказ клиента от обработки ответа, сессия не создается	Сессия не создана, ошибка в консоли клиента	PASS
SEC-03	Использование JWT с невалидной подписью	Модификация payload JWT без переподписи	Отказ с кодом 401 (Unauthorized) на шлюзе аутентификации	401 Unauthorized	PASS
SEC-04	Использование JWT с неверной aud	Подмена aud на client_id другого приложения	Отказ с кодом 401, запись в аудит "invalid audience"	401, аудит зафиксировал	PASS
SEC-05	Использование JWT с неверным iss	Подмена iss на фиктивный URL	Отказ с кодом 401, запись в аудит "invalid issuer"	401, аудит зафиксировал	PASS
SEC-06	Использование просроченного JWT	Использование токена через 1 ч после истечения срока	Отказ с кодом 401, запись "token expired"	401, аудит зафиксировал	PASS

Полученные результаты показывают, что фактическое *RTO* во всех протестированных сценариях не превысило 35 с, что соответствует целевому требованию менее 5 мин. Наиболее значимым фактором сохранения доступности оказалось кэширование *JWKS*: при наличии кэша проверка токенов сохраняется, тогда как отсутствие доверенного ключа приводит к корректному отказу по стратегии *fail-closed*.

8. Научная новизна

Научная новизна исследования состоит в формализации внешнего *IdP* как потенциально ненадежного элемента критической инфраструктуры и в разработке проверяемой архитектуры аутентификации, объединяющей модель угроз, резервные режимы доступа, метрики доступности и тест-план отказоустойчивости.

9. Практическая значимость

Практическая значимость работы заключается в возможности использования предложенной архитектуры, матрицы угроз и набора метрик при проектировании, аудите и приемке *SSO*-решений для критичных информационных систем.

Заключение

Проведенное исследование подтвердило, что использование внешних поставщиков идентификации в критичных приложениях должно рассматриваться не только как задача удобной аутентификации, но и как элемент обеспечения непрерывности функционирования системы. Отказ или компрометация внешнего *IdP* влияет на доступность пользовательских сессий, корректность проверки токенов и возможность выполнения критически важных операций. Поэтому архитектура *SSO* в таких условиях должна проектироваться с учетом рисков безопасности, отказоустойчивости и приватности данных.

В работе разработана и экспериментально валидирована риск-ориентированная архитектура аутентификации на основе *OAuth 2.0* и *OpenID Connect*. Ее ключевыми элементами являются централизованная проверка токенов на шлюзе аутентификации, принцип недоверия к токену до завершения полной валидации, кэширова-

ние *JWKS* с контролируемым *TTL*, разграничение интерактивных пользователей и сервисных учетных записей, а также заранее определенные резервные режимы доступа: *fallback*, *step-up*, *break-glass* и *BSP*. Предложенная модель угроз и верификационная матрица позволяют связать конкретные угрозы с контролями и способами проверки их работоспособности.

Экспериментальная проверка на лабораторном стенде показала, что предложенные механизмы позволяют сохранить доступность аутентификации при деградации или недоступности внешнего *IdP*. В рассмотренных сценариях отказа фактическое время восстановления не превысило 35 с, а кэширование *JWKS* позволило сохранить проверку токенов при недоступности внешнего *endpoint*. Результаты также подтвердили необходимость стратегии *fail-closed* при отсутствии доверенного ключа, поскольку сохранение доступности не должно приводить к неконтролируемому ослаблению защиты.

Практическая значимость предложенного подхода состоит в том, что он может использоваться как основа для проектирования, аудита и приемки SSO-решений в критичных информационных системах. Формализованные сценарии отказа, критерии активации резервных режимов, метрики доступности и тест-план делают архитектуру проверяемой и воспроизводимой.

Ограничением исследования является лабораторный характер стенда: в нем не учитывались все особенности промышленной сетевой инфраструктуры, высоконагруженных распределенных систем и организационных процедур реагирования на инциденты. Перспективы дальнейших исследований связаны с автоматизацией переключения между режимами аутентификации, расширением сценариев проверки компрометации *IdP* и применением методов поведенческого анализа для выявления аномалий при работе с токенами и пользовательскими сессиями.

Литература

- Абселямов, Лагуткина 2024 – Абселямов А.-Х.А., Лагуткина Т.В. Исследование методов аутентификации на веб-сервисах. Текущие тенденции и перспективы развития // Научный результат. Информационные технологии. 2024. Т. 9, № 2. С. 12–20. DOI: 10.18413/2518-1092-2024-9-2-0-2. EDN EUYTKM.
- Барабанов, Макрушин 2020 – Барабанов А., Макрушин Д. Аутентификация и авторизация в микросервисных приложениях: обзор архитектурных подходов // Вопросы кибербезопасности. 2020. № 4 (38). С. 32–43. DOI: 10.21681/2311-3456-2020-04-32-43. EDN ZPWRTG.

- Бетелин и др. 2021 – *Бетелин А.Б., Егорычев И.Б., Прилипко А.А., Прилипко Г.А., Романюк С.Г., Самборский Д.В.* О некоторых особенностях JWT-аутентификации в веб-приложениях // Труды НИИСИ РАН. 2021. Т. 11, № 1. С. 4–10. DOI: 10.25682/NIISI.2021.1.0001. EDN VRXEVL.
- Валеев, Кондратьева 2023а – *Валеев С.С., Кондратьева Н.В.* Особенности проектирования систем безопасности на базе архитектуры нулевого доверия // Инженерный вестник Дона. 2023. № 8 (104). С. 223–230. EDN TBWJXY.
- Валеев, Кондратьева 2023б – *Валеев С.С., Кондратьева Н.В.* Паттерны проектирования архитектуры нулевого доверия // Инженерный вестник Дона. 2023. № 9. С. 76–83. EDN FYJVBV.
- Демидова, Жуков 2020 – *Демидова А.Ю., Жуков А.В.* Технология Single Sign-On: инструменты централизованной аутентификации для функциональной системы сервисов // Инженерный вестник Дона. 2020. № 3. С. 10. EDN CKTJRR.
- Докучаев и др. 2025 – *Докучаев В.А., Мытенков С.С., Рахмани Д., Сафонов И.А.* Анализ уязвимостей и рисков традиционных парольных систем в контексте корпоративных распределенных систем и критически важных инфраструктур // Экономика и качество систем связи. 2025. № 2 (36). С. 135–147.
- Зими́на, Лапо́нина 2023 – *Зими́на К.И., Лапо́нина О.Р.* Механизмы межсервисной аутентификации в приложениях с микросервисной архитектурой // International Journal of Open Information Technologies. 2023. Т. 11, № 5. С. 146–154. EDN TQJTJ.
- Иванов, Капгер, Шабуров 2023 – *Иванов П.А., Капгер И.В., Шабуров А.С.* Модель реализации управления доступом к информационным активам в концепции нулевого доверия // Вестник ПНИПУ. Электротехника, информационные технологии, системы управления. 2023. № 45. С. 147–163. EDN ZHSITI.
- Кирильчук, Аблитаров 2024 – *Кирильчук С.П., Аблитаров Э.Р.* Архитектура нулевого доверия как инновационный инструмент в системе обеспечения защиты корпоративной информации // Ученые записки КФУ им. В.И. Вернадского. Экономика и управление. 2024. Т. 10 (76), № 3. С. 45–57. EDN FPNOXC.
- Ляшов 2024 – *Ляшов Е.И.* Виды аутентификации в современных веб-приложениях // Universum: технические науки. 2024. № 12 (129). С. 56–61. DOI: 10.32743/UniTech.2024.129.12.18989. EDN RQIQFT.
- Рахмани 2022 – *Рахмани Д.* Исследование методов управления рисками в инфокоммуникационной системе энергопроизводящей компании Исламской Республики Иран // Т-Comm: Телекоммуникации и транспорт. 2022. Т. 16, № 8. С. 30–37. DOI: 10.36724/2072-8735-2022-16-8-30-37. EDN EUYHCJ.
- Саломатин, Исхаков 2020 – *Саломатин А.А., Исхаков А.Ю.* Применение интегрированного показателя отпечатков браузера в задаче адаптивной аутентификации субъектов доступа // Информационные и математические технологии в науке и управлении. 2020. № 4 (20). С. 84–92. EDN MJQSHQ.
- Старосельский, Жиренкин 2023 – *Старосельский А.К., Жиренкин А.В.* Авторизация и обмен данными в веб-приложениях с использованием протокола OAuth 2.0 // Вестник науки. 2023. Т. 4, № 6 (63). С. 909–913.

- Шарилов, Олин, Габдрахманов 2025 – Шарипов Р.Р., Олин Р.А., Габдрахманов Э.М. Разработка системы двухфакторной аутентификации в web-приложениях на основе дактилоскопии // Computational nanotechnology. 2025. Т. 12, № 4. С. 131–142. EDN GDJYJS.

References

- Abseiyamov, A.-Kh.A. and Lagutkina, T.V. (2024), “Investigation of authentication methods on web services. Current trends and development prospects”, *Research result. Information technologies*, vol. 9, no. 2, pp. 12–20, DOI: 10.18413/2518-1092-2024-9-2-0-2.
- Barabanov, A. and Makrushin, D. (2020), “Authentication and authorization in microservice applications: an architectural overview”, *Voprosy kiberbezopasnosti*, no. 4 (38), pp. 32–43, DOI: 10.21681/2311-3456-2020-04-32-43.
- Betelin, A.B., Egorychev, I.B., Prilipko, A.A., Prilipko, G.A., Romanyuk, S.G. and Samborskiy, D.V. (2021), “Some features of JWT authentication in web applications”, *Trudy NIISI RAN*, vol. 11, no. 1, pp. 4–10, DOI: 10.25682/NIISI.2021.1.0001.
- Demidova, A.Yu. and Zhukov, A.V. (2020), “Single Sign On technology: centralized authentication tools for a functional service system”, *Inzhenernyy vestnik Dona*, no. 3, p. 10.
- Dokuchaev, V.A., Mytenkov, S.S., Rahmani, D.D. and Safonov, I.A. (2025), “Authentication algorithm based on verified encryption using one-time keys for smart grid systems in power grids”, *Ekonomika i kachestvo sistem svyazi*, no. 2 (36), pp. 135–147.
- Ivanov, P.A., Kapger, I.V. and Shaburov, A.S. (2023), “Model of access management to information assets in zero trust concept”, *Vestnik PNIPU. Elektrotehnika, informatsionnye tekhnologii, sistemy upravleniya*, no. 45, pp. 147–163.
- Kirilchuk, S.P. and Ablitarov, E.R. (2024), “Zero trust architecture as an innovative tool for protecting corporate information”, *Uchenye zapiski KFU im. V.I. Vernadskogo. Ekonomika i upravlenie*, vol. 10 (76), no. 3, pp. 45–57.
- Lyashov, E.I. (2024), “Types of authentication in modern web applications”, *Univer-sum: tekhnicheskie nauki*, vol. 12 (129), pp. 56–61, DOI: 10.32743/UniTech.2024.129.12.18989.
- Rahmani, D. (2022), “Research of risk management methods in the infocommunication system of the energy producing company of the Islamic Republic of Iran”, *T-Comm*, vol. 16, no. 8, pp. 30–37, DOI: 10.36724/2072-8735-2022-16-8-30-37.
- Salomatin, A.A. and Iskhakov, A.Yu. (2020), “Application of the integrated indicator of browser fingerprinting in the problem of adaptive authentication of access subjects”, *Informatsionnye i matematicheskie tekhnologii v nauke i upravlenii*, no. 4 (20), pp. 84–92.

- Sharipov, R.R., Olin, R.A. and Gabdrakhmanov, E.M. (2025), "Development of a two-factor authentication system in WEB applications based on fingerprinting", *Computational Nanotechnology*, vol. 12, no. 4, pp. 131–142.
- Staroselskiy, A.K. and Zhirenkin, A.V. (2023), "Authorization and data exchange in web applications using the OAuth 2.0 protocol", *Vestnik nauki*, vol. 4, no. 6 (63), pp. 909–913.
- Valeev, S.S. and Kondratyeva, N.V. (2023), "Features of designing security systems based on zero-trust architecture", *Inzhenernyy vestnik Dona*, no. 8 (104), pp. 223–230.
- Valeev, S.S. and Kondratyeva, N.V. (2023), "Zero-Trust Architecture Design Patterns", *Inzhenernyy vestnik Dona*, no. 9, pp. 76–83.
- Zimina, K.I. and Laponina, O.R. (2023), "Cross-Service Authentication Mechanisms in Applications with Microservice Architecture", *International Journal of Open Information Technologies*, vol. 11, no. 5, pp. 146–154.

Информация об авторах

Джахед Рахмани, Московский технический университет связи и информатики (МТУСИ), Москва, Россия; 111024, Россия, Москва, ул. Авиамоторная, д. 8А; jahed@mail.ru

Анна В. Сугрובה, студент, Московский технический университет связи и информатики (МТУСИ), Москва, Россия; 111024, Россия, Москва, ул. Авиамоторная, д. 8А; a.v.sugrobova@edu.mtuci.ru

Information about the authors

Dzhakhed Rakhmani, Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russia; 8A, Aviamotornaya Str., Moscow, 111024, Russia; jahed@mail.ru

Anna V. Sugrobova, student, Moscow Technical University of Communications and Informatics (MTUCI), Moscow, Russia; 8A, Aviamotornaya Str., Moscow, 111024, Russia; a.v.sugrobova@edu.mtuci.ru

Информационная безопасность как элемент устойчивости цифровых экосистем

Лилия А. Ежова

*Финансовый университет при Правительстве
Российской Федерации, Москва, Россия, laezhova@fa.ru*

Аннотация. В статье исследуется роль информационной безопасности в обеспечении эффективной системы устойчивости цифровых экосистем, функционирующих в киберфизических и социотехнических средах. Актуальность работы обусловлена ростом взаимосвязанности цифровых платформ, промышленного Интернета вещей, интеллектуальных сервисов и распределенных алгоритмических систем, что приводит к усилению системных рисков и повышению чувствительности экосистем к нарушениям целостности и конфиденциальности данных. В условиях высокой интеграции технологических и социальных компонентов информационная безопасность выходит за рамки традиционного понимания как совокупности защитных мер и рассматривается как структурный фактор поддержания устойчивости экосистемного взаимодействия.

В работе обосновано расширенное трактование функций информационной безопасности, включающее стабилизационную, координационную и институциональную составляющие. Показано, что нарушения в сфере защиты информации способны инициировать каскадные эффекты, затрагивающие не только техническую инфраструктуру, но и доверие участников, деловую активность и целостность управленческих процессов. Особое внимание уделено взаимосвязи информационной безопасности и социального капитала цифровых экосистем, а также влиянию алгоритмических механизмов на перераспределение рисков и уязвимостей.

Предложена концептуальная модель интеграции информационной безопасности в систему устойчивости цифровой экосистемы, основанная на взаимодействии технического, социального и управленческого контуров. Сделан вывод о необходимости перехода от реактивных моделей защиты к проактивным стратегиям, учитывающим социотехническую природу современных цифровых сред. Полученные результаты могут быть использованы при разработке комплексных подходов к обеспечению защиты информации в киберфизических и социотехнических системах.

Ключевые слова: информационная безопасность, цифровая экосистема, киберфизические системы, социотехнические системы, алгоритмическое управление

Для цитирования: Ежова Л.А. Информационная безопасность как элемент устойчивости цифровых экосистем // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 141–154. DOI: 10.28995/2686-679X-2026-3-141-154

Information security as an element of sustainability of digital ecosystems

Liliya A. Ezhova

*Financial University under the Government of the Russian Federation
Moscow, Russia, laezhova@fa.ru*

Abstract. This article examines the role of information security in ensuring the resilience of digital ecosystems operating in cyber-physical and socio-technical environments. The relevance of this work stems from the growing interconnectedness of digital platforms, the Industrial Internet of Things, intelligent services, and distributed algorithmic systems, which leads to increased systemic risks and increased sensitivity of ecosystems to breaches of data integrity and confidentiality. In the context of the high integration of technological and social components, information security goes beyond the traditional understanding of a set of protective measures and is considered a structural factor in maintaining the resilience of ecosystem interactions.

The paper substantiates an expanded interpretation of information security functions, including stabilization, coordination, and institutional components. It is shown that breaches in information security can trigger cascading effects affecting not only the technical infrastructure but also the trust of participants, business activity, and the integrity of management processes. Particular attention is paid to the relationship between information security and the social capital of digital ecosystems, as well as the influence of algorithmic mechanisms on the redistribution of risks and vulnerabilities.

A conceptual model for integrating information security into a digital ecosystem resilience system is proposed, based on the interaction of technical, social, and management components. A conclusion is drawn regarding the need to transition from reactive security models to proactive strategies that consider the sociotechnical nature of modern digital environments. The results obtained can be used in developing comprehensive approaches to ensuring information security in cyber-physical and sociotechnical systems.

Keywords: information security, digital ecosystem, cyber-physical systems, sociotechnical systems, algorithmic management

For citation: Ezhova, L.A. (2026), "Information security as an element of sustainability of digital ecosystems", *RSUH/RGGU Bulletin. "Information Science. Information Security. Mathematics" Series*, no. 3, pp. 141–154, DOI: 10.28995/2686-679X-2026-3-141-154

Введение

Цифровые экосистемы в последние годы превратились в доминирующую форму организации экономических, технологических и социальных взаимодействий. Их развитие связано с распространением платформенных моделей бизнеса, промышленного Интернета вещей, интеллектуальных сервисов и распределенных алгоритмических систем управления. В отличие от традиционных информационных систем, цифровые экосистемы характеризуются высокой степенью взаимосвязанности участников, интеграцией киберфизических компонентов и постоянным обменом данными между разнородными субъектами. Такая архитектура обеспечивает гибкость, масштабируемость и ускорение процессов кооперации, однако одновременно усиливает чувствительность к информационным угрозам.

В условиях цифровой трансформации информационная безопасность перестает быть исключительно технической функцией, направленной на защиту отдельных информационных ресурсов. Она приобретает системное значение, влияя на обеспечение устойчивости всей экосистемной структуры. Нарушения конфиденциальности, целостности или доступности данных способны инициировать цепную реакцию, затрагивающую технологическую инфраструктуру, алгоритмические процессы и поведение участников [Николаева 2023]. Таким образом, вопросы защиты информации выходят за пределы локальных задач администрирования и становятся фактором стратегического управления устойчивостью цифровых экосистем.

В теории управления устойчивость системы традиционно рассматривается как ее способность сохранять основные параметры функционирования при воздействии внешних и внутренних факторов, а также возвращаться в состояние равновесия после возмущений. В кибернетическом подходе устойчивость связывается с наличием механизмов обратной связи, адаптации и регулирования, обеспечивающих сохранение целостности системы

и достижение поставленных целей [Ху 2019]. Применительно к цифровым экосистемам целесообразно говорить не только об устойчивости как свойстве системы, но и о системе обеспечения устойчивости, включающей технические, организационные и информационные механизмы поддержания стабильного функционирования.

Особенность современных киберфизических и социотехнических систем заключается в их гибридной природе. Они объединяют физические объекты, программные компоненты, алгоритмы обработки данных и участников, чья деятельность опосредована цифровыми платформами. Устойчивость таких систем определяется не только надежностью технической инфраструктуры, но и уровнем доверия между участниками, прозрачностью алгоритмических решений и способностью быстро адаптироваться к изменяющимся условиям внешней среды [Цветкова, Родионов 2025]. Информационная безопасность в данном контексте выступает связующим элементом между техническими и социальными контурами экосистемы.

Современные исследования в области киберфизических систем и защиты информации преимущественно сосредоточены на анализе уязвимостей, разработке криптографических методов, систем обнаружения вторжений и протоколов безопасного взаимодействия. Эти направления безусловно остаются ключевыми, однако они не исчерпывают проблематику устойчивости цифровых экосистем. При высокой степени интеграции платформенных сервисов, распределенных вычислений и сетевых взаимодействий инциденты в сфере информационной безопасности способны подорвать доверие пользователей, снизить интенсивность кооперации и изменить структуру экосистемных связей. Следовательно, информационная безопасность должна рассматриваться как компонент, формирующий институциональную и социальную устойчивость цифровых сред.

Устойчивость цифровых экосистем предполагает их способность противостоять внешним и внутренним воздействиям, адаптироваться к новым условиям и восстанавливаться после сбоев. В отличие от традиционной отказоустойчивости, ориентированной преимущественно на технические параметры, экосистемная устойчивость охватывает также поведенческие и управленческие аспекты. Потеря доверия к механизмам защиты информации может привести к снижению активности участников, отказу от использования цифровых сервисов и фрагментации взаимодействий. В результате экосистема теряет связность, что негативно отражается на ее конкурентоспособности и жизнеспособности.

Особую значимость проблема приобретает в условиях роста алгоритмического управления и широкого использования систем искусственного интеллекта. Алгоритмы, принимающие решения на основе анализа больших массивов данных, становятся активными посредниками в распределении ресурсов, ранжировании информации и координации действий. При этом уязвимости в алгоритмических моделях или манипулирование входными данными способны не только нарушить безопасность, но и исказить управленческие процессы [Титов и др. 2025]. Таким образом, защита информации тесно связана с обеспечением корректности и прозрачности алгоритмических решений.

Следует учитывать и социотехническое измерение проблемы. В цифровых экосистемах участники взаимодействуют на основе платформенных правил и репутационных механизмов, что делает доверие ключевым элементом функционирования [Георгиевский 2022]. Информационная безопасность выступает условием поддержания этого доверия, поскольку гарантирует предсказуемость и защищенность цифровых транзакций. В противном случае возникает риск разрушения социальных связей и снижения коллективной готовности к обмену знаниями и ресурсами.

Несмотря на очевидную взаимосвязь информационной безопасности и устойчивости цифровых экосистем, в научной литературе данная связь зачастую рассматривается фрагментарно. Отсутствует целостная концептуальная рамка, позволяющая интегрировать технические, управленческие и социальные аспекты защиты информации в единую модель устойчивости. Это обстоятельство определяет необходимость переосмысления роли информационной безопасности в контексте киберфизических и социотехнических систем.

Цель статьи заключается в обосновании информационной безопасности как системообразующего элемента обеспечения устойчивости цифровых экосистем. В работе предлагается рассматривать защиту информации не только как совокупность технических мер, но и как механизм поддержания доверия, координации и адаптивности в распределенных цифровых средах. Такой подход позволяет расширить традиционную трактовку информационной безопасности и интегрировать ее в более широкий контекст управления цифровыми экосистемами.

Актуальность исследования определяется тем, что цифровые экосистемы становятся основой функционирования критически значимых инфраструктур, финансовых платформ, государственных сервисов и промышленных систем. Их устойчивость напрямую связана с качеством механизмов защиты информации. Формирование научно обоснованной модели интеграции инфор-

мационной безопасности в контур устойчивости является необходимым условием разработки эффективных стратегий защиты в киберфизических и социотехнических системах¹.

Основная часть

Цифровая экосистема представляет собой многокомпонентную социотехническую систему, включающую инфраструктурные элементы, алгоритмические механизмы обработки данных, нормативные правила взаимодействия и совокупность участников. В отличие от изолированных информационных систем, экосистемы характеризуются высокой степенью сетевой связанности, распределенностью архитектуры и постоянной циркуляцией данных между разнородными субъектами².

Ключевой особенностью цифровых экосистем является их структурная нелинейность – изменение состояния одного элемента способно повлиять на функционирование всей системы [Карев 2025]. При этом угрозы информационной безопасности затрагивают не только технологический уровень, но и социальные и управленческие процессы. Таким образом, объект защиты выходит за пределы отдельных информационных ресурсов и охватывает устойчивость всей экосистемной структуры.

С позиций системного анализа цифровая экосистема может быть представлена как совокупность взаимосвязанных контуров (табл. 1).

Структура, представленная в табл. 1 демонстрирует, что угрозы присутствуют не только на техническом уровне, но и в социальном и управленческом контурах.

Классическая модель информационной безопасности базируется на триаде конфиденциальность-целостность-доступность (CIA). Данная модель сохраняет актуальность, однако в контексте цифровых экосистем она оказывается недостаточной для описания системной устойчивости.

¹ Национальная программа «Цифровая экономика Российской Федерации». URL: <http://static.government.ru/media/files/9gFM4FHj4PsB7915v7yLVuPgu4bvR7M0.pdf> (дата обращения: 10.04.2026).

² ISO/IEC 27001:2022. Информационная безопасность, кибербезопасность и защита конфиденциальности. Системы менеджмента информационной безопасности. Требования / Пер. А. Горбунова. Неофициальный перевод. URL: <https://pqm-online.com/assets/files/pubs/translations/std/iso-mek-27001-2022.pdf> (дата обращения: 10.04.2026).

Таблица 1

Структурные контуры цифровой экосистемы

Контур	Основные элементы	Угрозы (последствия атак)
Технический	Серверы, сети, IoT-устройства, алгоритмы	1) Кибератаки и несанкционированный доступ; 2) Сбои программного обеспечения; 3) Отказ аппаратного оборудования; 4) Уязвимости сетевой инфраструктуры; 5) Потеря или искажение данных
Социальный	Пользователи, сообщества, партнеры	1) Человеческий фактор (ошибки, неосторожность); 2) Утрата доверия к платформе; 3) Недобросовестное поведение участников; 4) Информационные манипуляции; 5) Низкий уровень цифровой грамотности
Управленческий	Правила, протоколы, регламенты, алгоритмические решения	1) Непрозрачность управленческих процессов; 2) Концентрация принятия решений; 3) Ошибки алгоритмических моделей; 4) Недостаточная адаптивность регламентов; 5) Конфликты интересов

Источник: составлено автором.

В условиях высокой взаимосвязанности угрозы приобретают каскадный характер. Нарушение доступности сервиса способно повлиять на поведение участников, вызвать потерю доверия и снизить активность в экосистеме [Бистерфельд и др. 2025]. Следовательно, информационная безопасность выполняет не только защитную, но и стабилизирующую функцию.

Расширенный функциональный подход представлен в табл. 2.

Исходя из табл. 2, информационная безопасность выступает как элемент управления рисками, влияющий на способность экосистемы поддерживать структурную целостность.

Таблица 2

Расширенные функции информационной безопасности

Функция	Содержание	Роль в устойчивости
Защитная	Предотвращение утечек и атак	Снижение уязвимости
Стабилизирующая	Поддержание доверия	Сохранение взаимодействия
Координационная	Обеспечение предсказуемости процессов	Управляемость системы
Адаптивная	Быстрое реагирование на инциденты	Восстановление после сбоев

Источник: составлено автором.

Устойчивость цифровых экосистем проявляется в трех аспектах: техническом, социальном и институциональном. Информационная безопасность оказывает влияние на каждый из них.

1. Техническая устойчивость – способность инфраструктуры функционировать при попытках внешнего воздействия.

2. Социальная устойчивость – сохранение доверия и кооперации участников.

3. Институциональная устойчивость – легитимность правил и управленческих решений.

Нарушение требований ИБ может инициировать цепную реакцию, представленную на рис. 1.

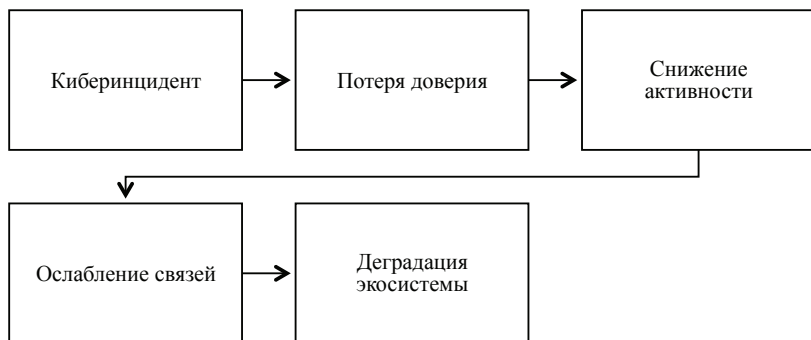


Рис. 1. Цепная реакция нарушения информационной безопасности

Источник: составлено автором

Данная логика на рис. 1 подчеркивает системообразующую роль защиты информации.

На основе проведенного анализа предлагается концептуальная модель, в которой информационная безопасность рассматривается как связующий элемент между тремя контурами экосистемы: техническим, социальным и управленческим. Модель представлена на рис. 2.



Рис. 2. Концептуальная модель информационной безопасности

Источник: составлено автором

В концептуальной модели выделены следующие компоненты (рис. 2):

- Технический контур (инфраструктура) обеспечивает физическую и программную защиту.
- Социальный контур (доверие, поведение) отражает поведенческую реакцию участников.
- Управленческий контур (правила, алгоритмы, регламенты) задает правила и механизмы реагирования.

Информационная безопасность выступает медиатором, обеспечивающим согласованность данных контуров.

Эффективность системы определяется степенью синхронизации этих уровней.

В цифровых экосистемах управление осуществляется с использованием алгоритмов анализа данных [Гришина 2022]. Поэтому устойчивость может оцениваться через набор индикаторов:

- уровень инцидентов безопасности;
- скорость реагирования;
- индекс доверия пользователей;
- показатели вовлеченности.

В упрощенной форме интегральный показатель устойчивости (U) цифровой экосистемы можно представить как функционал:

$$U = f(T, S, G), \quad (1)$$

где T – технический уровень защищенности, который отражает состояние инфраструктуры, включая устойчивость серверов, сетей, программного обеспечения и механизмов защиты данных;

S – социальный индекс доверия, который характеризует степень доверия пользователей и участников системы, их готовность взаимодействовать с платформой и соблюдать установленные правила;

G – управленческая эффективность реагирования, которая показывает способность системы управления своевременно выявлять риски, принимать решения и адаптировать регламенты в условиях изменений.

Подобная формализация демонстрирует необходимость комплексного подхода.

Интеграция информационной безопасности в систему обеспечения устойчивости сопряжена с рядом противоречий:

- централизация управления безопасностью усиливает зависимость экосистемы от платформенного ядра;
- алгоритмические механизмы фильтрации и ранжирования информации могут создавать дисбалансы в распределении ресурсов внутри цифровой экосистемы.

Ключевые риски для устойчивости предложенной модели представлены в табл. 3.

Таблица 3

Ключевые риски для устойчивости

Фактор	Потенциальное воздействие
Концентрация данных	Системный сбой при компрометации
Недостаточная прозрачность алгоритмов	Снижение доверия
Низкий уровень цифровой грамотности	Рост внутренних угроз

Источник: составлено автором.

В экосистемной логике информационная безопасность выполняет функцию мета-управления. Она обеспечивает:

- согласованность технических и социальных процессов;
- поддержание доверия;
- минимизацию системных рисков;
- способность к восстановлению.

Контур обеспечения устойчивости представлен на рис. 3.

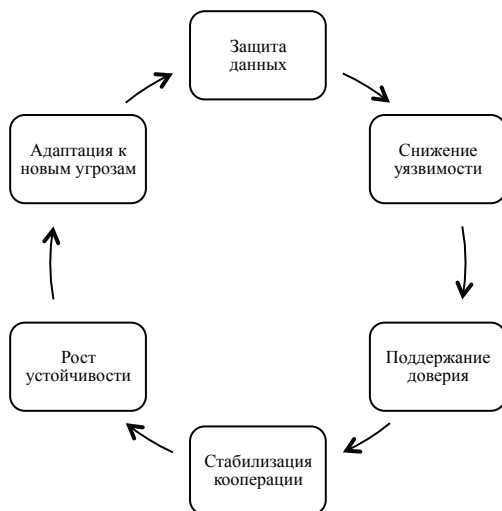


Рис. 3. Контур обеспечения устойчивости

Источник: составлено автором

Таким образом, в цифровой экосистеме информационная безопасность становится не внешним инструментом контроля, а встроенным механизмом поддержания системного равновесия.

Заключение

1. Проведенное исследование позволило обосновать положение о том, что информационная безопасность в цифровых экосистемах не может рассматриваться исключительно как совокупность технических мер защиты данных и инфраструктуры. В условиях интеграции киберфизических и социотехнических компонентов она приобретает статус системообразующего фактора, напрямую влияющего на устойчивость экосистемы в целом.

2. Устойчивость цифровых экосистем имеет многомерный характер и включает технический, социальный и управленческий аспекты. Нарушения в сфере защиты информации оказывают каскадное воздействие на все указанные уровни, трансформируясь из локальных инцидентов в структурные риски.

3. Классическая модель информационной безопасности, основанная на обеспечении конфиденциальности, целостности и доступности, применительно к экосистемной среде требует расширения. Информационная безопасность выполняет не только защитную, но и стабилизирующую, координационную и адаптивную функции.

4. В рамках статьи предложена концептуальная модель интеграции информационной безопасности в систему устойчивости цифровой экосистемы, основанная на взаимодействии трех взаимосвязанных контуров: технического, социального и управленческого. Технический контур обеспечивает инфраструктурную защищенность, социальный отражает поведенческие и доверительные аспекты взаимодействия акторов, управленческий – задает правила, регламенты и алгоритмические механизмы координации.

Литература

Бистерфельд и др. 2025 – Бистерфельд О.А., Фошина М.А., Чжан Ц. Аппаратно-программный комплекс экосистемы Huawei: реинжиниринг в условиях экономических санкций // Вестник РГГУ. Серия: Информатика. Информационная безопасность. Математика. 2025. № 4. С. 25–38. DOI: 10.28995/2686-679X-2025-4-25-38. EDN HWIPHF.

- Георгиевский 2022 – *Георгиевский А.Б.* Экосистемы российского ритейла: основные участники и индикаторы формирования // ЭКО. 2022. № 4 (574). С. 138–155. DOI: 10.30680/ECO0131-7652-2022-4-138-155. EDN FMPRGU.
- Гришина 2022 – *Гришина Н.В.* Анализ динамики утечки персональных данных в условиях реализации программы «Цифровая экономика Российской Федерации» // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2022. № 4. С. 34–43.
- Карев 2025 – *Карев А.В.* Особенности цифровой торговли и развитие цифровых экосистем // Друкеровский вестник. 2025. № 2 (64). С. 24–34. DOI: 10.17213/2312-6469-2025-2-24-34. EDN IGHGFY.
- Николаева 2023 – *Николаева М.О.* Информационная безопасность: современная картина проблемы информационной безопасности и защиты информации // Мониторинг. Образование. Безопасность. 2023. № 1 (1). С. 51–57. EDN IOIQDI.
- Титов и др. 2025 – *Титов А.П., Гришина Н.В., Титова Д.Н.* Разработка интеллектуального анализатора уязвимостей для динамического сканирования веб-приложений // Вестник РГГУ. Серия: Информатика. Информационная безопасность. Математика. 2025. № 4. С. 77–98. DOI: 10.28995/2686-679X-2025-4-77-98. EDN PDKTFO.
- Цветкова, Родионов 2025 – *Цветкова В.А., Родионов И.И.* О современном состоянии российской информационной инфраструктуры // Вестник РГГУ. Серия: Информатика. Информационная безопасность. Математика. 2025. № 4. С. 51–76. DOI: 10.28995/2686-679X-2025-4-51-76. EDN OGHDKE.
- Xu 2019 – *Xu S.* Cybersecurity Dynamics: A Foundation for the Science of Cybersecurity // Advances in Information Security. 2019. Vol. 74. P. 1–31. DOI: 10.1007/978-3-030-10597-6_1. EDN YAXBYD.

References

- Bisterfeld, O.A., Foshina, M.A. and Zhang, Z. (2025), “Hardware and software complex of the Huawei ecosystem: reengineering under economic sanctions”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, vol. 4, pp. 25–38, DOI: 10.28995/2686-679X-2025-4-25-38.
- Georgievskiy, A.B. (2022), “Russian retail ecosystems: main participants and formation indicators”, *ECO*, no. 4 (574), pp. 138–155, DOI: 10.30680/ECO0131-7652-2022-4-138-155.
- Grishina, N.V. (2022), “Analysis of the dynamics of personal data leaks in the context of the implementation of the “Digital Economy of the Russian Federation” program”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, vol. 4, pp. 34–43.
- Karev, A.V. (2025), “Features of digital commerce and the development of digital ecosystems”, *Drucker Bulletin*, no. 2 (64), pp. 24–34, DOI: 10.17213/2312-6469-2025-2-24-34.

- Nikolaeva, M.O. (2023), "Information security: the current state of information security and information protection", *Monitoring. Education. Security*, no. 1 (1), pp. 51–57.
- Titov, A.P., Grishina, N.V. and Titova, D.N. (2025), "Development of an intelligent vulnerability analyzer for dynamic scanning of web applications", *RSUH/RGGU Bulletin. "Information Science. Information Security. Mathematics" Series*, vol. 4, pp. 77–98, DOI: 10.28995/2686-679X-2025-4-77-98.
- Tsvetkova, V.A. and Rodionov, I.I. (2025), "On the current state of the Russian information infrastructure", *RSUH/RGGU Bulletin. "Information Science. Information Security. Mathematics" Series*, vol. 4, pp. 51–76, DOI: 10.28995/2686-679X-2025-4-51-76.
- Ху, S. (2019), "Cybersecurity dynamics: a foundation for the science of cybersecurity", *Advances in Information Security*, vol. 74, pp. 1–31, DOI: 10.1007/978-3-030-10597-6_1.

Информация об авторе

Лилия А. Ежова, кандидат экономических наук, Финансовый университет при Правительстве Российской Федерации, Москва, Россия; 125167, Россия, Москва, Ленинградский пр., д. 49/2; laezhova@fa.ru

Information about the author

Liliya A. Ezhova, Cand. of Sci. (Economics), Financial University under the Government of the Russian Federation, Moscow, Russia; 49/2, Leningradsky Av., Moscow, 125167, Russia; laezhova@fa.ru

Структурно-параметрический синтез биометрической системы аутентификации пользователя

Александр В. Винокуров

*Российский государственный гуманитарный университет,
Москва, Россия, VAV73@rambler.ru*

Аннотация. Современные требования, предъявляемые к информационным системам, в которых обрабатываются сведения ограниченного распространения, определяют необходимость разработки новых методов и подходов к аутентификации пользователей, сочетающих высокую надежность с удобством использования. В статье исследуется актуальный подход к созданию биометрической системы аутентификации пользователя, основанной на нейросетевом преобразовании его динамических биометрических параметров (анализа ввода пароля на клавиатуре) в уникальный идентификатор. Построена функциональная модель системы высоконадежной аутентификации пользователя. Проводится структурно-параметрический синтез биометрической системы аутентификации пользователей путем комплексирования сверточной нейросети, рекуррентных слоев и энкодера, что в комплексе обеспечивает повышение стабильности динамических биометрических параметров. В отличие от традиционных методов сжатия данных, основанных на учете их статистики и энтропии, разработан алгоритм на основе обученной нейронной сети, что позволило выявить закономерности в обрабатываемых данных и оптимизировать процесс компрессии нестабильных биометрических данных.

Практическая значимость предлагаемых решений заключается в комплексировании функций регистрации, верификации и идентификации пользователей с использованием единой биометрической системы, что позволяет повысить эффективность безопасности информации при сокращении выделяемых ресурсов. Проведенное тестирование нейросетевого преобразователя биометрия-код позволило выявить закономерность влияния увеличения вводимого пароля на повышение стабильности идентификатора.

Ключевые слова: аутентификация, биометрия, клавиатурный почерк, информационно-технические воздействия, угрозы безопасности информации

Для цитирования: Винокуров А.В. Структурно-параметрический синтез биометрической системы аутентификации пользователя // Вестник РГГУ. Серия «Информатика. Информационная безопасность. Математика». 2026. № 3. С. 155–165. DOI: 10.28995/2686-679X-2026-3-155-165

Structural and parametric synthesis of a user biometric authentication system

Aleksandr V. Vinokurov

*Russian State University for the Humanities, Moscow, Russia,
VAV73@rambler.ru*

Abstract. Modern requirements for information systems that process restricted information define the need for developing new methods and approaches to user authentication that combine high reliability with usability. The article examines a current approach to creating a biometric user authentication system based on the neural network transformation of dynamic biometric parameters (analysis of password entry on the keyboard) into a unique identifier. A functional model of a highly reliable user authentication system has been constructed. Structural and parametric synthesis of the biometric user authentication system is carried out by integrating convolutional neural networks, recurrent layers, and an encoder, which together enhance the stability of dynamic biometric parameters.

Unlike traditional data compression methods based on their statistics and entropy, an algorithm based on a trained neural network is proposed, which allows for identifying patterns in processed data and optimizing the compression process of unstable biometric data. The practical significance of the proposed solutions lies in integrating the functions of registration, verification, and user identification using a unified biometric system, which enhances information security efficiency while reducing allocated resources. Testing of the NPBC revealed a pattern showing that increasing the length of the entered password improves the stability of the identifier.

Keywords: authentication, biometrics keystroke, dynamics information and technical impacts, information security threats

For citation: Vinokurov, A.V. (2026), “Structural and parametric synthesis of a user biometric authentication system”, *RSUH/RGGU Bulletin. “Information Science. Information Security. Mathematics” Series*, no. 3, pp. 155–165, DOI: 10.28995/2686-679X-2026-3-155-165

В современных информационных системах (ИС), обрабатывающих информацию ограниченного распространения, должны

выполняться требования по санкционированному доступу пользователей к их информационным ресурсам.

Традиционные методы аутентификации пользователей, основанные на вводе пароля и идентификатора, обладают рядом недостатков, включая сложность использования (запоминания) пароля и высокую вероятность его компрометации, что определяет перспективу биометрической аутентификации. Одним из решений задачи сохранения в тайне биометрических данных пользователей является применение нейросетевого преобразователя биометрия-код (НПБК), который представляет собой заранее обученную искусственную нейронную сеть с большим числом входов и выходов, преобразующую частично случайный вектор входных биометрических параметров «Свой» в однозначный код криптографического ключа (длинного пароля) [Ахметов и др. 2014, с. 8]. В работе [Власов, Винокуров, Бочаров 2023, с. 1] предложены алгоритмы обучения НПБК и формирования идентификатора пользователя ИС с учетом рекомендаций ГОСТ Р 52633.0-2006, отличительными признаками которого является хранение в биометрических контейнерах таблицы связей и весовых коэффициентов обученной искусственной нейронной сети вместо шаблона биометрического образа пользователя ИС.

Анализ возможных для использования в настоящее время биометрических образов по показателям стойкости к атакам подбора, сложности реализации и ресурсоемкости позволил обосновать выбор применения динамических биометрических образов, полученных на основе клавиатурного почерка [Винокуров, Власов 2024, с. 170].

Обладая рядом преимуществ метод клавиатурного почерка, характеризуется высокой нестабильностью выходных данных. Увеличение количества ошибок первого рода на выходе НПБК возможно по следующим причинам:

- некорректная аппроксимация закона распределения входных биометрических параметров;
- недостаточная или смещенная обучающая выборка;
- нестабильность ввода данных на клавиатуре пользователями, в том числе с течением времени;
- погрешность функций вычисления биометрических параметров (ошибки округления на предварительном этапе обработки).

С учетом выявленных особенностей целью статьи является построение системы высоконадежной аутентификации пользователей ИС, определение структуры НПБК и разработка алгоритма идентификации пользователя по клавиатурному почерку. Под вы-

соконадёжной биометрической аутентификацией в работе понимается биометрическая аутентификация с приемлемой вероятностью ошибок первого рода и гарантированно малой вероятностью ошибок второго рода, сопоставимой по своему значению с вероятностью случайного подбора кода неизвестного криптографического ключа при малом числе попыток подбора [Ахметов и др. 2014, с. 7].

Функциональная модель предлагаемой к разработке системы высоконадёжной аутентификации пользователя ИС представлена на рис. 1.

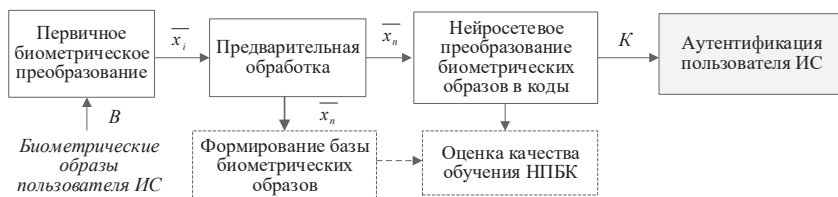


Рис. 1. Функциональная модель системы высоконадёжной аутентификации пользователя

Математическое описание модели (см. рис. 1).

Исходные данные:

B – биометрические параметры пользователя ИС;

S – пароль пользователя ИС;

K_d – уникальный идентификатор пользователя, где $K_d = (k_1, \dots, k_{256})$, $k_j \in \{0, 1\}$ – последовательность бит.

В представленной модели предусмотрены следующие операции.

1. Первичное биометрическое преобразование. Функция первичного биометрического преобразования $f_{\text{поп}}: B_i \rightarrow \bar{x}_i$ заключается в преобразовании биометрических образов пользователя ИС в цифровую форму.

2. Предварительная обработка. Функция предварительной обработки $f_{\text{по}}: \bar{x}_i \rightarrow \bar{x}_n$, где n – количество входов НПБК, заключается в вычислении биометрических параметров (осуществляется сглаживание, масштабирование, вычисление контролируемых биометрических параметров) по биометрическим образам, представленным в цифровой форме. При необходимости повышения стойкости системы к атакам противника применяются операции объединения биометрических признаков по входам НПБК или объединения биометрических образов с выходов НПБК.

3. Нейросетевое преобразование биометрических образов в коды и формирование идентификатора пользователя ИС.

Результат нейросетевого преобразования:

$$K_o = \varphi_{\text{нсп}}(\bar{x}_n),$$

где $\bar{x}_n$ – вектор входных биометрических параметров, $512 \leq n \leq 2048$;
 $\varphi_{\text{нсп}}$ – функция нейросетевого преобразования.

Достоинство предлагаемого построения биометрической системы на основе анализа клавиатурного почерка, заключающееся в совмещении процедур парольной и биометрической аутентификации, приводит к тому, что общий уровень защиты от несанкционированного доступа к ИС нарушителя (ошибки второго рода) будет определяться сразу тремя факторами:

- способностью системы высоконадежной аутентификации сформировать уникальный идентификатор пользователя ИС при использовании контрольной фразы;
- «секретностью» контрольной фразы;
- «секретностью» факта наличия биометрического контроля.

В этом случае реализуется трехфакторная парольно-клавиатурная система аутентификации. Первые два фактора определяют уровень защиты в условиях, когда злоумышленник знает о наличии встроенного клавиатурного контроля. Для осуществления несанкционированного доступа в этих условиях злоумышленнику необходимо организовать и совместить две трудоемкие процедуры: подбора пароля и имитацию клавиатурного почерка легального пользователя ИС. Третий фактор оградит от доступа к ИС случайного нарушителя, не осведомленного с системой анализа клавиатурного почерка.

В ГОСТ Р 52633.5-2011 рассматривается построение НПБК на основе двухслойной нейронной сети, первый слой которой состоит из M нейронов, каждый из которых связан с одним или несколькими из N входных биометрических параметров. Одним из обоснований построения двухслойной сети является необходимость решить задачу идентификации оператора при недостаточном качестве исходных биометрических данных, однако без применения дополнительных механизмов повышения стабильности динамических биометрических параметров, требуемые значения вероятностей ошибок первого и второго рода не достигнуты.

Структура нейронной сети для идентификации по клавиатурному почерку должна учитывать особенности исходных данных и специфику задачи. Так как набор текста представляет собой процесс, связанный с последовательностью нажатий клавиш, то обосновано

применение рекуррентной нейронной сети. Обработка временных характеристик осуществляется НПКБ после их предварительного разложения, например, при помощи вейвлет-преобразований по ортогональному базису функций Хаара, коэффициенты которых используются в качестве признаков [Ахметов и др. 2014, с. 8].

Структура НПКБ, основанная на основе гибридной нейросетевой модели, включает:

1. Слой приема данных.

2. Слой предварительной обработки (предназначен для преобразования первичных дискретных данных в кодовый вектор. Осуществляется предварительная очистка и сортировка данных с целью уменьшения нагрузки на основные слои нейросети и сокращения времени на ее обучение. Решается задача выявления нелинейных зависимостей и автоматической фильтрации случайных выбросов, возникающих из-за условий ввода данных пользователем.

3. Основные слои (гибридная модель сверточной нейросети с рекуррентными слоями).

Основным компонентом модели является сверточная нейросеть в комбинации с рекуррентными слоями, что обеспечивает выявление временных зависимостей и аномалий в последовательности нажатий клавиш и фильтрацию выбросов. Завершается конфигурация нейросети полносвязными слоями.

4. Слои, реализующие функцию энкодера.

После прохождения рекуррентных слоев данные попадают в полносвязные слои, задачей которых является формирование компактного и информативного представления данных. Отличительным признаком построения выходного слоя является то, что он не реализует функцию классификации, производящую вероятностное распределение по классу пользователя, а предназначен для сжатия данных, обеспечивая формирование уникального идентификатора пользователя.

Для реализации рассмотренных выше подходов предлагается использование подгоночного энкодера, являющегося одним из видов нейронных сетей, используемых для сжатия и извлечения наиболее важной информации за счет меньшей размерности скрытого слоя, чем входной. Подтверждение получения оптимальных параметров энкодера осуществляется методом градиентного спуска.

Входные и выходные данные:

- размер входной последовательности – 180 интервалов;
- количество анализируемых признаков – 2 (длительность нажатия, интервал паузы между нажатиями клавиш);
- размер идентификатора K – 256 бит.

Основные параметры модели приведены в табл. 1.

Таблица 1

Основные параметры модели

Название слоев	Количество нейронов (фильтров)	Функция активации	Другие параметры
Свёрточный	64	ReLU	параметр регуляризации – 0,5 размер матрицы фильтра (ядра) – 3
Рекуррентный с памятью:	128	ReLU	
Полносвязные	128 (первый слой) 64 (второй слой) 32 (третий слой)	ReLU	

Многослойные нейронные сети повышают точность детектирования и классификации и находят применение в различных областях, в том числе для распознавания объектов на местности [Мицевич, Тассов, Горбунов 2024].

На основании предложенной модели разработан алгоритм формирования уникального идентификатора пользователя:

1. Сбор данных.

Пользователи многократно вводят персональный пароль S . Сбор данных включает регистрацию кода каждой клавиши, моментов начала и окончания их нажатия, что формирует последовательность временных отметок для последующего анализа.

2. Выделение признаков.

Из собранных данных выделяются уникальные признаки, отражающие индивидуальный стиль пользователя. В качестве признаков используются:

- τ_i – значение длительности удержаний всех учитываемых клавиш i ;
- $t_{i,i+1}$ – значение длительности паузы между удержаниями соседних клавиш i и $i + 1$;
- s_i – кодировка (например, *ASCII*) контролируемых клавиш i .

Результат регистрации всех контролируемых клавиш $i = 1, 2, ..., n$ отображается следующими одномерными матрицами:

$T_y = [\tau_1, \tau_2, ..., \tau_n]$ – вектор длительностей удержания всех контролируемых клавиш;

$T_n = [t_1, t_2, \dots, t_{n-1}]$ – вектор длительностей пауз между всеми контролируруемыми клавишами;

$S = [s_1, s_2, \dots, s_n]$ – вектор кодировки контролируемых клавиш (на вход нейронной сети не подается).

3. Нормализация и предобработка.

Поскольку динамические биометрические данные на основе клавиатурного почерка обладают высоким уровнем нестабильности, осуществляется их предварительная обработка, фильтрация и нормализация, в частности, вычисляется отклонение от средних значений длительности нажатий клавиш и пауз между их нажатиями, определяется доверительный интервал $\pm 15\%$ от среднего значения.

4. Обучение модели.

Для выявления закономерности в поведенческих данных пользователей и формирования «нормального» клавиатурного почерка (эталонного биометрического шаблона) осуществляется обучение НПБК с использованием метода обратного распространения ошибки. Целью обучения является сокращение расстояния Хэмминга последовательности биометрических данных пользователя при повторяющемся вводе парольной фразы.

5. Генерация идентификатора.

На основании методов снижения размерности полученного вектора и использования подгоночного автокодера генерируется уникальный цифровой идентификатор размером 256 бит, соответствующий каждому пользователю.

6. Тестирование и проверка.

Перед началом эксплуатации средства высоконадежной биометрической аутентификации осуществляется его тестирование и проверка на тестовых данных, чтобы убедиться в стабильности формирования идентификатора. По результатам тестирования НПБК были получены следующие показатели, приведенные в табл. 2.

Таблица 2

Результаты тестирования НПБК

Номер оператора, j	Номер попытки ввода пароля, i	Количество ошибок в полученном идентификаторе при:	
		$S = 64$ бит	$S = 192$ бит
1	1	2	1
	2	1	0
2	1	2	1
	2	3	2
3	1	2	1
	2	2	0
4	1	2	1
	2	3	2
5	1	2	1
	2	3	2

Заключение

В статье предложено формирование уникального идентификатора пользователя ИС на основе нейросетевой обработки динамических биометрических параметров. Данный подход позволяет обеспечить многофакторную идентификацию и разграничение доступа пользователей ИС без необходимости сохранения в тайне содержания парольной фразы. Обоснована структура НПБК, отличительной особенностью которой является применение рекуррентной нейронной сети для обработки биометрических параметров с учетом последовательности нажатий клавиш и интервалов между ними.

Разработан алгоритм сжатия (устранения нестабильных данных) при анализе клавиатурного почерка на основании энкодера, что позволило повысить стабильность выходных данных и как результат, точность и надежность системы идентификации. В отличие от классических методов сжатия данных, основанных на учете их статистики и энтропии, таких как кодирование Хаффмана или арифметическое кодирование, предлагаемый алгоритм на основе обученных нейронных сетей позволяет выявить закономерности в обрабатываемых данных и оптимизировать процесс компрессии.

Практическая значимость предлагаемых решений заключается в комплексировании функций регистрации, верификации и идентификации пользователей ИС с использованием единой биометрической системы, что позволяет повысить эффективность функционирования системы безопасности информации при сокращении выделяемых ресурсов. Интерпретация результатов тестирования показала повышение стабильности выходных данных НПБК при увеличении длины вводимого пароля.

Направлением дальнейших исследований является проведение имитационного моделирования на увеличенной выборке пользователей и внесение внешних воздействий с целью корректировки варьируемых параметров нейросети, таких как количество нейронов в рекуррентном слое, количество полносвязных слоев и нейронов, а также порога принятия решений классификатором для минимизации пропусков уникальных биометрических параметров пользователя без увеличения количества повтора набора парольной фразы.

Литература

- Ахметов и др. 2014 – *Ахметов Б.С.* Технология использования больших нейронных сетей для преобразования нечетких биометрических данных в код ключа доступа: монография / Б.С. Ахметов, А.И. Иванов, В.А. Фунтиков, А.В. Безяев, Е.А. Малыгина. Алматы: ТОО «Издательство LEM», 2014. 144 с.
- Винокуров, Власов 2024 – *Винокуров А.В., Власов К.О.* Комплексное применение статических и динамических биометрических параметров для обеспечения оперативной и безопасной аутентификации операторов автоматизированных систем управления специального назначения // Состояние и перспективы развития современной науки по направлению «Робототехника»: Сб. ст. VI Всероссийской научно-технической конференции, Анапа, 16–18 июля 2024 года. Анапа: Военный инновационный технополис «ЭРА», 2024. С. 170–180. EDN ZQTFBV.
- Власов, Винокуров, Бочаров 2023 – *Власов К.О., Винокуров А.В., Бочаров Н.А.* Модели и алгоритмы высоконадежной аутентификации пользователей информационных систем // Приборы. 2023. № 1 (271). С. 1–8. EDN KMWSYF.
- Мицевич, Тассов, Горбунов 2024 – *Мицевич М.Д., Тассов К.Л., Горбунов Д.В.* Метод распознавания летательных аппаратов по аэрофотоснимкам с использованием нейронных сетей // Вестник РГТУ. Серия: Информатика. Информационная безопасность. Математика. 2024. № 4. С. 8–22. DOI: 10.28995/2686-679X-2024-4-8-22. EDN GJIVAT.

References

- Akhmetov, B.S., Ivanov, A.I., Funtikov, V.A., Bezyaev, A.V. and Malygina, E.A. (2014), *Tekhnologiya ispol'zovaniya bol'shikh neyronnykh setey dlya preobrazovaniya nechetkikh biometricheskikh dannykh v kod klyucha dostupa: monografiya* [Technology of using large neural networks for transforming fuzzy biometric data into an access key code: monograph], LEM Publishing House LLP, Almaty, Kazakhstan.
- Mitsevich, M.D., Tassov, K.L. and Gorbunov, D.V. (2024), "A method for recognizing aircraft in aerial photographs using neural networks", *RSUH/RGGU Bulletin, "Information Security. Information Security. Mathematics" Series*, no. 4, pp. 8–22.
- Vinokurov, A.V. and Vlasov, K.O. (2024), "Integrated use of static and dynamic biometric parameters for operational and secure authentication of automated control system operators", *Sostoyaniye i perspektivy razvitiya sovremennoy nauki po napravleniyu "Robototekhnika"* [Status and Prospects of Modern Science in the Field of "Robotics"]. Proceedings of the 6th All-Russian Scientific and Technical Conference, 16–18 July 2024, Anapa, The military innovative technopolis "ERA", Anapa, Russia, pp. 170–180.
- Vlasov, K.O., Vinokurov, A.V. and Bocharov, N.A. (2023), "Models and algorithms for highly reliable user authentication in information systems", *Pribery*, no. 1 (271), pp. 1–8.

Информация об авторе

Александр В. Винокуров, доктор технических наук, доцент, Российский государственный гуманитарный университет, Москва, Россия; 125047, Россия, Москва, Миусская пл., д. 6, стр. 6; VAV73@rambler.ru

Information about the author

Aleksandr V. Vinokurov, Dr. of Sci. (Computer Science), associate professor, Russian State University for the Humanities, Moscow, Russia; 6-6, Miusskaya Sq., Moscow, 125047, Russia; VAV73@rambler.ru

Научный журнал
Вестник РГГУ
Серия «Информатика.
Информационная безопасность. Математика»
№ 3
2026

Дизайн обложки
Е.В. Амосова

Корректор
Ж.П. Григорьева

Компьютерная верстка
Н.В. Москвина

Учредитель и издатель
Российский государственный гуманитарный университет
125047, г. Москва, вн. тер. г. муниципальный округ Тверской,
Миусская пл., д. 6, стр. 6

Свидетельство о регистрации СМИ
ПИ № ФС77-72977 от 25.05.2018 г.
Периодическое печатное издание

Подписано в печать 07.10.2026

Выход в свет 14.10.2026

Формат 60 × 90 ¹/₁₆

Уч.-изд. л. 9,0. Усл. печ. л. 10,4

Тираж 1050 экз. Свободная цена

Заказ № 2422

Отпечатано в типографии Издательского центра
Российского государственного гуманитарного университета
125047, Москва, Миусская пл., д. 6, стр. 6
www.rsuh.ru