Санкт-Петербургский
политехнический университет Петра Великого

Проблемы информационной безопасности.

Компьютерные системы

№ 2 • 2026

Problems of information security.

Computer systems

No. 2 • 2026

Санкт-Петербург
2026

№ 2 · 2026

Журнал является органом Совета Регионального Северо-Западного учебно-научного центра информационной безопасности

Журнал включен в перечень изданий, утвержденных ВАК, для публикации основных результатов диссертационных исследований

Цель журнала – популяризация результатов актуальных научных исследований в сфере обеспечения безопасности информационных инфраструктур, исследования автоматизированных систем управления технологическими процессами и производствами, а также оценки качества и сопровождения программных продуктов

УЧРЕДИТЕЛЬ И ИЗДАТЕЛЬ

Санкт-Петербургский политехнический университет Петра Великого

КОНТАКТЫ:

195251, Санкт-Петербург, ул. Политехническая, 29

Тел. +7 (812) 552-76-32

E-mail: ojs@ibks.spbstu.ru

Сайт журнала: https://jisp.spbstu.ru/

Свидетельство о регистрации **№ 018607** от 17.03.99

С 1 января 2019 г. подписка на журнал «Проблемы информационной безопасности. Компьютерные системы» осуществляется через объединенный каталог «Пресса России»

https://www.pressa-rf.ru

Подписной индекс – Т18237

С 2025 г. доступ к электронной версии предоставляется через ЭБС Руконт https://rucont.ru/

Цена свободная

Выпускающий редактор **В. Е. ФИЛИППОВА**

Ответственный секретарь **Н. Ю. ЛОВЧИНОВСКАЯ**

No. 2 · 2026

The journal is a body
of the Council of the Regional
North-West Educational and Scientific Center
for Information Security

The journal is included in the list
of editions approved by the Higher Attestation
Commission for the publication of the main
results of dissertation research

The purpose of the journal is to popularize
the results of current scientific research
in the field of information infrastructure
security, research of automated process
and production control systems,
as well as quality assessment
and maintenance of software products

FOUNDER AND PUBLISHER

Peter the Great
St. Petersburg Polytechnic University

CONTACTS:

29, Polytechnicheskaya str.,
St. Petersburg, 195251

Tel. +7 (812) 552-76-32

E-mail: ojs@ibks.spbstu.ru

Journal website: https://jisp.spbstu.ru/

Registration Certificate
No. 018607 dated 17.03.99

From January 1, 2019 subscription to the
journal "Problems of Information Security.
Computer Systems" is available through
the united catalog "Press of Russia"

https://www.pressa-rf.ru

Subscription index – T18237

From 2025 access to the electronic version is
provided through EBS Rucont
https://rucont.ru/

Free price

МиТСОБИ

Конференция **«Методы и технические средства обеспечения безопасности информации» (МиТСОБИ)** — это встреча профессионалов информационной безопасности, единственная и старейшая конференция, с 1991 г. ежегодно проходящая в Санкт-Петербурге.

МиТСОБИ — это возможность узнать самые современные направления и поделиться опытом. Это интересные доклады и горячие дискуссии, в которых молодые разработчики могут узнать мнение мэтров информационной безопасности, а руководители — выяснить, как на практике решать самые острые вопросы, оценить важность и действенность этих решений для обеспечения информационной безопасности как страны в целом, так и для каждого участника киберпространства. Особенность конференции — это диалог на пересечении теории и практики, науки и бизнеса.

Ежегодное количество участников — до 300 человек, среди которых руководство и специалисты органов государственной власти РФ, вузов, академических учреждений, разработчики и молодые ученые, представители научно-исследовательских организаций и коммерческих предприятий из различных регионов России.

Организаторы конференции

Комитет по информатизации и связи Правительства Санкт-Петербурга

Комитет по науке и высшей школе Правительства Санкт-Петербурга

СЗРО УМО по ИБ при СПбПУ

Российская Академия Наук

Научный совет по комплексной проблеме «Информационная безопасность»

АЗИ

МОО «Ассоциация Защиты Информации»

При участии

Федеральной службы безопасности РФ,
Федеральной службы охраны РФ,
Федеральной службы по финансовому мониторингу,
Федеральной службы по техническому и экспортному контролю

www.mitsobi.ru | mitsobi@neobit.ru

8 (800) 222-28-06 | +7 (812) 535-28-06

Содержание

МЕТОДЫ И СРЕДСТВА ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

БЕЗОПАСНОСТЬ РАСПРЕДЕЛЕННЫХ СИСТЕМ И ТЕЛЕКОММУНИКАЦИЙ

БЕЗОПАСНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

СИСТЕМЫ МАШИННОГО ОБУЧЕНИЯ И УПРАВЛЕНИЯ БАЗАМИ ЗНАНИЙ

ПРАКТИЧЕСКИЕ АСПЕКТЫ КРИПТОГРАФИИ

Contents

INFORMATION SECURITY APPLICATION

NETWORK AND TELECOMMUNICATION SECURITY

SOFTWARE SECURITY

MACHINE LEARNING AND KNOWLEDGE CONTROL SYSTEMS

APPLIED CRYPTOGRAPHY

Научная статья
DOI 10.66424/2071-8217-2026-2-1
УДК 343.34

ТРАНСФОРМАЦИЯ ПАРАДИГМЫ КИБЕРУГРОЗ: DEEPFAKE КАК ДЕТЕРМИНАНТА ЭСКАЛАЦИИ РИСКОВ В СОЦИАЛЬНО-ИНЖЕНЕРНЫХ АТАКАХ

А. Н. Киселёв*, В. С. Бондаренко, Д. Г. Татаренко

Военно-космическая академия имени А. Ф. Можайского, Санкт-Петербург, Россия

✉ *kan534@mail.ru

ДЛЯ ЦИТИРОВАНИЯ

Киселёв А. Н., Бондаренко В. С., Татаренко Д. Г. Трансформация парадигмы киберугроз: deepfake как детерминанта эскалации рисков в социально-инженерных атаках // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 9–21. DOI: 10.66424/2071-8217-2026-2-1

ПОСТУПИЛА 20.02.2026
ПРИНЯТА 06.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Исследуется как технологии искусственного интеллекта, прежде всего deepfake, изменяют природу социально-инженерных атак. Если раньше злоумышленники эксплуатировали доверчивость атакуемых через текст и голос, то сегодня они способны имитировать биометрические и поведенческие характеристики личности в реальном времени, нивелируя различия между реальностью и подделкой. На основе данных 2024–2026 гг. показано, что deepfake превратился из технологического эксперимента в системную угрозу доверию к цифровому контенту. Особое внимание уделено анализу уязвимостей современных методов детекции, которые демонстрируют отставание по эффективности от темпов развития генеративных моделей. Предложен комплекс мер – от многослойной технической защиты до изменений в правовом поле, – позволяющих перейти от реактивной защиты к проактивному контролю.

КЛЮЧЕВЫЕ СЛОВА

Кибербезопасность, deepfake, социальная инженерия, киберугрозы, искусственный интеллект, генеративно-состязательные сети, детекция подделок, меры защиты, дезинформация, экономический ущерб

Original article
DOI 10.66424/2071-8217-2026-2-1

TRANSFORMATION OF THE CYBER THREAT PARADIGM: DEEPFAKE AS A DETERMINANT OF RISK ESCALATION IN SOCIAL ENGINEERING ATTACKS

A. N. Kiselev*, V. S. Bondarenko, D. G. Tatarenko

Mozhaisky Military Space Academy, St. Petersburg, Russia

✉ *kan534@mail.ru

FOR CITATION

Kiselev A. N., Bondarenko V. S., Tatarenko D. G. Transformation of the cyber threat paradigm: deepfake as a determinant of risk escalation in social engineering attacks. *Problems of information security. Computer systems*. 2026. No. 2, pp. 9–21.
DOI: 10.66424/2071-8217-2026-2-1
(In Russian)

RECEIVED 20.02.2026
ACCEPTED 06.05.2026
PUBLICATION 15.06.2026

ABSTRACT

The article explores how artificial intelligence technologies, particularly deepfake, are fundamentally altering the nature of social engineering attacks. Whereas attackers previously exploited human gullibility through text and voice, they can now mimic biometric and behavioral characteristics in real time, blurring the line between authenticity and deception. Drawing on data from 2024–2026, the study demonstrates that deepfake has evolved from a technological experiment into a systemic threat to the credibility of digital content. Special attention is devoted to the vulnerabilities of current detection methods, which are shown to lag significantly behind the rapid advancement of generative models. The paper proposes a comprehensive set of measures – ranging from multi-layered technical safeguards to legal reforms – designed to shift the cybersecurity paradigm from reactive defense to proactive control.

KEYWORDS

Cybersecurity, deepfake, social engineering, cyber threats, artificial intelligence, generative adversarial networks, fake detection, protection measures, disinformation, and economic damage

1. ВВЕДЕНИЕ

Социальная инженерия всегда была искусством манипуляции, но до недавнего времени ее инструментарий ограничивался словами. Позвонить от имени банка, написать письмо от «руководства» – все это требовало лишь психологической подготовки и минимальных технических знаний. Ситуация изменилась с появлением deepfake – технологий синтеза медиа на основе глубокого обучения [1]. Сегодня атакующий может не просто представиться другим человеком, а быть им на экране видеозвонка. Это ставит под сомнение то, что раньше казалось незыблемым: deepfake бросает вызов самим основам доверия к аудиовизуальной информации, поскольку воспринимаемые зрительно и на слух данные более не могут считаться доказательством реальности [2, С. 1760].

Цель работы – на основе системного анализа инцидентов 2022–2026 гг. и сравнительной оценки современных методов детекции разработать прогностическую модель эволюции deepfake-угроз и сформировать научно обоснованный комплекс мер защиты, адаптированный к выявленным технологическим и организационным уязвимостям.

Научная новизна исследования заключается:

• в выполненной систематизации данных о deepfake-атаках в российском и глобальном сегментах за 2022–2026 гг. с выявлением тренда перехода от разовых инцидентов к гибридным кампаниям;

• количественном обосновании неэффективности современных детекторов в условиях реальной эксплуатации (на основе метаанализа отчета CSIRO [3]);

• разработке модели эволюции фишинга под воздействием deepfake (табл. 1), верифицированной данными открытых источников [4, 5];

• предложении ранжированного комплекса защитных мер с оценкой их ресурсоемкости и ограничений (табл. 2).

Актуальность темы сложно переоценить: в условиях цифровизации всех сфер жизни deepfake становится угрозой не только для бизнеса, но и для политической стабильности, правосудия и личной безопасности граждан. Как отмечается в современных исследованиях, игнорирование этой тенденции способно снизить эффективность предпринимаемых усилий по развитию цифровой экономики [6, С. 160].

2. МЕТОДОЛОГИЯ ИССЛЕДОВАНИЯ

Для достижения цели исследования применен комплекс методов, включающий

Таблица 1 | Сравнительная характеристика эволюции фишинговых атак

Table 1 | Comparative characteristics of the evolution of phishing attacks

Критерий	Традиционный фишинг (до 2020 г.)	Deepfake-атаки (2025–2026)
Масштабируемость	Требование ручной кастомизации, высокие затраты времени	Автоматическая генерация под любой контекст за минуты [5]
Уровень доверия	Низкий: орфографические ошибки, подозрительные ссылки	Высокий: видео- и аудиодоказательства «присутствия»
Барьеры входа	Требование навыков программирования или фишинговых наборов	Существуют как платные сервисы Deepfake-as-a-Service (от 150 дол./мес.), так и бесплатные open-source реализации (Roop, FaceSwap, Google Colab) [7]
Устойчивость к защите	Блокировка спам-фильтрами, двухфакторной аутентификацией	Обходит отдельные типы биометрических систем (например, основанные на анализе движения губ)
Объект воздействия	Текст, реже голос	Мультимодальный контент (видео + аудио + текст)
Экономический эффект	Локальные убытки, редко >1 млн дол. США	Единичный случай до 25 млн дол. США (Arup, 2024 г.), но медианный ущерб по данным Proofpoint (2025) составляет 50–200 тыс. дол. США [8]
Вычислительные ресурсы для атаки	Минимальные (ПК, офисное ПО)	Требуют GPU (от одной видеокарты уровня RTX 3060) для генерации в реальном времени; для пакетной генерации достаточно менее мощных GPU с 8+ ГБ видеопамяти

Таблица 2 | Сравнительная оценка предлагаемых мер защиты

Table 2 | Comparative assessment of the proposed protection measures

Мера защиты	Эффективность против диффузионных моделей	Сложность внедрения	Ограничения и векторы обхода
Ансамблевые детекторы	Средняя (требуется постоянное дообучение) [3]	Высокая	Уязвимы к состязательным атакам; задержка при анализе потокового видео
Верификация происхождения (C2PA)	Высокая (превентивная)	Средняя	Требует поддержки платформами; не защищает от подделок на уровне захвата до подписания
Объяснимый ИИ	Не повышает точность, но увеличивает доверие	Средняя	Субъективность интерпретации; возможность манипуляции визуализацией
Реалистичные бенчмарки	Косвенная (стимулирует улучшение моделей)	Низкая	Не решает проблему «здесь и сейчас»
Межотраслевой обмен **хешами**	Высокая (быстрое блокирование известных фейков)	Средняя	Зависит от оперативности участников; не детектирует новые подделки

Примечание: количественная оценка затрат не приводится ввиду сильной зависимости от конкретной ИТ-архитектуры, региона и вендора. Для приблизительного расчета рекомендуется обращаться к актуальным коммерческим предложениям системных интеграторов.

систематический обзор научной литературы, количественный анализ временных рядов инцидентов, качественный контент-анализ описаний реальных атак и сравнительное моделирование эволюции угроз.

Этап 1. Систематический обзор литературы. По ключевым словам «deepfake», «social engineering», «generative adversarial networks», «deepfake detection» в базах Scopus, IEEE Xplore, eLibrary.Ru за период 2014–2026 гг. первично отобрано 187 публикаций. После удаления дубликатов ($n=42$) и скрининга по аннотациям ($n=125$) для полного текстового анализа отобрано 62 работы. Из них 24 публикации соответствовали критериям включения (наличие эмпирических данных или формализованной модели угрозы с привязкой к социальной инженерии) и использованы для составления классификации. На основе обзора составлена классификация архитектур генерации deepfake и методов детекции [1, 9, 5].

Этап 2. Количественный анализ инцидентов. Использованы открытые отчеты CSIRO (2025) [3], Sensity AI (2024–2025) [10], Proofpoint (2024) [4], ENISA (2025) [11], Statista (2024) [12] и данные мониторинга АНО «Диалог Регионы» (2025) [13, 14]. Извлечены временные ряды числа уникальных deepfake-файлов, финансового ущерба и темпов роста атак за 2022–2026 гг. Для проверки репрезентативности данные сопоставлялись не менее чем по двум независимым источникам; расхождения более 15 % считались критическими и исключались из анализа.

Этап 3. Качественный контент-анализ кейсов. Отобрано 10 резонансных инцидентов 2022–2025 гг., удовлетворяющих критериям: подтвержденное использование deepfake; документально зафиксированные последствия (финансовые, политические, технологические).

Для каждого кейса выделены: тип атаки, канал воздействия, метод создания подделки, уязвимость защиты и экономический ущерб.

Этап 4. Сравнительное моделирование. Построена сравнительная таблица эволюции фишинга (см. табл. 1) путем сопоставления характеристик атак до 2020 г. и в период 2025–2026 гг. на основе данных, систематизированных в работах [4, 5, 15].

Этап 5. Синтез рекомендаций. На основе выявленных уязвимостей методов детекции [3, 16] и анализа регуляторных инициатив предложен комплекс мер, ранжированный по критериям «затраты – эффективность – реализуемость».

3. АКТУАЛЬНОСТЬ ПРОБЛЕМЫ: ОТ СТАТИСТИКИ К ТЕНДЕНЦИЯМ

Если в 2022 г. deepfake воспринимались как забава для создания порнографических видео со знаменитостями, то к 2025–2026 гг. они стали полноценным инструментом киберпреступности. Особенно показателен российский сегмент: согласно данным АНО «Диалог Регионы», за январь – сентябрь 2025 г. выявлено 342 уникальных deepfake – в 4,1 раза больше, чем за весь 2024 г. [13]. Рост начался в апреле, а в сентябре 2025 г. зафиксировано максимальное за период наблюдения значение – 65 случаев [14]. Примечательно, что 79 % фейков имитируют глав регионов и госслужащих – мишенью становится именно власть, что указывает на политический заказ, а не просто бытовое мошенничество.

Глобальная картина дополняет тревожные сигналы. Sensity AI (ныне Sentinel) насчитала 231 уникальный политический deepfake только за 2025 г., причем их копии разошлись тиражом в 29 тыс. [10]. Информационные операции с использованием синтетического видео набрали 84,5 млн просмотров [4]. По данным ENISA (опубликованным в октябре 2025 г.), deepfake-атаки входят в пятерку наиболее быстрорастущих угроз в Европе [11].

Значимость проблемы определяется не только частотой, но и тяжестью последствий. Финансовый ущерб от крупных deepfake-инцидентов в первом квартале 2025 г. превысил 200 млн дол. США [8], а совокупные потери от ИТ-мошенничества достигли 12,5 млрд дол. США в 2024 г., и по прогнозам Statista к 2027 г. эта цифра вырастет до 40 млрд дол. США

(CAGR 30 %) [12]. На наш взгляд это лишь видимая часть проблемы: значительная часть атак остается нераскрытой, а пострадавшие компании предпочитают не разглашать факты, чтобы сохранить репутацию.

Почему deepfake так эффективен? Потому что направлен на наиболее уязвимый элемент системы – доверие к аудиовизуальной информации. Человек привык верить своим глазам и ушам, и когда подделка технически безупречна, критическое мышление отключается. В этом смысле deepfake работает как инструмент воздействия на когнитивные процессы, подрывающий сами основы коммуникации.

Масштаб угрозы становится очевидным при взгляде на количественные показатели. К 2025 г. количество deepfake-файлов в социальных сетях достигло 8 млн дол. США, а ежегодный прирост составляет 900 %. Экспоненциальный рост объема синтезированных материалов уже сейчас воспринимается профессиональным сообществом как одна из главных опасностей, ведь 64 % экспертов по кибербезопасности относят deepfake к числу топ-угроз. Бизнес также начинает осознавать уязвимость, так как почти треть предприятий (30 %) признают, что традиционные и биометрические методы аутентификации не обеспечивают надежной защиты от атак с использованием дипфейков. И темпы роста подтверждают обоснованность этих опасений – за последние три года число таких атак увеличилось на 2000 % при низкой базе 2022 г., что объясняет столь высокий относительный рост. Иначе говоря, наблюдается не просто количественный скачок, а качественная трансформация угрозы: то, что вчера было экзотикой, сегодня становится стандартным инструментом злоумышленников.

4. ТЕХНИЧЕСКИЕ ОСНОВЫ И АРХИТЕКТУРЫ ГЕНЕРАЦИИ DEEPFAKE

Deepfake (от deep learning + fake) – это технология синтеза медиа, основанная на нейросетях. В отличие от традиционного монтажа, здесь не требуется участие человека: алгоритм самостоятельно обучается на тысячах фотографий или часов видео, а затем генерирует новые сцены с нужной персоной. Ключевую роль играют генеративно-состязательные сети (GAN). Суть этого подхода, заложенного еще в основополагающей работе Гудфеллоу с соавторами, заключается в «соревновании» двух моделей: генератор G учится создавать реалистичный контент, а дискриминатор D пытается отличить подделку от оригинала [1, С. 2673].

Современные методы синтеза поддельного медиаконтента можно разделить на три основных класса [5, 9]:

1. Генеративно-состязательные сети (GAN) – наиболее распространенный подход для замены лиц (face swap) и полной генерации изображений. Архитектуры типа StyleGAN2/3, FaceSwap-GAN обеспечивают высокую реалистичность, но требуют значительных вычислительных ресурсов и крупных обучающих выборок [1, 9].

2. Диффузионные модели (Diffusion Models) – начиная с 2023 г. активно применяются для генерации видео и изображений благодаря лучшей устойчивости к артефактам, однако для замены лица в реальном времени (live deepfake) вычислительные затраты диффузионных моделей пока остаются запретительно высокими, поэтому GAN продолжают использоваться в атаках реального времени. Модели семейства Stable Diffusion, Video Diffusion позволяют создавать синтетические видеоролики с минимальными визуальными искажениями, что затрудняет детекцию по классическим признакам сжатия [9].

3. Нейросетевые вокодеры и системы клонирования голоса – технологии типа WaveNet, Tacotron 2, VALL-E, способные синтезировать речь по образцу длительностью от трех секунд с точностью распознавания естественности >85 % [3].

В контексте социальной инженерии наиболее опасны гибридные атаки, сочетающие видео- и аудиоподмену в реальном времени (например, через API сервисов Deepfake-as-a-Service) [7].

С 2025 г. прорыв произошел в области клонирования голоса (voice cloning). Современные системы (например, VALL-E, YourTTS) позволяют синтезировать речь, неотличимую для человека в 70–85 % случаев при длительности образца от трех секунд в лабораторных условиях (метрика MOS). Однако в реальных акустических каналах (шум, сжатие кодеком) эффективность снижается. Это открывает путь к массовым атакам: злоумышленник может собрать образцы голоса из открытых интервью или даже из голосовых сообщений в мессенджерах и создать убедительный фейк за считанные минуты. Интеграция с социальными сетями усугубляет проблему: платформы сжимают и перекодируют видео, что, как показывают исследования, не только маскирует следы подделки, но и затрудняет работу детекторов.

5. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭВОЛЮЦИИ ФИШИНГА ПОД ВЛИЯНИЕМ DEEPFAKE

Чтобы понять глубину изменений, недостаточно просто перечислить характеристики – необходимо проследить как меняется экономика атак с появлением deepfake-технологий [5]. В табл. 1 представлено сопоставление традиционных методов фишинга (до 2020 г.) и современных deepfake-атак (2025–2026) по ключевым критериям, характеризующим эволюцию угрозы

6. ПРИМЕРЫ РЕАЛЬНЫХ КЕЙСОВ

Анализ конкретных инцидентов позволяет увидеть как deepfake из экзотики превращается в рутинный инструмент злоумышленников. Данные случаи разделены на несколько категорий в зависимости от типа воздействия.

Российские инциденты с использованием deepfake в открытых источниках представлены фрагментарно (в основном на уровне новостных заметок без технических деталей), поэтому в данном исследовании приведены международные кейсы.

Финансовые атаки:

• В феврале 2024 г. мошенники с помощью deepfake-видеозвонка, в котором использовались синтезированные лицо и голос нескольких сотрудников (в том числе фейкового финансового директора), инициировали перевод 25 млн дол. США. Инцидент зафиксирован в отчете полиции Гонконга и подтвержден компанией Arup.

• В 2023 г. поддельное интервью Илона Маска на YouTube привело к тому, что тысячи людей перевели криптовалюту мошенникам. Здесь интересна не техника, а психология: авторитет публичной личности работает как триггер немедленного действия.

Политические и гибридные атаки:

• В марте 2022 г. deepfake с президентом Зеленским, призывающим сложить оружие, был распространен в украинских мессенджерах в разгар боев. Хотя подделку быстро распознали, сам факт ее появления создал информационный шум и потребовал официального опровержения.

• В августе 2025 г. компания Storm-1679 использовала фейковые эфиры ABC News, BBC и Netflix для дезинформации о войне в Украине. Видео ретвитили Илон Маск и Дональд Трамп-младший [17]. Это показывает, что даже технически неидеальные подделки могут вирусно распространяться, если они попадают в нужную аудиторию.

• В июне 2025 г. deepfake с Кейром Стармером о повышении налогов набрал 430 тыс. просмотров до того, как был опровергнут. В эпоху клипового мышления первое впечатление часто становится решающим.

• В ноябре 2024 г. deepfake с Олексием Даниловым транслировался на федеральном российском телевидении с фальшивым заявлением об ответственности Украины за теракт, что спровоцировало дипломатический протест.

• В феврале 2024 г. робозвонки с синтезированным голосом Джо Байдена, призывающие избирателей не голосовать на праймериз в Нью-Гэмпшире, привели к официальному расследованию ФКС и установлению прецедента регулирования политических deepfake.

Технологические прецеденты:

• В январе 2025 г. в Нью-Гэмпшире впервые предпринята попытка использовать синтетическое видео с камеры полицейского в качестве доказательства в суде. Как отмечают эксперты: «этот случай создает опасный прецедент: рано или поздно такой фейк может повлиять на приговор, если система верификации не будет кардинально усилена».

• В апреле 2025 г. хакеры заменили аудиосигналы пешеходных переходов в Сиэтле на синтезированный голос Джеффа Безоса с политическими лозунгами. Инцидент уникален тем, что угроза перешла из цифрового пространства в физическое – подделка повлияла на поведение людей в реальном мире.

• В мае 2023 г. в социальных сетях распространено синтетическое изображение взрыва у Пентагона (созданное, предположительно, с помощью генеративных моделей, но не являющееся deepfake в классическом смысле замены лица). Тем не менее этот случай демонстрирует уязвимость финансовых рынков к мгновенной дезинформации: индекс S&P 500 снизился на 0,4 % за 5 мин до официального опровержения.

Каждый из этих случаев подтверждает нашу гипотезу: deepfake становится универсальным оружием, применимым в любой сфере, где требуется имитация личности. Причем эволюция идет от разовых акций к системным кампаниям, координируемым через социальные сети и мессенджеры [15].

7. МЕТОДЫ ДЕТЕКЦИИ DEEPFAKE: КЛАССИФИКАЦИЯ И ОГРАНИЧЕНИЯ

Логично предположить, что против deepfake должны существовать эффективные детекторы. Однако реальность такова, что современные системы обнаружения не справляются с быстротой эволюции генеративных моделей. В марте 2025 г. исследователи CSIRO и университета Сонгюнкван протестировали 16 популярных детекторов и пришли к тревожным выводам, указывающим на необходимость пересмотра существующих подходов: «ни один из них не показывает надежных результатов в реальных условиях» [3, С. 12].

Существующие подходы к обнаружению подделок можно сгруппировать по анализируемым признакам [3, 9]:

• пространственные методы – поиск артефактов генерации в статичных кадрах (несоответствия текстуры кожи, аномалии в области глаз и зубов, следы сглаживания), основаны на сверточных нейросетях (XceptionNet, EfficientNet);

• временно́й анализ – выявление неестественных движений губ, морганий, микромимики; используются рекуррентные сети (LSTM) и 3D-свертки;

• частотный анализ – детекция аномалий в спектре изображения после преобразования Фурье или вейвлет-разложения, показал высокую эффективность против GAN, но уязвим к диффузионным моделям;

• анализ артефактов сжатия – основан на различиях в «шумовом следе» реальных камер и генеративных моделей;

• биометрическая верификация – сравнение микродвижений, уникальных для живого человека (например, паттернов кровотока, отражения света от роговицы).

Однако, как показано в сравнительном исследовании CSIRO (2025), все перечисленные методы демонстрируют резкое падение точности при постобработке видео (сжатие, изменение разрешения) и практически не работают на диффузионных моделях, обученных с аугментацией [3].

Проблема обобщения. Эта проблема усугубляется тем, что детекторы часто не могут справиться даже с незначительными изменениями в техниках создания подделок. Как показали исследования в области компьютерного зрения, «детекторы, обученные на одном типе синтетических изображений, демонстрируют низкую обобщаемость», что делает их бесполезными против новых, еще неизвестных методов атак.

Уязвимость к преобразованиям. Социальные сети сжимают видео, меняют битрейт, добавляют шум – все это снижает точность детекции более чем на 10 %.

А если злоумышленник специально внедрит состязательный шум (adversarial noise), современные алгоритмы могут быть обмануты почти гарантированно.

Взаимное развитие угроз и мер защиты. В сфере кибербезопасности наблюдается классический цикл взаимного совершенствования методов атаки и защиты: как только появляется новый метод обнаружения, злоумышленники адаптируют свои модели, чтобы его обойти. Исследователи фиксируют циклический процесс взаимного влияния как атак, так и защиты. Любой детектор, основанный на машинном обучении, потенциально может стать «учителем» для генератора.

Проблема бинарного ответа. Большинство систем выдают просто «да/нет», не объясняя, на каких признаках основано решение. В юридических и журналистских расследованиях этого недостаточно – нужны доказательства, визуализация областей подделки. Без объяснимого ИИ (XAI) доверие к детекторам остается низким.

8. ПРЕДЛОЖЕНИЯ ПО УЛУЧШЕНИЮ СИСТЕМ ЗАЩИТЫ: ОТ РЕАКТИВНОСТИ К ПРОАКТИВНОСТИ

На основе выявленных недостатков разработан комплекс мер, нацеленный на смещение фокуса с «поимки подделок» на «подтверждение подлинности». Он представляет собой не просто перечень известных практик, а систему ранжированных рекомендаций, учитывающую текущий этап взаимного совершенствования методов атаки и защиты, а также уязвимости, выявленные ранее. Новизна заключается:

• в обосновании приоритетности внедрения проактивных методов (верификация происхождения) над реактивными (детекция артефактов) на основе данных о снижении эффективности последних [3];

• формулировке критериев выбора ансамблевых архитектур в зависимости от типа защищаемого ресурса;

• оценке ресурсоемкости внедрения каждой меры для организаций разного масштаба (см. табл. 2).

Многослойные (ансамблевые) системы. Вместо одного детектора следует использовать комбинацию методов, анализирующих контент через разные «линзы»: артефакты сжатия, несоответствия освещения, аномалии в движении губ, спектральный анализ голоса. Как показано в недавних работах, такой подход создает избыточность и затрудняет обход защиты [16, С. 182].

Верификация происхождения (provenance). Поскольку детекция всегда отстает, нужно внедрять стандарты цифровой подписи контента. Инициатива C2PA (Content Authenticity Initiative) предлагает встраивать в файлы метаданные о происхождении и изменениях. Если контент не имеет такой подписи, он должен рассматриваться как потенциально подозрительный [18].

Объяснимый ИИ. Системы детекции должны не только классифицировать, но и визуализировать области, вызвавшие подозрение. Это позволит экспертам принимать взвешенные решения и повысит доверие к автоматическим инструментам.

Реалистичные бенчмарки. Обучающие наборы данных часто не отражают реальных условий. Необходимы открытые платформы для тестирования детекторов на данных, прошедших через сжатие, шум, изменение размера. Только так можно объективно сравнивать подходы и стимулировать разработку устойчивых моделей.

Межотраслевое сотрудничество. Борьба с deepfake не должна быть задачей только IT-компаний. Нужны безопасные каналы для обмена хешами известных подделок между платформами, правоохранителями и исследователями – по аналогии с тем, как сегодня борются с фишингом.

Данный комплекс мер требует дифференцированного подхода к внедрению, поскольку их эффективность и ресурсоемкость существенно варьируются в зависимости от архитектуры информационной системы и актуального ландшафта

угроз. В частности, методы, основанные на детекции артефактов генерации, демонстрируют высокую чувствительность к эволюции диффузионных моделей, тогда как превентивные подходы, базирующиеся на верификации происхождения контента, обеспечивают более устойчивую защиту, но требуют перестройки процессов создания и распространения медиаданных [3, 16].

Для обоснованного выбора стратегии защиты представим сравнительную оценку предложенных мер (табл. 2). Оценка ресурсоемкости выполнена экспертным путем на основе среднерыночной стоимости лицензий, трудозатрат на интеграцию и эксплуатационных расходов для условной организации с численностью персонала свыше 500 человек. Ограничения и потенциальные векторы обхода систематизированы по результатам анализа академических публикаций [5, 9, 15] и технических отчетов [3].

9. РЕГУЛЯТОРНЫЕ ИНИЦИАТИВЫ

Эффективное противодействие deepfake-угрозам невозможно без формирования адекватной нормативно-правовой базы, которая, с одной стороны, устанавливает ответственность за злонамеренное использование технологий синтеза, а с другой – стимулирует внедрение механизмов верификации контента. Анализ международного опыта показывает наличие трех основных моделей регулирования: превентивная маркировка контента (ЕС, КНР), введение уголовной и административной ответственности за отдельные составы правонарушений (США) и точечные поправки в информационное законодательство (РФ).

Европейский союз. Наиболее системный подход реализован в рамках AI Act, вступившего в силу в 2024 г. Согласно статье 50 AI Act, с 2 августа 2026 г. вводится требование к маркировке синтетического контента при его публичном распространении, за исключением случаев, когда использование deepfake является частью художественного творчества, сатиры или не вводит публику в заблуждение (например, при явном указании на вымышленный характер). Для технической реализации маркировки разрабатывается Кодекс практики [19].

Соединенные Штаты Америки. Регулирование deepfake в США носит фрагментарный характер и развивается преимущественно на уровне отдельных штатов (Калифорния, Техас, Нью-Йорк). Однако наиболее показательным прецедентом федерального реагирования стало решение Федеральной комиссии по связи (FCC) от сентября 2024 г., согласно которому на политического консультанта Стива Крамера наложен штраф в размере 6 млн дол. США за организацию робозвонков с синтезированным голосом президента Дж. Байдена, призывавших избирателей не участвовать в праймериз в Нью-Гэмпшире. FCC признала использование AI-генерированной аудиозаписи для имитации голоса кандидата незаконным в соответствии с Законом о защите прав потребителей телефонной связи [20].

Китайская Народная Республика. Администрация киберпространства Китая (CAC) совместно с Министерством промышленности и информатизации, Министерством общественной безопасности и Государственным управлением радио и телевидения 7 марта 2025 г. утвердила регламент «Меры по маркировке контента, синтезированного с помощью искусственного интеллекта», вступивший в силу 1 сентября 2025 г. Документ вводит обязательную маркировку синтетического контента в двух формах: видимые метки (текст, звук, графика), воспринимаемые пользователем, и скрытые метаданные, содержащие сведения о факте генерации и поставщике услуг. Одновременно введен обязательный национальный стандарт, регламентирующий технические способы реализации маркировки [21].

Российская Федерация. В отечественном правовом поле наблюдается активизация нормотворческой деятельности. 14 ноября 2025 г. группа депутатов во главе с Д. Гусевым внесла в Государственную Думу пакет законопроектов (№ 1069302-8

и 1069331–8), обязывающих владельцев социальных сетей, видеохостингов и других интернет-площадок маркировать видеоматериалы, созданные с использованием технологий искусственного интеллекта. Маркировка должна включать видимый знак («Создано с использованием ИИ» или «Сгенерировано ИИ») на протяжении всего видео и машиночитаемую метку в метаданных, содержащую информацию о факте использования ИИ, дату создания и идентификатор владельца ресурса. Предусмотрены административные штрафы за нарушение требований: для граждан – от 10 до 50 тыс. руб., для должностных лиц – от 100 до 200 тыс. руб., для юридических лиц – от 200 до 500 тыс. руб. По состоянию на февраль 2026 г. законопроект № 1069302-8 находится на рассмотрении в Государственной Думе РФ (первое чтение). Предлагаемые штрафы для юридических лиц – от 200 до 500 тыс. руб. – носят предварительный характер и могут измениться в ходе обсуждения [22].

Проблемы правоприменения. Несмотря на активность законодателей, ключевой проблемой остается низкая эффективность правоприменения, обусловленная трансграничным характером атак и сложностью атрибуции источника подделки. В этой связи перспективным направлением представляется не только введение санкций, но и создание экономических стимулов для внедрения систем верификации, например, через механизмы налоговых льгот или снижения страховых взносов для организаций, сертифицировавших свои информационные системы на соответствие стандартам C2PA [18].

10. ЗАКЛЮЧЕНИЕ

Проведенный анализ позволяет утверждать, что deepfake – это не просто новый вид мошенничества, а фактор, трансформирующий всю парадигму кибербезопасности [15]. Доверие к аудиовизуальной информации, считавшееся долгое время надежной основой коммуникации, в условиях распространения deepfake требует пересмотра. Традиционный принцип «доверяй, но проверяй» уступает место модели «никогда не доверяй, всегда проверяй» (Zero Trust), что уже зафиксировано в стандартах NIST SP 800-207 [18, С. 7]. Это означает, что любое взаимодействие, даже если голос и лицо собеседника идентифицируются как принадлежащие известному субъекту, должно проходить процедуру верификации.

Наши рекомендации – от многослойных детекторов до законодательной маркировки – направлены на создание системы, которая позволит если не остановить, то хотя бы сдержать волну deepfake-атак. Но главное, что необходимо обществу, – это формирование новой цифровой гигиены, предполагающей критическое отношение к любому аудиовизуальному контенту, особенно в контексте финансовых или конфиденциальных операций.

Практическая значимость работы заключается в том, что предложенные меры могут быть использованы при разработке корпоративных стандартов безопасности, государственных программ и образовательных курсов[1]. В условиях, когда различия между подлинным и синтезированным контентом все труднее уловить, адаптация возможна лишь за счет комплексного подхода: сочетания технологий, правовых инструментов и бдительности человека.

Перспективы дальнейших исследований связаны с разработкой проактивных методов защиты, основанных на принципах объяснимого искусственного интеллекта (XAI) и верификации происхождения контента в реальном масштабе времени. Особого внимания заслуживает создание адаптивных ансамблевых детекторов, способных к самообучению в условиях непрерывной эволюции генеративных моделей, а также формирование международных стандартов цифровой маркировки, обеспечивающих юридическую значимость процедур атрибуции подделок.

[1] Распоряжение Правительства РФ от 10.10.2019 № 2446-р «Об утверждении Национальной стратегии развития искусственного интеллекта на период до 2030 года» // Собрание законодательства РФ. 2019. № 42. Ст. 6298.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Goodfellow I. J.**, **Pouget-Abadie J.**, **Mirza M. et al.** Generative Adversarial Nets // Advances in Neural Information Processing Systems 27: Conference Proceedings. 2014. Vol. 27. P. 2672–2680. DOI: 10.48550/arXiv.1406.2661.
2. **Chesney R.**, **Citron D.** Deep Fakes: A Looming Challenge for Privacy, Democracy, and National Security // California Law Review. 2019. Vol. 107. P. 1753–1819. DOI: 10.15779/Z38GQ6Q.
3. CSIRO Data61 & Sungkyunkwan University. Evaluation of Deepfake Detection Systems: Limitations and Challenges: Technical Report. URL: https://data61.csiro.au/en/News/News-releases/2025/March/Deepfake-detection-benchmarking (дата обращения: 16.02.2026).
4. Proofpoint. 2024 State of the Phish Report. URL: https://www.proofpoint.com/us/resources/threat-reports/state-of-phish (дата обращения: 16.02.2026).
5. **Kumar R.**, **Sharma A.** Deepfake Social Engineering Attacks: Taxonomy, Security Risks and Defense Mechanisms // Computers & Security. 2024. Vol. 145. № 103312. DOI: 10.1016/j.cose.2024.103312.
6. **Адамский А. С.** Кибербезопасность в цифровой экономике: вызовы и решения // Вопросы государственного и муниципального управления. 2023. № 4. С. 156–178. DOI: 10.21293/2071-2590.2023.4.156-178.
7. **Грачёв А. Ю.**, **Соколов И. В.** Социально-инженерные атаки эпохи искусственного интеллекта // Проблемы информационной безопасности. Компьютерные системы. 2024. № 4. С. 33–49. DOI: 10.25610/2618-9516.2024.4.33.
8. World Economic Forum. Global Risks Report 2025. URL: https://www.weforum.org/reports/global-risks-report-2025 (дата обращения: 16.02.2026).
9. **Wang S.**, **Zhang X.**, **Xu Z. et al.** Deepfake Detection: A Survey and Evaluation of Current Methods // IEEE Access. 2023. Vol. 11. P. 57379–57399. DOI: 10.1109/ACCESS.2023.3283241.
10. Sensity AI (now Sentinel). The State of Deepfakes 2024. London: Sentinel Ltd, 2024. 52 p.
11. ENISA. Threat Landscape 2025. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025 (дата обращения: 16.02.2026).
12. Statista. Digital Market Outlook: Cybersecurity 2024. URL: https://www.statista.com/outlook/dmo/cybersecurity (дата обращения: 16.02.2026).
13. В России зафиксирован исторический максимум распространения дипфейков // Gazeta.ru. URL: https://www.gazeta.ru/social/news/2025/10/06/26892920.shtml (дата обращения: 16.02.2026).
14. **Борисов С.** В России в сентябре зафиксирован исторический максимум распространения дипфейков // МК в Костроме. URL: https://kostroma.mk.ru/social/2025/10/08/v-rossii-v-sentyabre-zafiksirovan-istoricheskiy-maksimum-rasprostraneniya-dipfeykov.html (дата обращения: 16.02.2026).
15. RAND Corporation. Synthetic Media and Hybrid Warfare: Systemic Threats to Democratic Institutions and Critical Infrastructure. URL: https://www.rand.org/pubs/research_reports/RRA2456-1.html (дата обращения: 16.02.2026).
16. Ensemble Methods for Deepfake Detection: A Comparative Study // Journal of Cybersecurity and Privacy. 2024. Vol. 4. № 2. P. 178–196. DOI: 10.3390/jcp4020012.
17. Microsoft. Digital Defense Report 2024. URL: https://www.microsoft.com/en-us/security/business/security-intelligence-report (дата обращения: 16.02.2026).
18. National Institute of Standards and Technology. NIST Special Publication 800–207. Zero Trust Architecture. URL: https://csrc.nist.gov/publications/detail/sp/800-207/final (дата обращения: 16.02.2026).

19. European Commission. Code of Practice on marking and labelling of AI-generated content // Digital Strategy. URL: https://digital-strategy.ec.europa.eu/en/policies/code-practice-ai-generated-content (дата обращения: 16.02.2026).
20. Federal Communications Commission. In the Matter of Steve Kramer: Forfeiture Order. FCC 24-104. URL: https://www.fcc.gov/document/fcc-issues-6m-fine-nh-robocalls (дата обращения: 16.02.2026).
21. State Internet Information Office. Artificial intelligence-generated Synthetic content Identification Method (National Credit Office Tongzi [2025] No. 2). URL: https://gxt.hebei.gov.cn/main/policy/zxzcdetail?id=a9ba0dd3-56e8-420f-95af-8f5b272f4788 (дата обращения: 16.02.2026).
22. В Государственную Думу внесен законопроект о маркировке видео, созданных с помощью ИИ // Парламентская газета. URL: https://sozd.duma.gov.ru/bill/1069302-8 (дата обращения: 16.02.2026).

REFERENCES

1. **Goodfellow I. J.**, **Pouget-Abadie J.**, **Mirza M. et al.** Generative Adversarial Nets. Advances in Neural Information Processing Systems 27: Conference Proceedings. 2014. Vol. 27, pp. 2672–2680. DOI: 10.48550/arXiv.1406.2661.
2. **Chesney R.**, **Citron D.** Deep Fakes: A Looming Challenge for Privacy, Democracy, and National Security. *California Law Review*. 2019. Vol. 107, pp. 1753–1819. DOI: 10.15779/Z38GQ6Q.
3. CSIRO Data61 & Sungkyunkwan University. Evaluation of Deepfake Detection Systems: Limitations and Challenges: Technical Report. URL: https://data61.csiro.au/en/News/News-releases/2025/March/Deepfake-detection-benchmarking (accessed: 16.02.2026).
4. Proofpoint. 2024 State of the Phish Report. URL: https://www.proofpoint.com/us/resources/threat-reports/state-of-phish (accessed: 16.02.2026).
5. **Kumar R.**, **Sharma A.** Deepfake Social Engineering Attacks: Taxonomy, Security Risks and Defense Mechanisms. Computers & Security. 2024. Vol. 145. No. 103312. DOI: 10.1016/j.cose.2024.103312.
6. **Adamskiy A. S.** Cybersecurity in the Digital Economy: Challenges and Solutions. *Voprosy gosudarstvennogo i munitsipal'nogo upravleniya*. 2023. No. 4, pp. 156–178. DOI: 10.21293/2071-2590.2023.4.156-178. (In Russian)
7. **Grachev A. Yu.**, **Sokolov I. V.** Social Engineering Attacks in the Era of Artificial Intelligence. *Problemy informatsionnoy bezopasnosti. Komp'yuternye sistemy*. 2024. No. 4, pp. 33–49. DOI: 10.25610/2618-9516.2024.4.33. (In Russian)
8. World Economic Forum. Global Risks Report 2025. URL: https://www.weforum.org/reports/global-risks-report-2025 (accessed: 16.02.2026).
9. **Wang S.**, **Zhang X.**, **Xu Z. et al.** Deepfake Detection: A Survey and Evaluation of Current Methods. *IEEE Access*. 2023. Vol. 11, pp. 57379–57399. DOI: 10.1109/ACCESS.2023.3283241.
10. Sensity AI (now Sentinel). The State of Deepfakes 2024. London: Sentinel Ltd, 2024, 52 p.
11. ENISA. Threat Landscape 2025. URL: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025 (accessed: 16.02.2026).
12. Statista. Digital Market Outlook: Cybersecurity 2024. URL: https://www.statista.com/outlook/dmo/cybersecurity (accessed: 16.02.2026).
13. Historical maximum of deepfake distribution recorded in Russia. *Gazeta.ru*. URL: https://www.gazeta.ru/social/news/2025/10/06/26892920.shtml (accessed: 16.02.2026). (In Russian)
14. **Borisov S.** In September, a historical maximum of deepfake distribution was recorded in Russia. *MK v Kostrome*. URL: https://kostroma.mk.ru/social/2025/10/08/v-rossii-v-sentyabre-zafiksirovan-istoricheskiy-maksimum-rasprostraneniya-dipfeykov.html (accessed: 16.02.2026). (In Russian)
15. RAND Corporation. Synthetic Media and Hybrid Warfare: Systemic Threats to Democratic Institutions and Critical Infrastructure. URL: https://www.rand.org/pubs/research_reports/RRA2456–1.html (accessed: 16.02.2026).
16. Ensemble Methods for Deepfake Detection: A Comparative Study. Journal of Cybersecurity and Privacy. 2024. Vol. 4. No. 2, pp. 178–196. DOI: 10.3390/jcp4020012.
17. Microsoft. Digital Defense Report 2024. URL: https://www.microsoft.com/en-us/security/

business/security-intelligence-report (accessed: 16.02.2026).
18. National Institute of Standards and Technology. NIST Special Publication 800–207. Zero Trust Architecture. URL: https://csrc.nist.gov/publications/detail/sp/800-207/final (accessed: 16.02.2026).
19. European Commission. Code of Practice on marking and labelling of AI-generated content. Digital Strategy. URL: https://digital-strategy.ec.europa.eu/en/policies/code-practice-ai-generated-content (accessed: 16.02.2026).
20. Federal Communications Commission. In the Matter of Steve Kramer: Forfeiture Order. FCC 24-104. URL: https://www.fcc.gov/document/fcc-issues-6m-fine-nh-robocalls (accessed: 16.02.2026).
21. State Internet Information Office. Artificial intelligence-generated Synthetic content Identification Method (National Credit Office Tongzi [2025] No. 2). URL: https://gxt.hebei.gov.cn/main/policy/zxzcdetail?id=a9ba0dd3-56e8-420f-95af-8f5b272f4788 (accessed: 16.02.2026).
22. A bill on marking videos created using AI has been submitted to the State Duma. *Parlamentskaya gazeta*. URL: https://sozd.duma.gov.ru/bill/1069302–8 (accessed: 16.02.2026). (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

КИСЕЛЁВ Алексей Николаевич – канд. техн. наук, доцент, Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, Ждановская ул., д. 13
E-mail: kan534@mail.ru
ORCID: 0009-0000-8038-0717

KISELEV Alexey N. – Candidate of Engineering Sciences, Associate Professor, Mozhaisky Military Space Academy, Russia, 197198, St. Petersburg, Zhdanovskaya str., 13

БОНДАРЕНКО Василий Сергеевич – слушатель, Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, Ждановская ул., д. 13
E-mail: vasja13012004@gmail.ru
ORCID: 0009-0000-9892-249X

BONDARENKO Vasily S. – Listener, Mozhaisky Military Space Academy, Russia, 197198, St. Petersburg, Zhdanovskaya str., 13

ТАТАРЕНКО Даниил Григорьевич – слушатель, Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, Ждановская ул., д. 13
E-mail: daniil_tatarenk@mail.ru
ORCID: 0009-0002-2999-9631

TATARENKO Daniil G. – Listener, Mozhaisky Military Space Academy, Russia, 197198, St. Petersburg, Zhdanovskaya str., 13

Научная статья
DOI 10.66424/2071-8217-2026-2-2
УДК 004.056

МЕТОД ВЫБОРА ТЕХНИЧЕСКОЙ РЕАЛИЗАЦИИ МЕР РЕАГИРОВАНИЯ НА ИНЦИДЕНТЫ

А. В. Кузнецов[1,2*]

[1]ООО «РТК ИБ», Москва, Россия

[2]Финансовый университет при Правительстве Российской Федерации, Москва, Россия

✉ *1283_my@mail.ru

ДЛЯ ЦИТИРОВАНИЯ

Кузнецов А. В. Метод выбора технической реализации мер реагирования на инциденты // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 22–31. DOI: 10.66424/2071-8217-2026-2-2

ПОСТУПИЛА 16.02.2026
ПРИНЯТА 27.04.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Принимая во внимание возрастающее значение своевременности реагирования на инциденты информационной безопасности предложен метод выбора технической реализации мер реагирования на инциденты информационной безопасности без участия группы реагирования. Метод принимает во внимание заданные ограничения на предоставленные манданты и покрытие средствами реагирования. В рамках метода, в отличие от известных, рассматривается задача выбора как задача целочисленного (булевого) линейного программирования, в которой члены целевой функции являются логическими переменными, учитывающими логические действия по локализации инцидентов информационной безопасности, предусмотренные планами реагирования. Применение метода позволяет минимизировать время, затрачиваемое на локализацию инцидентов информационной безопасности.

КЛЮЧЕВЫЕ СЛОВА

Средство реагирования, группа реагирования, локализация (сдерживание) инцидента, автоматическое реагирование, мандат на действие, план реагирования

Original article
DOI 10.66424/2071-8217-2026-2-2

THE METHOD FOR SELECTING TECHNICAL IMPLEMENTATION OF INCIDENT RESPONSE MEASURES

A. V. Kuznetsov[1,2*]

[1]RTK IB LLC, Moscow, Russia

[2]Financial University under the Government of the Russian Federation, Moscow, Russia

✉ *1283_my@mail.ru

FOR CITATION

Kuznetsov A. V. The method for selecting technical implementation of incident response measures. *Problems of information security. Computer systems*. 2026. No. 2, pp. 22–31. DOI: 10.66424/2071-8217-2026-2-2 (In Russian)

ABSTRACT

Considering the increasing importance of timely response to information security incidents, the method for selecting technical implementation of information security incident response measures without the involvement of a response team is proposed. The method considers specified constraints on provided mandates and the coverage of response tools. Unlike known methods, this method considers the selection problem as an integer (boolean) linear programming problem. The terms of the objective function are logical variables for the infor-

RECEIVED 16.02.2026
ACCEPTED 27.04.2026
PUBLICATION 15.06.2026

mation security incident localization that included into response plans. Thereby minimizing the time spent for information security incident localization.

KEYWORDS

Response tool, response team, incident (containment) localization, automated response, action mandate, response plan

1. ВВЕДЕНИЕ

По итогам 2025 г. сразу несколько российских организаций, специализирующихся на оказании услуг по контролю и анализу защищенности информационных (компьютерных) систем, отметили, что современные атакующие (нарушители) все чаще преследуют цель полного уничтожения информационной инфраструктуры и накопленных в ней данных, в том числе резервных копий [1, 2].

В сложившейся ситуации обеспечение своевременности и корректности технической реализации мер реагирования на инциденты информационной безопасности (ИБ), возникающие вследствие компьютерных атак (кибератак), является актуальным направлением исследований. При этом стоит отметить, что для технической реализации мер реагирования на возникающие инциденты ИБ в организациях могут применяться различные средства реагирования. К таким средствам можно отнести как специализированные средства защиты информации (Endpoint Detection and Response, Network Detection and Response, Extended Detection and Response) [3], так и встроенные в общесистемное или прикладное программное обеспечение (ПО) механизмы управления и защиты информации (табл. 1). Например, сетевая изоляция хоста компьютерной сети может быть реализована

Таблица 1 | Сравнение средств реагирования

Table 1 | Comparison of response tools

Наименование средства	Область применения	Основные возможности по локализации
Endpoint Detection and Response	Серверы и рабочие станции (общесистемное ПО)	•Остановка служб, процессов; • изолирование «зараженных» объектов; • сетевая изоляция
Network Detection and Response	Каналы связи	• Запрет (блокирование, ограничение) прохождения сетевого трафика; • сетевая изоляция
Extended Detection and Response	Серверы и рабочие станции (общесистемное ПО), каналы связи	• Остановка служб, процессов; • изолирование «зараженных» объектов; • сетевая изоляция; • запрет (блокирование, ограничение) прохождения сетевого трафика
Межсетевые экраны уровня сети (тип «А»), веб-сервера (тип «Г»), узла (тип «В»)	Периметр сети, веб-сайты, серверы и рабочие станции	Запрет (блокирование, ограничение) прохождения сетевого трафика
Средства антивирусной защиты (типы «Б», «В», «Г»)	Серверы и рабочие станции	• Удаление вредоносного ПО; • изолирование «зараженных» объектов
Встроенные механизмы управления доступом общесистемного или прикладного ПО	Общесистемное ПО, прикладное ПО	• Блокировка учетных записей; • ограничение прав учетных записей; • остановка служб, процессов

различными средствами реагирования: встроенными или наложенными персональными межсетевыми экранами, активным сетевым оборудованием или сетевыми средствами защиты информации. Другой пример – блокировка скомпрометированной учетной записи, которая может быть реализована различными техническими способами: временная блокировка или постоянное отключение в централизованной службе каталогов (Directory Service, DS) или в системе класса Identity and Access Management (IdM), перенос в специальную группу или назначение специальной роли в DS, IdM и/или в прикладном ПО, дополнительно могут быть сброшены открытые сетевые сессии, а также изменен пароль или другой аутентификатор. Поддерживать различные технические варианты реагирования затратно, требуется время и профильная экспертиза, также обычно каждый коннектор к средству реагирования лицензируется отдельно [4], т.е. целесообразно минимизировать их количество.

На практике для выбора того или иного технического действия даже при наличии плана (сценария) реагирования применяются экспертные методы и системы [5, 6], напрямую зависящие от квалификации привлекаемых членов (экспертов) групп реагирования на инциденты ИБ. Применение экспертных методов усугубляется дефицитом кадров в области обеспечения ИБ [7] и не всегда позволяет обеспечивать воспроизводимость принимаемых решений, особенно при высокой вариативности технических действий. Это не позволяет перевести реализацию мер реагирования на возникающие инциденты ИБ в автоматический режим, т.е. без участия сил групп реагирования на инциденты ИБ [8]. Особенно актуальна данная ситуация в масштабах крупных территориально распределенных гетерогенных информационных инфраструктур, где квалификация и доступность членов групп реагирования на инциденты ИБ может разниться от площадки к площадке. Таким образом, задача оптимального выбора технического действия в рамках реагирования является актуальной и требует решения путем формирования метода выбора технической реализации мер реагирования на возникающие инциденты ИБ, направленного на минимизацию участия сил групп реагирования на инциденты ИБ и сокращение времени реагирования.

В рамках исследования будут приняты следующие ограничения:

• не рассматривается задача формирования планов (сценариев) реагирования, они выступают исходными данными, содержащими логические действия, такие как изоляция хоста компьютерной сети, блокировка скомпрометированной учетной записи, завершение работы приложения и т.п. (требования к содержанию планов реагирования определены ГОСТ Р 59711-2022 «Защита информации. Управление компьютерными инцидентами. Организация деятельности по управлению компьютерными инцидентами»);

• в части мер реагирования на инциденты ИБ рассматриваются только действия по локализации (сдерживанию) инцидентов ИБ, направленные на активное противодействие атакующему [9] (методы и средства расследования инцидентов ИБ не рассматриваются);

• локализация предусмотрена только для подтвержденных инцидентов ИБ, не требующих дополнительного расследования [10] (методы и средства подтверждения инцидентов ИБ не рассматриваются);

• исследование инвариантно к используемому терминологическому аппарату: «инцидент ИБ», «компьютерный инцидент», «инцидент защиты информации» и «киберинцидент».

Объектом исследования выступают процессы выбора технических мер реагирования на инциденты ИБ в рамках распределенной защищаемой информационной инфраструктуры, размещенной на P площадок, с использованием средств реагирования, реализующих R технических действий. Предметом исследования выступают методы выбора (оптимизации) технических мер локализации инцидентов ИБ.

Цель исследования – синтез метода выбора технической реализации мер реагирования на инциденты ИБ, позволяющего минимизировать время, затрачиваемое на непосредственную реализацию технических мероприятий по локализации инцидентов ИБ (T_{Me}), за счет формализации процесса выбора в терминах теории исследования операций.

2. МЕТОДЫ

Обозначим $\overline{X_1}$ вектором переменных все технические действия по локализации инцидентов ИБ, которые потенциально можно выполнить группе реагирования на инциденты ИБ, а $\overline{X_2}$ – вектором технических действий, выбранных группой реагирования на инциденты ИБ:

$$x_r = \begin{cases} 1, \text{техническое действие выбрано} \\ 0, \text{техническое действие не выбрано} \end{cases}, (1)$$

под методом выбора технической реализации мер реагирования на инциденты ИБ будем понимать отображение $M_{loc}: \overline{X_1} \to \overline{X_2}$. Разрабатываемый метод должен обладать свойством масштабируемости и учитывать гетерогенность используемых средств реагирования.

Сравнение возможностей применения наиболее популярных средств реагирования для реализации технических действий по локализации инцидентов ИБ приведено в табл. 1.

Важно задать ограничения на выбор тех или иных технических действий с учетом возможности реализации данных действий в автоматическом режиме, т.е. без непосредственного участия членов групп реагирования на инциденты ИБ.

Первоначальным ограничением выступает минимизация количества выбираемых технических действий, что позволяет поддерживать меньшее количество технических интеграций (сокращает стоимость лицензий), минимизирует нагрузку на силы групп реагирования на инциденты ИБ и сами средства реагирования:

$$\sum_{r=1}^{R} x_r \to \min. \quad (2)$$

Одним из первичных ограничений выступает наличие покрытия средствами реагирования защищаемой информационной инфраструктуры (размещения компонентов средств реагирования), так как если нет покрытия, то нет технической возможности реализовать меру по локализации инцидента ИБ ни в автоматическом, ни в ручном режиме.

Указанные ограничения задаются матрицей $S = \|s_{pr}\|_{P,R}$ (табл. 2), где $s_{pr} \in \{0;1\}$ – булевая (логическая) переменная, отражающая наличие покрытия средствами реагирования для выполнения r технического действия на p-й площадке информационной инфраструктуры, $p = \overline{1,P}$, $r = \overline{1,R}$.

Еще одним из первичных ограничений выступает наличие мандата (отсутствие мандата) на автоматическое выполнение технического действия, так как если его нет, то требуется вовлечение для реализации членов группы реагирования на инциденты ИБ в ручном режиме.

Мандат представляет собой цифровую запись, например, в системе классов Incident Response Platform или Security Orchestration, Automation and Response [11, 12], содержащую идентификационную информацию, характеризующую условия локализации инцидента ИБ, с учетом заданных критериев, характерных для конкретной организации и ее информационной инфраструктуры (например: по территории, информационном системам, периоду времени и т.п.) [13].

Указанные ограничения задаются матрицей $M = \|m_{pr}\|_{P,R}$ (табл. 3), где $m_{pr} \in \{0;1\}$ – булевая (логическая) переменная, отражающая предоставление мандата на r техническое действие на p-й площадке

Таблица 2 | Фрагмент матрицы *S* (пример)

Table 2 | Fragment of matrix *S* (example)

S		***r***				
		1	*2*	…	*R–1*	*R*
p	*1*	0	1	…	1	1
	…	0	1	…	0	1
	P	1	0	…	1	1

информационной инфраструктуры, $p=\overline{1,P}$, $r=\overline{1,R}$.

Формируется $P\times R$ систем условий, отражающих техническую возможность выполнить *r* техническое действие на *p*-й площадке информационной инфраструктуры без участия сил группы реагирования на инциденты ИБ:

$$\begin{cases} s_{pr}x_r=1 \\ m_{pr}x_r=1 \end{cases} \text{ для } p=const. \qquad (3)$$

Для совокупности используемых в организации документированных планов реагирования формируется матрица $D=\|d_{kl}\|_{K,L}$, отражающая вхождение в них логических действий по локализации инцидентов ИБ (табл. 4), где $d_{kl}\in[0;1]$ – переменная, отражающая вклад *l* логического действия в *k*-й план реагирования, с соблюдением условия:

$$\sum_{l=1}^{L} d_{kl}=1 \text{ для } k=const. \qquad (4)$$

Для сопоставления логических и технических действий по локализации инцидентов ИБ сформирована булевая матрица $A=\|a_{rl}\|_{R,L}$ (табл. 5), где $a_{rl}\in\{0;1\}$ – булевая (логическая) переменная, отражающая возможность реализации *l* логического действия *r*-м техническим действием, т. е. средством реагирования.

Формируется *L* условий, отражающих возможность реализации *l* логического действия как минимум одним техническим действием:

Таблица 3 | Фрагмент матрицы *M* (пример)

Table 3 | Fragment of matrix *M* (example)

M		***r***				
		1	*2*	…	*R–1*	*R*
p	*1*	1	0	…	1	0
	…	0	1	…	0	1
	P	1	1	…	0	1

Таблица 4 | Фрагмент матрицы *D* (пример)

Table 4 | Fragment of matrix *D* (example)

D		***l***				
		1	*2*	…	*L–1*	*L*
k	*1*	0,5	0	…	0	0,5
	…	0,25	0,25	…	0,25	0,25
	K	1	0	…	0	0

Таблица 5 | Фрагмент матрицы *A* (пример)

Table 5 | Fragment of matrix *A* (example)

A		***l***				
		1	*2*	…	*L–1*	*L*
r	*1*	0	0	…	1	1
	…	1	1	…	0	0
	R	1	1	…	0	1

$$y_l(x) = \sum_{r=1}^{R} a_{rl} x_r \geq 1 \text{ для } l = \text{const}. \quad (5)$$

С учетом условия (5) формируется $\overline{Y}$ – вектор возможности выполнения логических действий, зависящий от $\overline{X_2}$:

$$y_l(x) = \begin{cases} 1, \text{ логическое действие реализовать возможно} \\ 0, \text{ логическое действие реализовать невозможно} \end{cases}. \quad (6)$$

Вводится член целевой функции I_k, отражающий автоматическое выполнение k-го плана реагирования, содержащего набор логических действий y_l, реализация которых возможна выбранными техническими действиями x_r (т.е. характеризующей автоматическую локализацию инцидента ИБ):

$$I_k = \sum_{l=1}^{L} d_{kl} y_l(x), \quad (7)$$

где

$$I_k = \begin{cases} I_k = 1, \text{ автоматическая реализация плана реагирования;} \\ I_k \neq 1, \text{ неавтоматическая реализация плана реагирования, тогда } I_k = 0 \end{cases}. \quad (8)$$

Сокращение времени, затрачиваемого на непосредственную реализацию технических мероприятий по локализации инцидентов ИБ в информационной инфраструктуре T_{ME}, и минимизация участия сил групп реагирования на инциденты ИБ в этом предусматривает увеличение количества планов реагирования, выполняемых в автоматическом режиме, т. е. целевая функция F принимает вид:

$$F = \sum_{k=1}^{K} I_k \to \max \Rightarrow T_{Me} \to \min. \quad (9)$$

Целевая функция F является линейной функцией. Таким образом, поставленная задача сводится к задаче поиска экстремума на множествах, заданных системами линейных равенств и неравенств (3), (5) с целочисленными (булевыми) переменными x_r, т. е. является задачей целочисленного (булевого) линейного программирования [14, 15]. Таким образом, в такой постановке задачи возникает интерпретация теории исследования операций применительно к принятию решений в рамках реагирования на инциденты ИБ.

В качестве математического метода решения выбран метод ветвей и границ [15, 16], который модифицирован в части учета того, что члены целевой функции являются логическими переменными (8), определяемыми планами реагирования и возможностью их реализации техническим действиями с учетом заданных ограничений на мандаты и покрытие средствами реагирования.

Предложенный метод M_{loc} (последовательность действий, приводящая к выбору оптимального набора технических действий) включает в себя следующие шаги и применяется последовательно:

1. Формирование матрицы S ограничений на покрытие средствами реагирования с учетом реального размещения компонентов средств реагирования (агентов и/или шлюзов).

2. Формирование матрицы M ограничений на предоставленные манданты согласно политике управления доступом в информационной инфраструктуре.

3. Формирование матрицы D ограничений на состав планов реагирования в части логических действий согласно документированным планам реагирования.

4. Формирование матрицы A ограничений на реализацию логических действий техническими действиями с учетом технических возможностей применяемых средств реагирования.

5. Формирование условий (3) и (5).

6. Поиск решения с применением модифицированного метода ветвей и границ.

По результатам применения метода M_{loc} будут установлены: минимальное количество и состав необходимых (оптимальных) технических действий $\overline{X_2}$; количество автоматически выполняемых планов реагирования F.

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Предложенный метод применен на следующем контрольном примере, в котором матрицы S, M и A сформированы с использованием генераторов псевдослучайных чисел, а в матрице D предусмотрено по четыре логических действия для каждого плана реагирования ($d_{kl} = 0{,}25$), остальные параметры задачи принимали следующие значения: $R = 4$; $P = 4$; $L = 8$; $K = 4$, чтобы в качестве проверки провести исчерпывающий перебор. Результаты применения метода и проверки совпали (табл. 6).

Таким образом, существует оптимальный набор технических действий, выбранных группой реагирования на инциденты ИБ, позволяющий обеспечить реализацию максимального количества планов реагирования в автоматическом режиме.

Принимая во внимание, что согласно исследованию компании Logshero среднее время реагирования на инциденты ИБ занимает у 82 % организаций более одного часа [17], а реализация данного метода и выбранных в рамках него технических действий по локализации – менее одной минуты (шестой шаг), то время, затрачиваемое на непосредственную реализацию технических мероприятий по локализации инцидентов ИБ, сокращается в 60 раз.

4. ЗАКЛЮЧЕНИЕ

По результатам проведенного исследования предложен метод выбора технической реализации мер реагирования на инциденты ИБ, принимающий во внимание заданные ограничения на предоставленные манданты и покрытие средствами реагирования, в рамках которого задача выбора рассмотрена как задача целочисленного (булевого) линейного программирования, в которой члены целевой функции являются логическими переменными, учитывающими логические действия по локализации инцидентов ИБ, предусмотренные планами реагирования. Применение данного метода позволяет максимизировать количество планов реагирования, выполняемых в автоматическом режиме (без участия сил реагирования на инциденты ИБ), тем самым минимизировать время, затрачиваемое на локализацию инцидентов ИБ.

Применение результатов исследования дает положительный эффект в области технических наук (методы и системы защиты информации, информационная безопасность) и наиболее значимо для владельцев (операторов) распределенных информационных (компьютерных) систем и входящих в их состав групп реагирования на инциденты ИБ.

Предложенный метод проходит практическую апробацию на базе крупнейшего в России коммерческого центра мониторинга и реагирования на кибератаки ООО «РТК ИБ».

Таблица 6 | Значения для контрольного примера

Table 6 | Values for the test case

Вар-т	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	***10***	*11*	*12*	*13*	*14*	*15*	*16*
x_1	1	0	1	1	1	1	0	0	1	**1**	0	1	0	0	0	0
x_2	1	1	0	1	1	1	0	1	0	**0**	1	0	1	0	0	0
x_3	1	1	1	0	1	0	1	1	0	**1**	0	0	0	1	0	0
x_4	1	1	1	1	0	0	1	0	1	**0**	1	0	0	0	1	0
F	4	0	4	4	4	2	0	0	0	**4**	0	0	0	0	0	0

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. От выявления компрометации до реагирования: как российские компании справлялись с кибератаками в 2025 году. URL: https://bi.zone/expertise/insights/ot-otsenki-komprometatsii-do-reagirovaniya-kak-rossiyskie-kompanii-spravlyalis-s-kiberatakami-v-2025/ (дата обращения: 12.02.2026).
2. Курс на антихрупкость стратегический обзор киберугроз 2025. URL: https://jetcsirt.su/analytics/kurs-na-antikhrupkost-strategicheskiy-obzor-kiberugroz-2025/ (дата обращения: 12.02.2026).
3. **Метельков А. Н.** Многоликость мониторинга в обеспечении информационной безопасности // Правовая информатика. 2025. № 4. С. 69–78.
4. Security Vision 5: эволюция автоматизации. URL: https://safe.cnews.ru/articles/2021-11-01_security_vision_5_evolyutsiya_avtomatizatsii (дата обращения: 03.04.2026).
5. **Голицын С. А., Шульженко А. Д.** Концептуальный подход к построению центра мониторинга системы обнаружения, предупреждения и ликвидации последствий компьютерных атак // Globus: Технические науки. 2021. Т. 7. № 1 (37). С. 40–43.
6. **Микрюков А. А., Куулар А. В.** Совершенствование процесса управления инцидентами на основе прецедентного подхода // Открытое образование. 2021. Т. 25. № 4. С. 47–54.
7. 41 % компаний испытывают нехватку специалистов в области информационной безопасности. URL: https://www.kaspersky.ru/about/press-releases/globalnoe-issledovanie-laboratorii-kasperskogo-41-kompanij-ispytyvayut-nehvatku-specialistov-v-oblasti-informacionnoj-bezopasnosti (дата обращения: 12.02.2026).
8. **Кузнецов А. В.** Эволюция реагирования на инциденты информационной безопасности // Защита информации. Инсайд. 2024. № 5 (119). С. 14–20.
9. **Милославская Н. Г., Толстой А. И.** Управление инцидентами информационной безопасности. М. : Горячая Линия – Телеком, 2024. С. 105–109; 169–193.
10. **Чижевский М. А., Серпенинов О. В., Лапсарь А. П.** Оптимизация алгоритма расследования компьютерных инцидентов в SIEM-системах // Проблемы информационной безопасности. Компьютерные системы. 2025. № 3 (66). С. 69–80. DOI: 10.48612/jisp/rmzt-68hn-ung8.
11. **Мухитов А. А., Шафиков М. Р.** Проблематика автоматизации процесса реагирования на инциденты ИБ // Advances in Science and Technology : сборник статей LX международной научно-практической конференции, 30 апреля 2024 г., Москва, Россия. М. : ООО АКТУАЛЬНОСТЬ.РФ, 2024. С. 90–93.
12. **Власова А. В., Дударев В. А., Новикова Т. И.** Обзор основных направлений и технологий применения систем SOC в системе информационной безопасности // Фундаментальные и прикладные научные исследования: инноватика в современном мире: сборник научных статей по материалам IX Международной научно-практической конференции, 25 ноября 2022 г., Уфа, Россия. Уфа : ООО «Научно-издательский центр «Вестник науки»», 2022. С. 235–239.
13. **Кузнецов А. В.** Анализ критериев предоставления мандата на локализацию инцидента информационной безопасности // Инженерный вестник Дона. 2025. № 3 (123). С. 217–226.
14. **Трушков А. С.** Задача целочисленного программирования с булевыми переменными // Актуальные вопросы современной информатики : материалы IX Всероссийской научно-практической конференции, 1–15 апреля 2019 г., Коломна, Россия. Коломна : Государственный социально-гуманитарный университет, 2019. С. 87–95.

15. **Дейкина А. С., Червякова М. В.** Метод ветвей и границ для решения задачи линейного программирования с булевыми переменными // Материалы секционных заседаний 57-й студенческой научно-практической конференции ТОГУ, 17–27 апреля 2017 г., Хабаровск, Россия. В 2 т. Хабаровск : Тихоокеанский государственный университет, 2017. С. 16–21.
16. Применение Метода ветвей и границ для решения Экстремальных задач. URL: https://naukamirowozreniya.ru/public/202511/application/1764226987579351894/primenenie-metoda-vetvej-i-granic-dlya-resheniya-ekstremalnyh-zadach.pdf (дата обращения: 12.02.2026).
17. 2024 Observability Pulse Report. URL: https://logz.io/observability-pulse-2024/?utm_medium=referral&utm_source=cncf#executive-summary (дата обращения: 12.02.2026).

REFERENCES

1. From compromise detection to response: How Russian companies coped with cyberattacks in 2025. URL: https://bi.zone/expertise/insights/ot-otsenki-komprometatsii-do-reagirovaniya-kak-rossiyskie-kompanii-spravlyalis-s-kiberatakami-v-2025/ (accessed: 12.02.2026). (In Russian)
2. Antifragility: A Strategic Cyber Threat Review 2025. URL: https://jetcsirt.su/analytics/kursna-antikhrupkost-strategicheskiy-obzor-kiberugroz-2025/ (accessed: 12.02.2026). (In Russian)
3. **Metelkov A. N.** Diversity of monitoring in ensuring information security. *Legal Informatics*. 2025. No. 4, pp. 69–78. (In Russian)
4. Security Vision 5: evolution of automation. URL: https://safe.cnews.ru/articles/2021-11-01_security_vision_5_evolyutsiya_avtomatizatsii (accessed: 03.04.2026). (In Russian)
5. **Golitsyn S. A., Shulzhenko A. D.** Conceptual approach to the construction of a monitoring center for detection, prevention and elimination of the consequences of computer attacks. *Globus: Technical Sciences*. 2021. Vol. 7. No. 1 (37), pp. 40–43. (In Russian)
6. **Mikryukov A. A., Kuular A. V.** Improving the incident management process based on a use case approach. *Open Education*. 2021. Vol. 25. No. 4, pp. 47–54. (In Russian)
7. 41 % of companies are experiencing a shortage of information security specialists. URL: https://www.kaspersky.ru/about/press-releases/globalnoe-issledovanie-laboratorii-kasperskogo-41-kompanij-ispytyvayut-nehvatku-specialistov-v-oblasti-informacionnoj-bezopasnosti (accessed: 12.02.2026). (In Russian).
8. **Kuznetsov A. V.** The evolution of information security incident response. *Zaŝita informacii. Inside*. 2024. No. 5 (119), pp. 14–20. (In Russian)
9. **Miloslavskaya N. G., Tolstoy A. I.** Information Security Incident Management. Moscow : Hot Line – Telecom, 2024, pp. 105–109; 169–193. (In Russian)
10. **Chizhevsky M. A., Serpeninov O. V., Lapsar A. P.** Optimization of computer incident investigation algorithm in siem systems. *Problems of information security. Computer systems.* 2025. No. 3 (66), pp. 69–80. DOI: 10.48612/jisp/rmzt-68hn-ung8. (In Russian)
11. **Mukhitov A. A., Shafikov M. R.** Problems of automation of the process of response to information security incidents. Advances in Science and Technology. Collection of articles of the LX international scientific and practical conference, 30 April 2024, Moscow, Russia. Moscow: OOO AKTUAL'NOST'.RF, 2024, pp. 90–93. (In Russian)
12. **Vlasova A. V., Dudarev V. A., Novikova T. I.** Review of the main directions and technologies for the application of SOC systems in the information security system. Fundamental and applied scientific research: innovation in the modern world. Collection of scientific articles based on the materials of the IX International scientific and practical conference, 25 November 2022, Ufa, Russia. Ufa : OOO "Nauchno-izdatel'skij centr "Vestnik nauki", 2022, pp. 235–239. (In Russian)
13. **Kuznetsov A. V.** The analysis of criteria for granting a mandate to an information security incident localization. *Engineering Journal of Don*. 2025. No. 3 (123), pp. 217–226. (In Russian)

14. **Trushkov A. S.** Integer programming problem with Boolean variables. Actual issues of modern informatics. Proceedings of the IX All-Russian scientific and practical conference, 1–15 April 2019, Kolomna, Russia. Kolomna, Gosudarstvennyj social'no-gumanitarnyj universitet, 2019, pp. 87–95. (In Russian)
15. **Deikina A. S., Chervyakova M. V.** Branch and bound method for solving a linear programming problem with Boolean variables. Materials of sectional meetings of the 57th student scientific and practical conference, 17–27 April 2017, Khabarovsk, Russia. In 2 volumes. Khabarovsk, Pacific National University, 2017, pp. 16–21. (In Russian)
16. Application of the Branch and Bound Method to Solve Extremal Problems. URL: https://naukamirowozreniya.ru/public/202511/application/1764226987579351894/primenenie-metoda-vetvej-i-granic-dlya-resheniya-ekstremalnyh-zadach.pdf (accessed: 12.02.2026). (In Russian)
17. 2024 Observability Pulse Report. URL: https://logz.io/observability-pulse-2024/?utm_medium=referral&utm_source=cncf#executive-summary (accessed: 12.02.2026).

СВЕДЕНИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHOR

КУЗНЕЦОВ Александр Васильевич – канд. техн. наук, руководитель отдела комплексных технических решений, ООО «РТК ИБ», Россия, 125009, Москва, пер. Никитский, д. 7, стр. 1; доцент, Финансовый университет при Правительстве Российской Федерации, Россия, 125167, Москва, пр-т Ленинградский, д. 49/2
E-mail: 1283_my@mail.ru
ORCID: 0000-0002-7160-1845

KUZNETSOV Aleksandr V. – Candidate of Engineering Sciences, Head of Integrated Technical Solutions Department, RTK IB LLC, Russia, 125009, Moscow, Nikitsky Lane, 7, build. 1; Associate Professor, Financial University under the Government of the Russian Federation, Russia, 125167, Moscow, Leningradsky ave., 49/2

Научная статья
DOI 10.66424/2071-8217-2026-2-3
УДК 004.056.5

МЕТОД ОПРЕДЕЛЕНИЯ ТИПИЧНЫХ ВРЕМЕННЫХ ХАРАКТЕРИСТИК СОБЫТИЙ БЕЗОПАСНОСТИ НА ОСНОВЕ СТАТИСТИЧЕСКИХ ДАННЫХ ДЛЯ ЗАДАЧ КОРРЕЛЯЦИОННОГО АНАЛИЗА

А. Д. Миханько*, И. В. Машкина

Уфимский университет науки и технологий, Уфа, Республика Башкортостан, Россия

✉ *mikhanko45@gmail.com

ДЛЯ ЦИТИРОВАНИЯ

Миханько А. Д., Машкина И. В. Метод определения типичных временных характеристик событий безопасности на основе статистических данных для задач корреляционного анализа // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 32–48. DOI: 10.66424/2071-8217-2026-2-3

ПОСТУПИЛА 10.04.2026
ПРИНЯТА 13.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Представлен метод определения типичных временных параметров событий информационной безопасности на основе анализа журналов событий. Метод ориентирован на обработку межсобытийных интервалов и позволяет выявлять характерные временные закономерности функционирования источников событий безопасности. Предложенный подход включает: формирование выборки временных интервалов, выделение структурного разрыва межсобытийных и межсессионных интервалов, фильтрацию выбросов с использованием межквартильного размаха, определение типичных значений на основе кластеризации и группового анализа. Для учета вариативности данных применяется оценка среднего значения и стандартного отклонения с последующим разбиением на интервальные окна. Проведен численный эксперимент по данным журналов реальных событий, подтверждающий работоспособность метода при анализе источников с различной интенсивностью генерации событий. Эксперимент проведен на журналах OPC-сервера, Windows Server, СУБД PostgreSQL. Полученные результаты демонстрируют устойчивость метода к выбросам, мультимодальности распределений и наличию нулевых интервалов. Разработанный метод может быть использован при построении правил корреляции в SIEM-системах, а также в задачах анализа поведения и выявления аномалий в инфраструктуре информационной безопасности.

КЛЮЧЕВЫЕ СЛОВА

Информационная безопасность, журналы событий, SIEM, временные интервалы, межсобытийные интервалы, анализ логов, обнаружение аномалий, межквартильный размах, кластеризация, статистический анализ, корреляция событий, поведенческий анализ

Original article
DOI 10.66424/2071-8217-2026-2-3

A METHOD FOR DETERMINING TYPICAL TIME CHARACTERISTICS OF SECURITY EVENTS BASED ON STATISTICAL DATA FOR CORRELATION ANALYSIS TASKS

A. D. Mikhanko*, I. V. Mashkina

Ufa University of Science and Technology, Ufa, Republic of Bashkortostan, Russia

✉ *mikhanko45@gmail.com

FOR CITATION

Mikhanko A. D., Mashkina I. V. A method for determining typical time characteristics of security events based on statistical data for correlation analysis tasks. *Problems of information security. Computer systems*. 2026. No. 2, pp. 32–48. DOI: 10.66424/2071-8217-2026-2-3 (In Russian)

RECEIVED 10.04.2026
ACCEPTED 13.05.2026
PUBLICATION 15.06.2026

ABSTRACT

The article presents a method for determining typical time parameters of information security events based on the analysis of event logs. The method is focused on processing inter-event intervals and makes it possible to identify characteristic temporal patterns of functioning of sources of security events. The proposed approach includes sampling time intervals, identifying the structural gap between event and inter-session intervals, filtering outliers using the interquartile range, and determining typical values based on clustering and group analysis. To account for the variability of the data, an estimate of the mean and standard deviation is used, followed by a division into interval windows. A numerical experiment has been conducted based on data from real-world event logs, confirming the method's operability when analyzing sources with different event generation rates. The experiment was conducted on logs of the OPC server, Windows Server, PostgreSQL database management system. The results obtained demonstrate the method's stability to outliers, multimodal distributions, and the presence of zero intervals. The developed method can be used in the construction of correlation rules in SIEM systems, as well as in the tasks of behavior analysis and detection of anomalies in the information security infrastructure.

KEYWORDS

Information security, event logs, SIEM, time intervals, event intervals, log analysis, anomaly detection, interquartile range, clustering, statistical analysis, event correlation, behavioral analysis

1. ВВЕДЕНИЕ

Системы мониторинга информационной безопасности в информационной среде объекта защиты анализируют угрозы на основе событий безопасности. В информатике под событием понимается действие, инициированное пользователем, программой, устройством или операционной системой и зафиксированное средствами регистрации событий. Упорядоченная по времени последовательность записей о событиях образует журнал событий (или лог-файл) [1], являющийся основным источником данных для анализа информационной безопасности [2].

При этом практическая ценность журналов событий определяется не только объемом собираемых данных, но и возможностью их нормализации и последующего автоматизированного анализа [3].

В центрах обеспечения безопасности одной из основных задач является мониторинг поступающих событий информационной безопасности. Для решения этой задачи используются системы управления событиями безопасности – Security Information and Event Management (SIEM). Такие системы обеспечивают централизованный сбор, агрегацию и анализ событий, поступающих из различных источников инфраструктуры информационной системы (антивирус, межсетевой экран, системы управления базами данных (СУБД), операционные системы…) [4]. Помимо агрегации событий безопасности современные SIEM-системы, системы нового поколения (Next-Generation SIEM, SIEM-NG) и платформы аналитики безопасности (Security Analytics Platform, SAP) позволяют выявлять угрозы и инциденты информационной безопасности на основе анализа журналов событий.

Современное развитие SIEM-системы связано с их переходом от средств централизованного сбора и корреляции событий к комплексным платформам аналитики безопасности и технологическому ядру центров мониторинга безопасности (SOC). В современных SOC такие решения используются совместно с решениями классов SOAR и XDR, платформами Threat Intelligence, средствами долговременного хранения данных и компонентами поведенческой

аналитики. В результате SIEM-система рассматривается не только как средство регистрации и корреляции событий, но и как элемент единой инфраструктуры обнаружения, расследования и реагирования на инциденты информационной безопасности [4–7].

Одновременно возрастает объем и разнородность обрабатываемых журналов событий. В высоконагруженных инфраструктурах потоки событий могут достигать десятков и сотен тысяч событий в секунду, что требует масштабируемых механизмов сбора, нормализации, хранения и последующего анализа как исходных, так и нормализованных данных. В этих условиях качество предварительной обработки журналов, корректная настройка правил корреляции и учет временных характеристик событий напрямую влияют на точность выявления инцидентов и снижения числа ложноположительных срабатываний. Поэтому разработка формализованных методов определения типичных временных параметров источников событий является актуальной задачей для SIEM-систем нового поколения и последующего корреляционного анализа [4, 6–8].

В современных исследованиях отмечается, что эффективность SIEM во многом определяется количеством источников событий безопасности, качеством предварительной нормализации событий и подготовки правил [9, 10].

Большинство SIEM оснащены подсистемой корреляции событий – механизмом, предназначенным для выявления взаимосвязей между событиями безопасности. Анализ функциональных возможностей решений, представленных в ежегодном отчете Magic Quadrant компании Gartner, показывает, что несмотря на внедрение методов машинного обучения и поведенческого анализа, значительная часть механизмов корреляции по-прежнему основана на правилах корреляции, предварительно сформированных экспертами для защищаемой инфраструктуры [11].

2. АКТУАЛЬНОСТЬ ИССЛЕДОВАНИЯ

При разработке правил корреляции событий безопасности перед экспертом по информационной безопасности стоит задача формализации сценария потенциальной атаки. Для построения такого сценария, необходимо знать не только статический состав инфраструктуры (перечень узлов, систем и программного обеспечения), но и динамику протекающих процессов – скорость и последовательность действий нарушителя [12]. Ключевым источником этих сведений являются временные метки событий – это точный момент времени регистрации события в последовательности данных.

При анализе временных интервалов необходимо учитывать их распределение и вариативность. Для решения этой задачи применяется кластеризация.

Для анализа временных интервалов между событиями могут применяться различные методы кластеризации. В статье Т. А. Шевцовой [13] приводится сравнение трех методов. Метод K-means обеспечивает высокую скорость обработки данных, однако требует предварительного задания числа кластеров. Алгоритм DBSCAN позволяет автоматически выделять группы интервалов различной плотности, но чувствителен к выбору параметров. Иерархическая кластеризация позволяет выявлять структуру данных без предварительных предположений о числе кластеров, однако характеризуется высокой вычислительной сложностью.

Актуальность настоящего исследования обусловлена тем, что временные интервалы между событиями не являются детерминированной и постоянной величиной. Обычно события распределены случайным образом на рассматриваемом временном отрезке [14]. Это обстоятельство усложняет выявление взаимосвязи событий и требует применения специальных математических методов для выявления как аномалий, так и закономерностей.

В задачах информационной безопасности анализ временных и поведенческих характеристик событий обычно сопровождается применением методов выявления выбросов и аномалий, поскольку исходные данные отличаются неоднородностью и слабой предсказуемостью [15].

Целью работы является разработка метода определения типичных временных параметров событий безопасности по данным из журналов событий. Предлагаемый

метод основан на фильтрации статистических выбросов интервалов и последующей кластеризации для разделения межсобытийных и межсессионных промежутков, что позволяет вычислить эталонные значения с использованием среднего значения.

3. ПОСТАНОВКА ЗАДАЧИ

В качестве исходных данных используется журнал событий источника безопасности за продолжительный период T. Журнал событий представлен упорядоченной последовательностью событий:

$$E = \{e_1, e_2, ..., e_i, ..., e_I\},$$

где e_i – i-е событие в журнале, $i = \{1, 2, ..., I\}$; I – количество записей в журнале.

Каждому событию e_i соответствует момент времени t_i. Предполагается, что временные метки упорядочены неубывающим образом:

$$t_1 \le t_2 \le ... \le t_I .$$

Выборка временных интервалов между соседними событиями представлена множеством Δ как

$$\Delta = \{\delta_1, \delta_2, ..., \delta_{I-1}\},$$

где $\delta_i = t_i - t_{i-1}$.

Распределение межсобытийных интервалов в журнале событий может быть неоднородным. В выборке могут присутствовать как короткие интервалы между соседними событиями в рамках одной активности, так и длительные интервалы между рабочими сессиями. Таким образом, множество временных интервалов Δ может быть представлено как объединение нескольких подмножеств, соответствующих различным типам временных интервалов. Задача исследования заключается в определении типичных временных параметров событий источника безопасности на основе анализа журнала событий.

Формально требуется по выборке межсобытийных интервалов Δ разделить интервалы между соседними событиями и интервалы между рабочими сессиями; исключить из анализа аномальные значения интервалов; определить типичный временной интервал регистрации событий и допустимые границы его изменения. Для решения поставленной задачи используются статистические характеристики распределения интервалов.

Межквартильный размах – это мера статистической дисперсии, показывающая диапазон, в котором сосредоточено некоторое количество (чаще всего 50 %) центральных значений выборки. Квартили – значения, разделяющие множество на четыре части [14]. Стандартное отклонение – мера рассеивания значений, относительно среднего, характеризующая вариативность данных.

Использование указанных характеристик позволяет выделить типичный диапазон временных интервалов между событиями и определить допустимые границы их изменения для рассматриваемого источника событий безопасности.

Математическое описание метода определения временных параметров событий безопасности на основе журнала событий. Интервалы δ_i – являются вещественными числами. Тогда из выборки Δ удаляются нулевые значения, после чего оставшиеся интервалы сортируются по возрастанию. Обозначим через *sort*(Δ) последовательность элементов множества Δ, упорядоченных по возрастанию. Тогда полученная упорядоченная выборка обозначается S:

$$S = sort\left(\{\delta_i \in \Delta | \delta_i > 0\}\right),$$

$$S = \{s_1, s_2, ..., s_m, ..., s_M\},$$

где $s_1 \le s_2 \le ... \le s_M$, $M = |S|$ – размер выборки S.

Упорядоченная выборка межсобытийных интервалов может содержать интервалы различной природы: короткие между соседними событиями и более длительные интервалы между сессиями активности. Для разделения этих типов определяется точка максимального структурного разрыва между соседними элементами выборки. Распределение интервалов выборки S можно представить функцией, показанной на рис. 1.

Поскольку выборка упорядочена по возрастанию, структурный разрыв представляет собой точку перехода к максимальным

значениям – предполагаемым интервалам между рабочими сессиями. Искомый структурный разрыв представлен точкой на рис. 1.

Для определения структурного разрыва выборки S необходимо ввести последовательность разностей между соседними элементами:

$$G = \{g_1, g_2, \ldots, g_m, \ldots, g_{M-1}\},$$

где $g_m = s_{m+1} - s_m$; $|G| = M - 1$ – мощность последовательности G.

Последовательность G можно представить функцией, изображенной на рис. 2.

Поскольку элементы выборки S – это упорядоченные межсобытийные интервалы, то в начале выборки расположены самые короткие временные переходы между событиями, в конце – самые длинные.

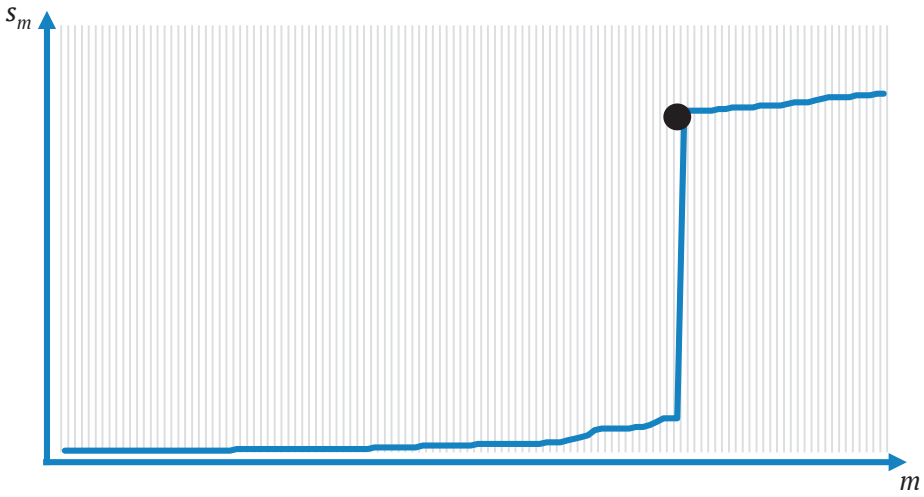


Рис. 1 | Распределение упорядоченной выборки межсобытийных интервалов времени

Fig. 1 | Distribution of an ordered sample of inter-event time intervals

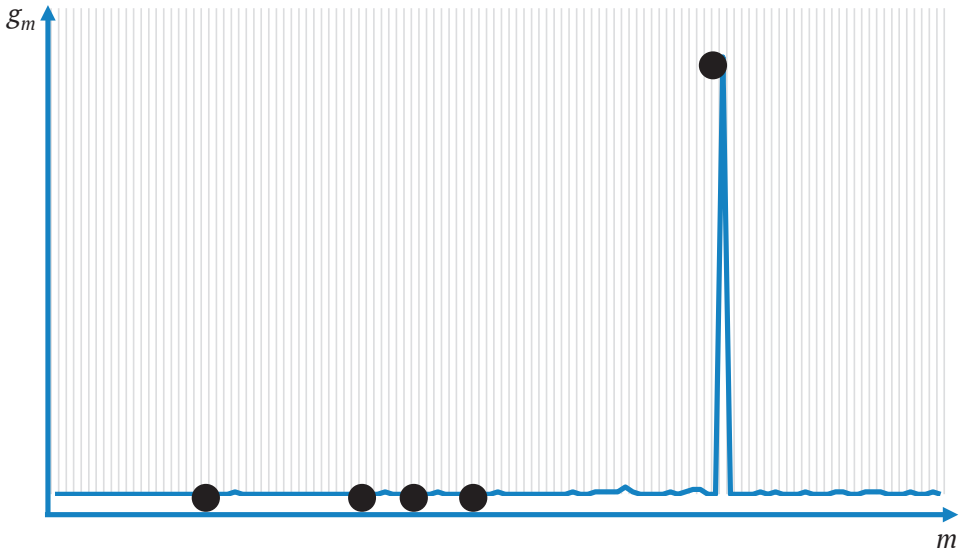


Рис. 2 | График распределения разностей элементов множества S
● – ненулевые разности

Fig. 2 | Graph of the distribution of the differences of the elements of the set S
● – nonzero differences

Тогда отмеченные на рис. 2 точки – это те точки, когда события на источнике начинают регистрироваться в логе с новым временным интервалом. Эти точки принимаем за точки структурного разрыва выборки *S*.

Тогда индекс точки максимального структурного разрыва определяет переход между короткими и длинными интервалами:

$$m^* = \arg\max_{1 \le m \le M-1} G,$$

где max*G* – поиск максимального значения последовательности *G*; arg max*G* – выбор индекса максимального элемента последовательности *G*.

Определенный индекс m^* – это точка, когда выборка *S* разделяется на интервалы между событиями и интервалы между рабочими сессиями. Формируется выборка потенциальных временных интервалов между событиями и между рабочими сессиями. Упорядоченная выборка *S* содержит интервалы двух типов: между соседними событиями и между рабочими сессиями.

Предполагается, что интервалы между событиями меньше интервалов между рабочими сессиями, что позволяет разделить выборку с использованием m^*. Тогда выборка потенциальных временных интервалов между событиями определена как S_{ev}, выборка потенциальных интервалов между сессиями – как S_s:

$$S_{ev} = \{s_1, s_2, ..., s_{m^*}\},$$

$$S_s = \{s_{m^*+1}, ..., s_M\},$$

где $s \in S$.

Таким образом, выполняется соотношение $S = S_{ev} \cup S_s$.

Если max*G* = 0, то в выборке *S* отсутствует структурный разрыв и разделение на S_{ev} и S_s не выполняется. Поскольку основная выборка *S* упорядочена по возрастанию, производные выборки S_{ev} и S_s также упорядочены по возрастанию.

После формирования S_{ev} и S_s должно выполняться следующее условие: $\min S_s > \max S_{ev}$, в противном случае разделение выборки *S* выполнено некорректно.

Для выборки потенциальных межсобытийных интервалов S_{ev} вычисляет межквартильный размах. Выборка *S* содержит временные интервалы между событиями безопасности за продолжительный период *T*. Предполагается, что за рассматриваемое время зарегистрированы как легитимные события, так и события, вызванные действиями нарушителя или сбоями в работе системы регистрации. Цель исследования – определение временных характеристик нормального поведения источника событий безопасности, поэтому нелегитимные события необходимо исключить из анализа.

Пусть упорядоченная выборка потенциальных межсобытийных интервалов имеет вид:

$$S_{ev} = \{s_1, ..., s_{m^*}\}.$$

Тогда квартиль первой четверти Q1 выборки S_{ev} соответствует значению, ниже которого находится приблизительно 25 % элементов выборки [16], и определен как:

$$Q_1 = s_{\lceil 0{,}25m^* \rceil}.$$

Квартиль третьей четверти Q3 соответствует значению, ниже которого находится приблизительно 75 % элементов выборки [16], и определен как:

$$Q_3 = s_{\lceil 0{,}75m^* \rceil}.$$

Использование границ 25 и 75 % выборки введено Джоном Тьюки как компромисс между устойчивостью и чувствительностью к выбросам. Использование этих границ позволяет выделить центральные 50 % данных, характеризующие типичное поведение выборки. Индексы квартилей округляются вверх до ближайшего целого значения.

Межквартильный размах определен как:

$$IQR = Q_3 - Q_1.$$

Межквартильный размах характеризует диапазон значений, содержащий центральную часть выборки, и используется для определения границ фильтрации аномальных интервалов [16].

Выборка S_{ev} может содержать как типичные интервалы между событиями, так

и аномальные значения. Тогда отфильтрованные промежутки времени формируют выборку S'_{ev} как:

$$S'_{ev} = \left\{ s_i \in S_{ev} \middle| \left(Q_1 - 1{,}5IQR\right) \le s_i \le \left(Q_3 + 1{,}5IQR\right) \right\},$$

где 1,5 – стандартная практика *IQR* для сохранения баланса между чувствительностью (выявлением аномалий) и сохранением данных.

На рис. 3 представлено распределение выборки S_{ev} до и после фильтрации.

Даже после фильтрации выборка интервалов между событиями содержит неоднородные данные. Необходимо определить тот интервал, который считается нормальным для рассматриваемого источника событий. Допускается, что после фильтрации в выборке преобладает искомое значение. Тогда можно предположить, что среднее значение выборки будет близко к искомому. Среднее значение выборки S'_{ev} определяется как:

$$\overline{S'_{ev}} = \frac{1}{\left|S'_{ev}\right|} \sum_{s_i \in S'_{ev}} s_i .$$

В идеальной среде интервалы между событиями источника имеют стабильное значение. Однако на практике на источник событий воздействуют внешние факторы, поэтому легитимный интервал меняется. Предлагается в качестве допустимых границ изменения искомого значения использовать стандартное отклонение от среднего значения выборки:

$$\sigma = \sqrt{\frac{1}{\left|S'_{ev}\right|} \sum_{s_i \in S'_{ev}} \left(s_i - \overline{S'_{ev}}\right)^2} .$$

Среднее значение является максимально близким к искомому значению, но не эталонным. Предполагается, что в отфильтрованной выборке значения, принадлежащие интервалу допустимого отклонения и имеющие наибольшую плотность распределения, будут являться искомыми. Тогда необходимо определить все возможные интервальные окна выборки S'_{ev}, после чего определить интервальное окно с самым большим количеством элементов. Поскольку интервалы положительные, окна определяются как:

$$K_j = \left\{ s_i \in S'_{ev} \middle| (j-1)\sigma < s_i \le j\sigma \right\},$$

где σ – стандартное отклонение S'_{ev}, если $\sigma = 0$, то все элементы равны, типичное значение интервала определяется как $\overline{S'_{ev}}$, без разбиения на окна; j – порядковый номер окна интервалов, $j \in \{1, 2, \ldots, J\}$; J – количество окон интервалов, $J = \left\lceil \frac{\max\left(S'_{ev}\right)}{\sigma} \right\rceil$, с округлением вверх до целого. Скобками «$\lceil\ \rceil$» обозначается округление до целого числа.

Индекс окна интервалов, которое можно считать типичным окном временных интервалов между событиями на источнике, определен как:

$$j^* = \arg\max_{1 \le j \le J} \left| K_j \right| .$$

Поскольку j^* – индекс интервального окна с самым большим количеством элементов, то окно интервалов перехода между событиями для источника K_{j^*}.

Таким образом, типичный временной интервал между событиями определяется границами окна K_{j^*}.

Проверка структурной значимости интервального окна. Для проверки рассмотрим исходную выборку событий $E = \{e_1, \ldots, e_I\}$, где каждому событию e_i соответствует временная метка t_i. В рамках проверки структурной значимости интервального окна дополнительно учитывается тип события. Тогда событие можно представить кортежем:

$$e_i = \left(c(e_i), t_i\right),$$

где $c(e_i)$ – тип события; t_i – временная метка события.

В основном методе анализа нулевые значения временных интервалов исключаются из дальнейшего рассмотрения, поскольку они не характеризуют временной переход между событиями. Для сохранения этой логики необходимо сформировать временной слой – события, зарегистрированные в одно время.

Пусть множество уникальных временных меток определяется как:

$$\tau = \left\{ \tau_1, \ldots, \tau_H \right\},$$

где $\tau_1 < \tau_2 \ldots < \tau_H$, а каждая τ_h является одной из временных меток исходного журнала.

Для каждой уникальной временной метки τ_h формируется временной слой событий:

$$B_h = \{e_i \in E | t_i = \tau_h\}.$$

Множество всех временных слоев обозначим как $B=\{B_1,\ldots,B_h\}$. Временной слой B_h не является новым самостоятельным событием и не заменяет события исходного журнала. Он представляет собой служебный контейнер неопределенного внутреннего порядка, содержащий все события, зарегистрированные с одной временной меткой. Неопределенность внутреннего порядка означает, что для событий внутри одного слоя отсутствует ненулевой временной интервал, поэтому на основании временных меток нельзя обоснованно утверждать, какое из этих событий произошло раньше или позже.

Пусть C_h – множество типов событий, входящих во временной слой B_h:

$$C_h = \{c(e_i) | e_i \in B_h\}.$$

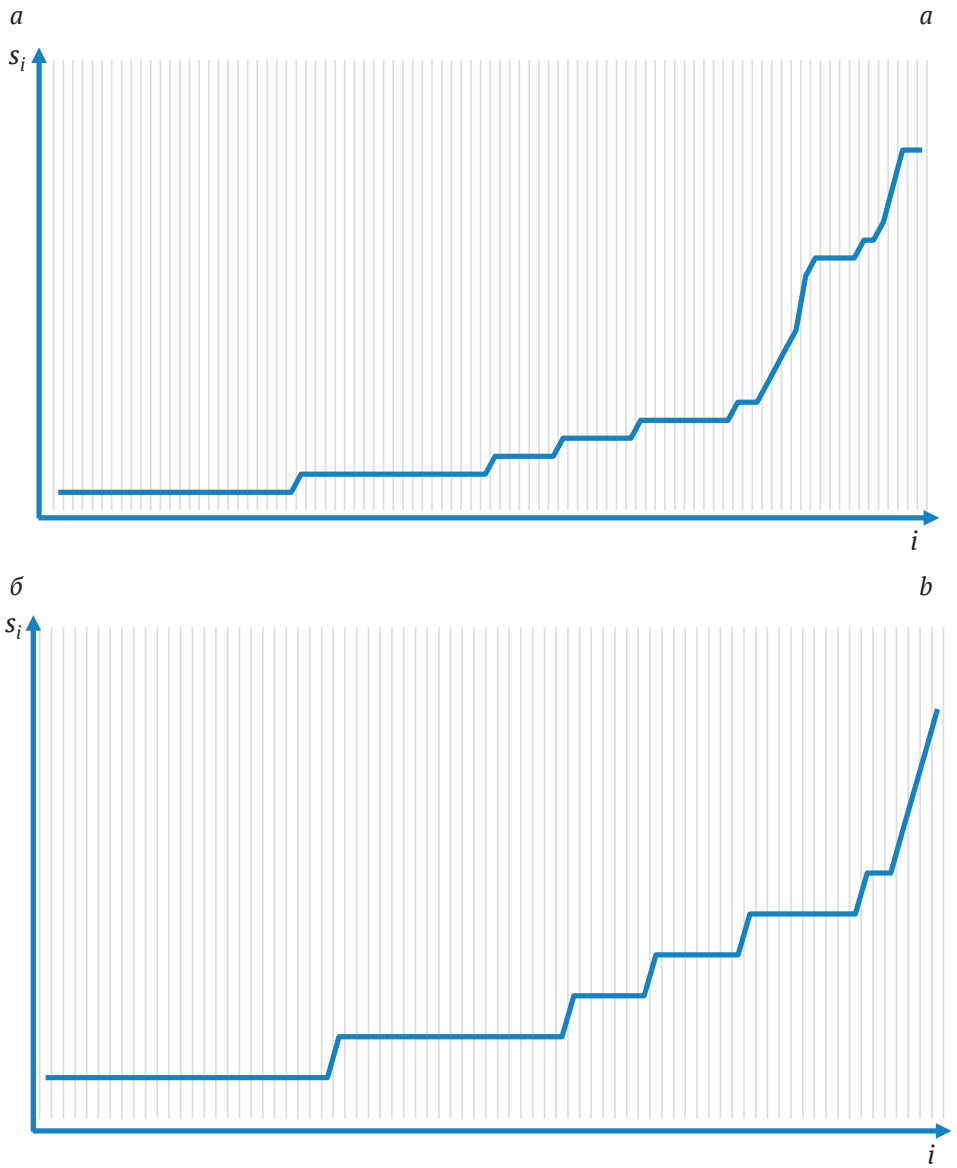


Рис. 3 | Распределение выборки S_{ev} (*а*) и отфильтрованной выборки S'_{ev} (*б*)

Fig. 3 | Distribution of S_{ev} sample (*a*) and filtered S'_{ev} sample (*b*)

Тогда для двух временных слоев B_h и B_p множество потенциальных переходов между типами событий определяется как:

$$R_h = \{(a,b) | a \in C_h, b \in C_p,$$
$$0 < \tau_p - \tau_h \le \max(S'_{ev}), p > h\}.$$

Здесь пара (a, b) означает, что событие типа a зарегистрировано в предыдущем временном слое, а событие типа b в следующем, при этом время между этими слоями не превышает максимально найденный интервал времени между событиями. Такая связь не утверждает наличие причинно-следственной зависимости между событиями, а фиксирует только потенциальный временной переход между типами событий.

Полное множество наблюдаемых переходов по журналу определяется как объединение переходов между всеми соседними временными слоями:

$$R = \bigcup_{h=1}^{H-1} R_h .$$

Для оценки устойчивости переходов каждому ребру $(a, b) \in R$ ставится в соответствие вес:

$$\omega(a,b) = \left|\{h | a \in C_h, b \in C_{h+1}, h = 1,...,H-1\}\right|.$$

Вес $\omega(a,b)$ показывает, сколько раз переход от события типа a к событию типа b наблюдается между соседними временными слоями журнала. Тогда граф переходов между временными слоями можно определить как взвешенный ориентированный граф:

$$G = (V, R, \omega),$$

где $V = \bigcup_{h=1}^{H} C_h$ – множество вершин графа, соответствующих типам событий; R – множество ориентированных ребер между типами событий, сформированных по соседним временными слоям; ω – вес ребра, определяющий частоту наблюдения соответствующего перехода.

Построенный граф отражает все потенциальные переходы между типами событий, наблюдаемые между соседними временными слоями. Однако наличие ребра в таком графе не означает, что существующий переход является устойчивым или полезным для последующего корреляционного анализа. Часть ребер может возникнуть случайно за счет высокой плотности регистрации событий, частого появления отдельных типов событий или пакетного характера логирования.

Для оценки статистической обоснованности ребра введем ожидаемую частоту перехода между типами событий. Она показывает, сколько раз переход ($a \to b$) мог бы возникнуть в рассматриваемом графе при условии независимого появления типов событий в соседних временных слоях.

Пусть $a, b \in V$ – типы событий. Вероятность появления типа события a во временном слое определяется как доля временных слоев, наблюдающих хотя бы одно событие данного типа:

$$P(a) = \frac{\left|\{h | a \in C_h, h = 1,...,H\}\right|}{H}.$$

Аналогично для типа b:

$$P(b) = \frac{\left|\{h | b \in C_h, h = 1,...,H\}\right|}{H}.$$

Суммарный вес всех ребер графа определяется как:

$$W = \sum_{(a,b) \in R} \omega(a,b).$$

Тогда математическое ожидание частоты перехода от типа событий a к типу событий b определяется как:

$$M(a, b) = WP(a)P(b).$$

Величина $M(a,b)$ показывает ожидаемое количество появлений перехода (a,b) в рассматриваемом графе при случайном независимом распределении типов событий по соседним временным слоям.

Если фактический вес ребра $\omega(a,b)$ меньше ожидаемого значения $M(a,b)$, то такой переход встречается реже, чем можно было бы ожидать при случайном сочетании типов событий. Такое ребро рассматривается как статистически слабое и исключается из дальнейшего анализа. На основе сравнения фактического ребра с ожидаемой частотой выделим множество статистически подтвержденных переходов:

$$R^* = \{(a,b) \in R \mid \omega(a,b) \ge M(a,b)\}.$$

Переходы множества R^* обладают статистическим основанием и могут рассматриваться как более устойчивые по сравнению со связями, возникшими реже ожидаемого уровня.

Для количественной оценки структуры графа введет коэффициент структурной устойчивости:

$$Q_{global} = \frac{\sum_{(a,b)\in R^*}\omega(a,b)}{\sum_{(a,b)\in R}\omega(a,b)}.$$

Значение Q_{global} показывает, какая доля наблюдаемых переходов в исходном графе приходится на статистически подтвержденные связи. Чем выше значение Q_{global}, тем большая часть переходов между временными слоями объясняется устойчивыми сочетаниями типов событий, а не случайными связями.

Для каждого найденного окна K_j формируется частный граф переходов: $G_j = (V_j, R_j, w_j)$, в котором учитываются только переходы между такими временными слоями B_h и B_p, для которых временной интервал между слоями принадлежит рассматриваемому окну:

$$R_{hj} = \left\{(a,b)\middle| a \in C_h, b \in C_p, \tau_p - \tau_h \in K_j, p > h\right\}.$$

Тогда все возможные связи между событиями в частном графе R_j определяются как:

$$R_j = \bigcup_{hj=1}^{Hj-1} R_{hj}.$$

После применения метода очистки графа от случайных связей определяется частная Q_j, отражающая какая доля переходов внутри интервального окна K_j приходится на статистически подверженные связи. Если значение Q_j превышает или не меньше значения Q_{global}, то рассматриваемое временно окно сохраняет не меньшую долю устойчивых переходов, чем исходных граф журнала:

$$Q_j \geq Q_{global}.$$

Такое окно можно считать структурно подтвержденным для задач последующего корреляционного анализа. Если анализ выявляет несколько допустимых интервальных окон, то каждое такое окно характеризует нормальный режим работы источника событий. В этом случае результатом метода является не единственное типичное значение, а множество допустимых интервальных окон, отражающих вариативность штатного функционирования источника.

Полученная математическая модель формализует процедуру определения типичных временных параметров событий безопасности по журналам событий и позволяет последовательно решить четыре взаимосвязанные задачи: отделить межсобытийные интервалы от межсессионных, исключить аномальные значения из анализа, выделить наиболее характерные интервальные окна нормального функционирования источника событий и выполнить их структурную проверку для задач последующего корреляционного анализа. В отличие от прямого усреднения исходной выборки, предложенная модель учитывает неоднородность временных распределений, наличие выбросов и вариативность реальных журналов регистрации и возможность существования нескольких нормальных режимов работы источника безопасности. Полученная модель формализует процедуру определения типичных временных параметров событий безопасности.

Методика определения типичных межсобытийных интервалов. На основе разработанной математической модели предложена методика, предназначенная для определения типичных временных интервалов между событиями и интервалов между рабочими сессиями источника безопасности. Методика позволяет разделить межсобытийные интервалы на короткие и длинные, удалить аномальные значения и определить типичные временные параметры генерации событий источника. Поскольку на реальный источник событий могут воздействовать внешние факторы, метод также позволяет определить допустимое отклонение временных интервалов. Методики реализуются в виде последовательности этапов обработки журнала событий.

Этап 1. Получение данных. Метки времени событий, зарегистрированных в лог-файлах, могут быть представлены

строкой даты и времени, только временем либо количеством миллисекунд от начала Unix эпохи [17].

Этап 2. Формирование выборки временных интервалов. Извлеченные метки времени отображают момент регистрации событий. Однако для анализа требуется определить временные интервалы между соседними событиями.

Этап 3. Упорядочивание выборки временных интервалов. Полученные промежутки между соседними событиями распределены по всей выборке случайным образом. Поскольку метод не зависит от последовательности событий в процессах, полученную выборку интервалов можно упорядочить по возрастанию.

Этап 4. Очистка от нулевых значений. Если система регистрации событий зарегистрировала несколько событий в одно и то же время, то интервал между ними будет нулевым. Такие интервалы не несут информации о временной динамике событий и исключаются из дальнейшего анализа.

Этап 5. Определение максимального структурного разрыва выборки. Сформированная выборка содержит промежутки как между событиями, так и между рабочими сессиями. Для их разделения определяется наибольшая разница между соседними интервалами.

Этап 6. Формирование выборок потенциальных интервалов между событиями и потенциальных интервалов между рабочими сессиями. Определенный ранее разрыв позволяет разделить исходную выборку на две части. Допущение метода – интервал между событиями будет меньше, чем интервал между сессиями. На основе разрыва формируются две новые выборки: в первую попадают значения, индекс которых меньше индекса разрыва – потенциальные интервалы между событиями, во вторую – оставшиеся элементы, т.е. потенциальные интервалы между сессиями. Если максимальный разрыв равен нулю, значит, все промежутки между событиями одинаковые. Тогда выборка остается целой.

Этап 7. Фильтрация данных. Среди элементов новой выборки находятся как легитимные, так и ошибочные данные. Чтобы исключить ошибочные элементы множества, необходимо воспользоваться фильтрацией на основе межквартильного размаха. Допущение метода – считается, что основную часть времени источник событий работает в штатном режиме, значит промежуток времени между событиями будет встречаться чаще других интервалов, и, исключив значения, выходящие за пределы межквартильного диапазона, можно допустить, что легитимные элементы будут преобладать в остатке.

Этап 8. Ввод допустимого отклонения отфильтрованной выборки. Несмотря на очистку выборки от ошибочных данных, выборка все еще содержит разнородные элементы, среди которых необходимо определить тот элемент, который считается типичным временем между двумя соседними событиями. Для определения типичного интервала вычисляется среднее значение отфильтрованной выборки. Если на шестом этапе не было проведено разделение выборки на две, то среднее значение является типичным промежутком между соседними событиями.

В реальных условиях интервалы между событиями могут изменяться под воздействием внешних факторов. Необходимо определить допустимое отклонение на основе среднего значения. Стандартное отклонение используется для определения ширины интервалов (окон), применяемых при группировке временных интервалов.

Этап 9. Разбиение выборки на окна. Поскольку для реального объекта типичное время перехода между событиями – это строгая оценка, необходимо определить окно интервалов. После определения стандартного отклонения выборка разбивается на интервальные окна. Каждое окно содержит значения интервалов, попадающие в соответствующий диапазон. Интервальное окно с наибольшим количеством элементов рассматривается как основной кандидат на типичное временное окно источника событий.

Этап 10. Проверка структурной значимости каждого найденного интервального окна. Для каждого найденного интервального окна проводится проверка его пригодности для последующего корреляцион-

ного анализа. Для этого строится частный граф потенциальных связей между типами событий, в который включаются только те связи, временной интервал между которыми принадлежит рассматриваемому окну. Для полученного графа вычисляется коэффициент структурной устойчивости Q_j, отражающий долю статистически подтвержденных связей. Затем значение Q_j сравнивается с базовым коэффициентом Q_{global}, рассчитанным для графа потенционых связей без ограничения конкретным интервальным окном. Если $Q_j \geq Q_{global}$, то соответствующее окно считается структурно подтвержденным. Если проверку проходят несколько окон, они рассматриваются как различные нормальные режимы работы источника событий.

Этап 11. Этапы 7–10 выполняются независимо для каждой из выборок, сформированных на шестом этапе.

В результате выполнения описанных этапов определяется интервальное окно, соответствующее типичным временным интервалам между событиями источника.

4. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Для системы мониторинга информационной безопасности характерна обработка разнородных событий, отличающихся по частоте регистрации, структуре записи и механизмам логирования (Данилова & Абельдинов, 2024). Поэтому для апробации предложенного метода использовались журналы событий, полученные с сервера промышленной сети. Проанализированы журналы OPC-сервера, журнал безопасности Windows и журнал СУБД PostgreSQL. Предложенные источники событий информационной безопасности функционируют на действующем промышленном объекте, характеризуются различной интенсивностью регистрации событий. В табл. 1 представлены результаты сравнения.

Журнал OPC-сервера содержит 1500000 записей за час работы системы. Реализация алгоритма произведена средствами MS Excel для наглядного представления этапов обработки данных. В результате анализа установлено, что значительная часть событий имеет нулевые межсобытийные интервалы, что связано с пакетной фиксацией событий OPC-сервером. После удаления нулевых интервалов и применения фильтрации методом межквартильного размаха определен типичный интервал регистрации событий сервером. Это окно подтверждено методом проверки, однако, так как для анализа использована небольшая выборка за час работы, для достижения порога устойчивости $Q_{global} = 0{,}87$ необходимо было, чтобы максимальное число связей, полученное после фильтрации временными рамками, оказалось статистически подтвержденными. В рассматриваемой выборке только первое окно достигло 100 % сохраненных связей, остальные окна не достигли 10 %.

Журнал событий безопасности Windows содержит 29000 записей за 2 мес работы и примерно 95 % нулевых интервалов между событиями с точностью 1 с. Это связано с особенностями системы регистрации событий безопасности Windows Server. По результатам исследования выборки выявлена мультимодальность распределения, т.е. два выраженных пика: окно 1 = 48 % интервалов между событиями, окно 3 = 33 %. Проверка этих окон

Таблица 1 | Сравнение выборок для численного эксперимента

Table 1 | Comparison of samples for numerical experiment

Источник	Количество анализируемых событий	Средняя частота, соб./ч	Интенсивность регистрации событий
OPC-сервер	1500000	1,5 млн	Высокая
PostgreSQL	52000	42 тыс.	Средняя
Журнал безопасности Windows Server	29000	20	Низкая

показала, что при пороге Q_{global}=0,77 показывают примерно одинаковые показатели сохранения связей, при этом менее плотное окно сохраняет на 11 % больше связей, чем более плотное. По этим результатам можно сделать вывод, что мультимодальность является следствием двух равнозначных режимов работы.

Механизм логирования СУБД PostgreSQL записывает в файл все SQL. Каждый файл содержит не больше 10 Гб записей, после чего формируется новый файл. Так, за 80 мин зафиксировано 52500 событий, 97 % нулевых интервалов с точностью 1 с. Поскольку отфильтрованная выборка содержит одинаковые значения, то и последующая работа не требуется, стандартный интервал между событиями СУБД PostgreSQL равен 1 с. Результаты экспериментов представлены в табл. 2.

Для проведения сравнительного анализа рассчитаны значения медианы и моды по выборке межсобытийных интервалов после удаления выбросов методом межквартильного размаха. Медиана определялась как центральное значение упорядоченной выборки, а при четном количестве элементов – как среднее двух центральных значений. Мода – это значение самого частого межсобытийного интервала в очищенной выборке. Полученные значения применены в методе проверки структурной значимости. Результаты представлены в табл. 3.

Таблица 2 | Результаты численного метода

Table 2 | Results of the numerical method

Метрика	OPC-сервер	Windows Server	СУБД PostgreSQL
Нулевые значения, %	79	94,5	97
Максимальный структурный разрыв	66 мс	112184 с	1 с
Разделение межсобытийных интервалов	216458	1522	1359
Разделение межсобытийных интервалов после фильтрации	184657	1520	
Средний интервал	2,124 мс	3235 с	
Стандартное отклонение	2,253 мс	2873 с	
Сформировано окон	6	5	
Наиболее плотное окно	1	1	
Q важных окон	Окно 1: 1	Окно 1: 0.83 Окно 3: 0.94	
Плотность окна	144454 (78 %)	731 (48 %)	
Допустимый временной интервал источника	$0<\delta\leq$ 2,253 мс	$0<\delta\leq$ 2873 с	1 с

Таблица 3 | Результаты сравнения методов

Table 3 | Results of comparison of methods

Метрика	Журнал OPC	Журнал безопасности Windows Server
Допустимый временной интервал	$0<\delta\leq$ 2,253 мс	$0<\delta\leq$ 2873 с
Медиана	1 мс	30 с
Мода	1 мс	30 с
Q допустимого интервала	1	0,83
Q медианы	0,87	0,74
Q моды	0,87	0,74

По результатам сравнения, представленным в табл. 3, видно, что медиана и мода после фильтрации *IQR* дают близкие к границе Q_{global} характеристики, однако предложенный метод создает условия, при которых межсобытийная связь более устойчивая. Попадание медианного или модального значения в границы найденного интервального окна не противоречит результатам предложенного метода, напротив, оно показывает, что точечная оценка может находиться внутри области типичного поведения, однако сама по себе не позволяет определить границы этой области.

Анализ данных, представленных в табл. 2, подтверждает гипотезу о высокой доле избыточной информации в исходных журналах событий: наличие нулевых интервалов свидетельствует о специфике механизмов логирования, фиксирующих группы событий в рамках одной секунды или миллисекунды. Предложенный метод успешно нивелирует влияние этой избыточности, позволяя выделить значимые межсобытийные интервалы. В то время как классические статистические методы могли бы дать смещенную оценку из-за наличия нескольких типичных окон активности, использование группового анализа и выделение «наиболее плотного окна», а также метод подтверждения структурной значимости каждого выявленного окна, позволили корректно определить границы типичного поведения источника.

Полученные результаты демонстрируют, что фильтрация на основе межквартильного размаха в сочетании с поиском максимального структурного разрыва эффективно отделяет штатную активность от редких сессионных всплесков и случайных выбросов. Сформированные для каждого источника «типичные окна» (например, до 2,253 мс для OPC-сервера и ровно 1 с для СУБД PostgreSQL) являются готовыми параметрами для настройки поведенческих профилей. Это позволяет формализовать границы нормальности для каждого типа источника в инфраструктуре, что критически важно для минимизации ложных срабатываний при последующем корреляционном анализе в SIEM-системах.

5. ЗАКЛЮЧЕНИЕ

Описывается метод определения типичных временных параметров событий безопасности на основе анализа журналов событий. Предложенный подход позволяет по накопленным данным выявлять характерные межсобытийные интервалы, отделять их от интервалов между рабочими сессиями и определять допустимые границы изменения временных параметров источника событий.

В отличие от прямого статистического усреднения всей выборки, предложенный метод последовательно учитывает особенности реальных журналов событий: наличие нулевых интервалов, выбросов, неоднородности данных и мультимодальности распределения. Для этого используется сортировка межсобытийных интервалов, поиск максимального структурного разрыва, фильтрация методом межквартильного размаха и последующее выделение наиболее плотного интервального окна на основе стандартного отклонения, после чего все найденные окна подтверждаются или отклоняются на основе проверки структурной значимости окна.

По результатам численного эксперимента на журналах OPC-сервера, Windows Server и СУБД PostgreSQL подтверждена работоспособность метода для источников событий с высокой, средней и низкой интенсивностью регистрации. Установлено, что метод сохраняет применимость даже при значительной доле нулевых интервалов между событиями, достигающей 79–97 %, и позволяет выделять типичные временные параметры в условиях различной структуры логирования.

Для журнала OPC-сервера определено типичное окно межсобытийных интервалов $0 < \delta <= 2{,}253$ мс, содержащее 78 % элементов отфильтрованной выборки. Для журнала безопасности Windows Server метод позволил корректно выделить наиболее плотное окно $0 < \delta <= 2873$ с, в то же время определить наиболее значимое окно $5746 < \delta <= 8619$ с в условиях мультимодального распределения, где ручной выбор интервала мог привести к ошибке.

Для СУБД PostgreSQL установлено, что после фильтрации все значения интервалов совпадают, вследствие чего типичный интервал определяется однозначно – 1 с.

Практическая значимость работы заключается в том, что полученные типичные временные параметры могут быть использованы при формировании правил корреляции событий в SIEM-системах, а также при решении задач профилирования поведения источников событий и выявления отклонений от нормального режима функционирования.

Дальнейшее развитие исследования целесообразно связать с распространением метода на совокупность источников событий безопасности, автоматизацией обработки в специализированном программном модуле, а также с разработкой критериев объединения временных окон отдельных источников в рабочие сессии всей информационной системы.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Шамсутдинова Т. М.** Цифровой след как источник больших данных (Big Data) в образовании // Открытое образование. 2024. Т. 28. № 6. С. 13–21. DOI: 10.21686/1818-4243-2024-6-13-21.
2. **Жаксыбай С. М.** Управление событиями информационной безопасности с помощью SIEM-системы // Интеллектуальные технологии в транспорте. 2023. № 1. С. 66–69.
3. **Москвичева К. С., Сай С. В.** Нормализация журналов событий с использованием регулярных выражений // Мировые исследования в области естественных и технических наук, 30 апреля 2023 г., Ставрополь. Ставрополь: ООО «Ставропольское издательство “Параграф”». С. 215–219.
4. **Коклянов А. Е.** Применение SIEM-систем в ходе проведения учений на платформе киберполигона // Математическое моделирование, компьютерный и натурный эксперимент в естественных науках. 2025. № 1. С. 45–51. DOI: 10.24412/2541-9269-2025-1-45-51.
5. **Листратор И. С., Милославская Н. Г., Сирбай И. С., Рейносо Б. А.** Расширенная модель зрелости SOC компании Cybereason // Безопасность информационных технологий. 2025. Т. 32. № 1. С. 68–84.
6. **Кузнецов А. В.** Организация раздельного хранения данных о событиях безопасности // Вопросы кибербезопасности. 2024. Т. 60. № 2. С. 22–28.
7. **Пахомов В. В.** Оценка эффективности центров мониторинга и реагирования на киберугрозы: ограничения временных метрик и операционные индикаторы качества // Моделирование, оптимизация и информационные технологии. 2025. Т. 13. № 4. DOI: 10.26102/2310-6018/2025.51.4.040.
8. **Долгачев М. В., Костюнин В. А.** Комплексный анализ поведения системы Windows для обнаружения киберугроз // Вопросы кибербезопасности. 2025. Т. 66. № 2. С. 71–77.
9. **Данилова О. Т., Абельдинов Р. М.** Практическое применение SIEM-модели на базе платформы Elastic Stack для мониторинга системы информационной безопасности // Динамика систем, механизмов и машин. 2024. Т. 12. № 1. С. 105–113. DOI: 10.25206/2310-9793-2024-12-1-105-113.
10. **Путято М. М., Макарян А. С., Черкасов А. Н., Кучер В. А.** Оценка функционирования SIEM-систем на основе комплекса критериев эффективности // Вестник Адыгейского государственного университета. Серия 4: Естественно-математические и технические науки. 2024. № 1. С. 36–42. DOI: 10.53598/2410-3225-2024-1-336-36-42.

11. Gartner Magic Quadrant for Security Information and Event Management // Gartner Research. URL: https://www.gartner.com/en/documents/7040298 (дата обращения: 10.03.2026).
12. **Машкина И. В., Уразаева А. М.** Метод разработки базы знаний сценариев угроз для системы реагирования на инциденты (IRP) // Известия Южного федерального университета. Технические науки. 2024. № 5. С. 79–88.
13. **Шевцова Т. А.** Выявление аномалий в сложных данных с помощью кластеризации // Профессиональный бюллетень: Информационные технологии и безопасность. 2024. № 4. С. 39–46.
14. **Yang Chen, Yijia Ma, Wei Wu**. Rank-Based Mixture Models for Temporal Point Processes // Front. Appl. Math. Stat. 2022. Vol. 8. DOI: 10.3389/fams.2022.852314.
15. **Кечеджиев А. С., Цветкова О. Л.** Исследование обнаружения аномалий с использованием Isolation Forest в машинном обучении // Вестник Дагестанского государственного технического университета. Технические науки. 2024. Т. 51. № 1. С. 106–112. DOI: 10.21822/2073-6185-2024-51-1-106-112.
16. **Дорофеев В. С., Волосатова Т. М.** Ансамблирование методов обнаружения выбросов при подготовке обучающей выборки данных // Научный журнал Моделирование, оптимизация и информационные технологии. 2022. Т. 10. № 3. С. 1–13. DOI: 10.26102/2310-6018/2022.38.3.013.
17. Timestamp formats in logs // new relic. URL: https://docs.newrelic.com/docs/logs/ui-data/timestamp-support/ (дата обращения: 03.03.2026).

REFERENCES

1. **Shamsutdinova T. M.** Digital Footprint as a Source of Big Data in Education. *Open Education*. 2024. Vol. 28. No. 6, pp. 13–21. DOI: 10.21686/10.21686/1818-4243-2024-6-13-21. (In Russian)
2. **Zhaksybay S. M.** Information Security Event Management Using SIEM System**.** *Intellektual'nye tehnologii v transporte*. 2023. No. 1, pp. 66–69. (In Russian)
3. **Moskvicheva K. S., Say S. V.** Normalizing event logs using regular expressions. World Research in the field of natural and technical sciences, 30 April 2023, Stavropol. Stavropol: OOO "Stavropol'skoe izdatel'stvo 'Paragraf'", pp. 215–219. (In Russian)
4. **Koklyanov A. E.** Application of SIEM systems during exercises on the cyber range platform. *Matematicheskoe modelirovanie, komp'yuternyj i naturnyj ehksperiment v estestvennyh naukah*. 2025. No. 1, pp. 45–51. DOI: 10.24412/2541-9269-2025-1-45-51. (In Russian)
5. **Listratov I. S., Miloslavskaya N. G., Sirbay I. S., Reinoso B. A.** Extended Cybereason's SOC maturity model. *Bezopasnost informacionnyh tehnology*. 2025. Vol. 32. No. 1, pp. 68–84. (In Russian)
6. **Kuznetsov A. V.** The organization of separate security event data storage. *Voprosy kiberbezopasnosti*. 2024. Vol. 60. No. 2, pp. 22–28. (In Russian)
7. **Pakhomov V. V.** Evaluating the effectiveness of cyber-threat monitoring and response centers: limits of time-based metrics and operational quality indicators. *Modeling, Optimization and Information Technology*. 2025. Vol. 13. No. 4. DOI: 10.26102/2310-6018/2025.51.4.040. (In Russian)
8. **Dolgachev M. V., Kostyunin V. A.** Comprehensive analysis of windows system behavior for cyber threat detection. *Voprosy kiberbezopasnosti*. 2025. Vol. 66. No. 2, pp. 71–77. (In Russian)
9. **Danilova O. T., Abeldmov R. M.** Practical application of SIEM-model based on elastic stackplatform for monitoring information security system. *Dinamika sistem, mehanizmov i mashin*. 2024. Vol. 12. No. 1, pp. 105–113. DOI: 10.25206/2310-9793-2024-12-1-105-113. (In Russian)
10. **Putyato M. M., Makaryan A. S., Cherkasov A. N., Kucher V. A.** Estimation of siem-systems functioning on the basis of set of effectiveness criteria. *The Bulletin of the Adyghe State University. Ser.: Natural-Mathematical and Technical Sciences*. 2024. No. 1, pp. 36–42. DOI: 10.53598/2410-3225-2024-1-336-36-42. (In Russian)

11. Gartner Magic Quadrant for Security Information and Event Management. Gartner Research. URL: https://www.gartner.com/en/documents/7040298 (accessed: 10.03.2026).
12. **Mashkina I. V., Urazaeva A. M.** Method of development of threat scenarios knowledge base for incident response platform (IRP). *Izvestia SFedU. Engineering Sciences*. 2024. No. 5, pp. 79–88. (In Russian)
13. **Shevtsova T. A.** Anomaly detection in complex data using clustering. *Professional Bulletin: Information Technology and Security*. 2024. No. 4, pp. 39–46. (In Russian)
14. **Yang Chen, Yijia Ma, Wei Wu**. Rank-Based Mixture Models for Temporal Point Processes. *Front. Appl. Math. Stat*. 2022. Vol. 8. DOI: 10.3389/fams.2022.852314.
15. **Kechedzhiev A. S., Tsvetkova O. L.** Anomaly detection research using Isolation Forest in Machine Learning. *Herald of Dagestan State Technical University. Technical Sciences*. 2024. Vol. 51. No. 1, pp. 106–112. DOI: 10.21822/2073-6185-2024-51-1-106-112. (In Russian)
16. **Dorofeev V. S., Volosatova T. M.** Ensemble methods for detecting outliers in the preparation of a training data set. *Modeling, Optimization and Information Technology*. 2022. Vol. 10. No. 3, pp. 1–13. DOI: 10.26102/2310-6018/2022.38.3.013. (In Russian)
17. Timestamp formats in logs. *new relic*. URL: https://docs.newrelic.com/docs/logs/ui-data/timestamp-support/ (accessed: 03.03.2026).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

МИХАНЬКО Антон Дмитриевич – аспирант, Уфимский университет науки и технологий, Россия, Республика Башкортостан, 450076, Уфа, ул. Заки Валиди, д. 32
E-mail: mikhanko45@gmail.com
ORCID: 0009-0007-7389-0429

MIKHANKO Anton D. – Postgraduate Student, Ufa University of Science and Technology, Russia, Republic of Bashkortostan, 450076, Ufa, Zaki Validi str., 32

МАШКИНА Ирина Владимировна – д-р техн. наук, профессор, Уфимский университет науки и технологий, Россия, Республика Башкортостан, 450076, Уфа, ул. Заки Валиди, д. 32
E-mail: profmashkina@mail.ru
ORCID: 0009-0003-2546-4354

MASHKINA Irina V. – Doctor of Engineering Sciences, Professor, Ufa University of Science and Technology, Russia, Republic of Bashkortostan, 450076, Ufa, Zaki Validi str., 32

Научная статья
DOI 10.66424/2071-8217-2026-2-4
УДК 004.056

ОБНАРУЖЕНИЕ АНОМАЛИЙ В СОБЫТИЯХ БЕЗОПАСНОСТИ ОС НА ОСНОВЕ СТАТИСТИЧЕСКОГО АНАЛИЗА И БОЛЬШИХ ЯЗЫКОВЫХ МОДЕЛЕЙ

А. В. Немчинов[1], Т. Д. Овасапян[2*], Е. В. Жуковский[2]

[1]Санкт-Петербургский государственный университет телекоммуникаций им. профессора М. А. Бонч-Бруевича, Санкт-Петербург, Россия

[2]Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *otd@ibks.spbstu.ru*

ДЛЯ ЦИТИРОВАНИЯ

Немчинов А. В., Овасапян Т. Д., Жуковский Е. В. Обнаружение аномалий в событиях безопасности ОС на основе статистического анализа и больших языковых моделей // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 49–59.
DOI: 10.66424/2071-8217-2026-2-4

ПОСТУПИЛА 27.04.2026
ПРИНЯТА 06.05.2026
ОПУБЛИКОВАНА 15.06.2026

АННОТАЦИЯ

Исследованы возможности применения больших языковых моделей и статистического анализа для автоматизации обнаружения аномалий в событиях безопасности ОС. Предложен метод обнаружения аномалий, позволяющий автоматически выделять значимые отклонения и формировать их интерпретацию. Разработан программный прототип, реализующий данный метод, и проведено его тестирование.

КЛЮЧЕВЫЕ СЛОВА

Статистический анализ, большие языковые модели, обнаружение аномалий

Original article
DOI 10.66424/2071-8217-2026-2-4

DETECTING ANOMALIES IN SECURITY EVENTS BASED ON STATISTICAL ANALYSIS AND LARGE LANGUAGE MODELS

A. V. Nemchinov[1], T. D. Ovasapyan[2*], E. V. Zhukovsky[2]

[1]St. Petersburg State University of Telecommunication Named After Professor M. A. Bonch-Bruevich, St. Petersburg, Russia

[2]Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *otd@ibks.spbstu.ru*

FOR CITATION

Nemchinov A. V., Ovasapyan T. D., Zhukovsky E. V. Detecting anomalies

ABSTRACT

The possibilities of using large language models and statistical methods to automate the detection of anomalies in OS security events

in security events based on statistical analysis and large language models. *Problems of information security. Computer systems*. 2026. No. 2, pp. 49–59. DOI: 10.66424/2071-8217-2026-2-4 (In Russian)

RECEIVED 27.04.2026
ACCEPTED 06.05.2026
PUBLICATION 15.06.2026

are investigated. A method for detecting anomalies is proposed that allows to automatically identify significant deviations and form their interpretation. A software prototype implementing this method has been developed and tested.

KEYWORDS

Statistical analysis, large language models, anomaly detection

1. ВВЕДЕНИЕ

Каждый компонент инфраструктуры генерирует события безопасности: журналы доступа, сообщения от средств защиты, системные логи. Security Information and Event Management (SIEM) – системы сбора и корреляции событий безопасности ежедневно обрабатывают миллионы таких событий. Каждое из них потенциально может свидетельствовать об инциденте безопасности. Однако большинство событий являются следствием легитимной работы системы или ложными срабатываниями [1].

Ручной анализ таких больших объемов данных требует значительных трудозатрат и высокой квалификации специалиста. Даже при использовании стандартных правил корреляции оператор получает множество предупреждений. Возникает потребность в автоматизированных средствах, которые могли бы не только выделять подозрительные отклонения в потоке событий, но и представлять их в удобной для восприятия форме, снижая нагрузку на специалиста [1, 2].

Активно развиваются методы обнаружения аномалий на основе статистического анализа [3, 4]. Они применяются для отсеивания фоновой активности и выделения статистически значимых отклонений. Параллельно с этим большие языковые модели демонстрируют хорошие результаты в понимании и генерации текста [5]. Их применение открывает возможности для автоматической интерпретации выявленных отклонений и формирования понятных оператору сообщений. Однако прямое использование LLM для анализа всего потока событий нецелесообразно из-за высоких вычислительных затрат и риска потери значимых сигналов в шуме [1].

Таким образом, актуальной задачей является разработка комбинированного подхода, сочетающего статистическую фильтрацию для выделения ограниченного числа значимых аномалий и последующую их обработку с помощью LLM. Это позволит снизить нагрузку на оператора, предоставляя ему только действительно важные отклонения с понятными пояснениями.

2. СТАТИСТИЧЕСКИЕ МЕТОДЫ ОБНАРУЖЕНИЯ АНОМАЛИЙ

Задача обнаружения аномалий заключается в выявлении ключевых данных, событий или наблюдений, которые значительно отклоняются от нормального набора данных [6, 7]. В контексте анализа событий безопасности под аномалией понимается резкое изменение интенсивности событий определенного типа или нарушение статистических закономерностей.

Поскольку события безопасности регистрируются во времени, их удобно представлять в виде временных рядов – последовательностей значений, упорядоченных в порядке времени. Далее будут рассмотрены основные из применяемых статистических подходов, основанных на анализе этих временных рядов.

Пороговые методы. Самым простым является установление фиксированных порогов. Если значение некоторой метрики, например, количество событий в минуту, превышает заданный порог T, фиксируется аномалия. Пороги могут быть

относительными или абсолютными. Абсолютные пороги задаются в виде фиксированных числовых значений и не зависят от контекста функционирования системы. Относительные пороги, наоборот, вычисляются на основе нормального поведения системы, например, как отклонение от среднего значения за определенный период.

Достоинством метода является простота реализации и минимальные вычислительные затраты. Однако фиксированные пороги не адаптируются к изменениям в поведении системы, требуют ручной настройки и могут приводить к большому числу ложных срабатываний.

Методы скользящего среднего (SMA и EWMA). Более гибкими являются методы, использующие скользящие статистики [8, 9]. Их суть в построении сглаженной оценки текущего уровня ряда на основе недавних наблюдений. Эта оценка будет прогнозом ожидаемого значения, а отклонение фактического наблюдения от прогноза может свидетельствовать об аномалии. Простое скользящее среднее (SMA) вычисляется как среднее арифметическое последних k наблюдений:

$$\bar{x}_t = \frac{1}{k} \sum_{i=t-k+1}^{t} x_i,$$

где x_i – количество событий в i-й период времени; x_t – количество событий за фиксированный период времени t; k – количество последовательных наблюдений.

Аномалия формируется при $\left|x_t - \bar{x}_{t-}\right| > \delta$. Параметр δ выбирается эмпирически, например, как удвоенное стандартное отклонение значений внутри окна.

Экспоненциально взвешенное скользящее среднее (EWMA) придает больший вес свежим данным:

$$\hat{x}_t = ax_t + (1-a)\hat{x}_{t-1},$$

где $a \in (0,1)$ – параметр сглаживания. Чем ближе a к единице, тем быстрее модель реагирует на новые наблюдения.

Отклонение оценивается по правилу:

$$\left|x_t - \hat{x}_{t-1}\right| > \beta\sigma,$$

где σ – оценка стандартного отклонения процесса; β – коэффициент.

Если неравенство выполняется, наблюдение помечается как аномальное. Метод EWMA хорошо реагирует на резкие изменения, однако не учитывает сезонные колебания – регулярные, повторяющиеся изменения метрики во времени. Например, в выходные и праздничные дни интенсивность событий снижается, тогда как в рабочие дни она выше. В результате такие закономерные изменения могут ошибочно интерпретироваться как аномалии.

Метод кумулятивных сумм (CUSUM). Метод кумулятивных сумм предназначен для обнаружения небольших постоянных смещений среднего [10]. Накопленные отклонения вверх и вниз рассчитываются по формулам:

$$S_t^+ = \max\left(0, S_{t-1}^+ + x_t - \mu_0 - k\right),$$

$$S_t^- = \max\left(0, S_{t-1}^- - x_t + \mu_0 - k\right),$$

где μ_0 – целевое среднее (ожидаемое значение при отсутствии нарушений); k – допустимое отклонение.

Аномалия фиксируется при превышении порога h $\left(S_t^+ > h,\ S_t^- > h\right)$. CUSUM эффективен для обнаружения длительного повышения количества событий, однако чувствителен к сезонным колебаниям, если они не учтены при выборе μ_0.

Робастные статистики. Робастные статистики устойчивы к наличию выбросов, в отличие от рассматриваемых [7, 11]. Наиболее распространенные робастные статистики – медиана и медианное абсолютное отклонение (MAD). Медиана M – центральное значение упорядоченной выборки, половина значений меньше, половина больше. MAD определяется как медиана абсолютных отклонений наблюдений от медианы:

$$M = median\left\{x_1, x_{2,\ldots}, x_n\right\},$$

$$\mathrm{MAD} = median\left\{\left|x_i - M\right|\right\}.$$

На основе медианы и MAD строится модифицированная Z-оценка. Она означает число стандартных отклонений, на которое значение наблюдаемой величины отклоняется от медианы [6]:

$$Z = \frac{|x_t - M|}{1{,}4826\,\mathrm{MAD}},$$

где x_t – количество событий за фиксированный период времени t.

Коэффициент 1,4826 возникает из отношения между MAD и стандартным отклонением для нормального распределения [12]. Для нормально распределенной случайной величины справедливо равенство:

$$\mathrm{MAD} \approx 0{,}6745\sigma.$$

Обратное соотношение позволяет выразить стандартное отклонение через MAD:

$$\sigma \approx \frac{\mathrm{MAD}}{0{,}6745} \approx 1{,}4826\,\mathrm{MAD}.$$

Таким образом, коэффициент 1,4826 приводит MAD к масштабу стандартного отклонения в предположении нормального распределения. Это позволяет интерпретировать Z так же, как классическую Z-оценку. Значения $Z > 3$ обычно считаются признаком аномалии.

Достоинства данного подхода заключаются в устойчивости к единичным выбросам в данных, отсутствии предположений о распределении, в простоте вычисления и наглядности получаемых результатов.

Каждый метод имеет свои сильные и слабые стороны. Для использования выбран метод робастных статистик и Z-оценки, так как имеет наибольшую устойчивость к выбросам при сохранении интерпретируемости и низкой вычислительной сложности.

3. ОПИСАНИЕ ПРЕДЛАГАЕМОГО МЕТОДА ОБНАРУЖЕНИЯ АНОМАЛИЙ

Исходными данными для работы метода будут события безопасности, поступающие от различных источников инфраструктуры. В SIEM-системе они агрегируются по двум ключевым признакам: источнику событий и типу событий. Для каждой такой пары ведется учет количества срабатываний за фиксированные временные интервалы. Получаемые временные ряды сохраняются и будут использоваться для построения профиля нормального поведения [9]. Общая схема этого процесса показана на рис. 1.

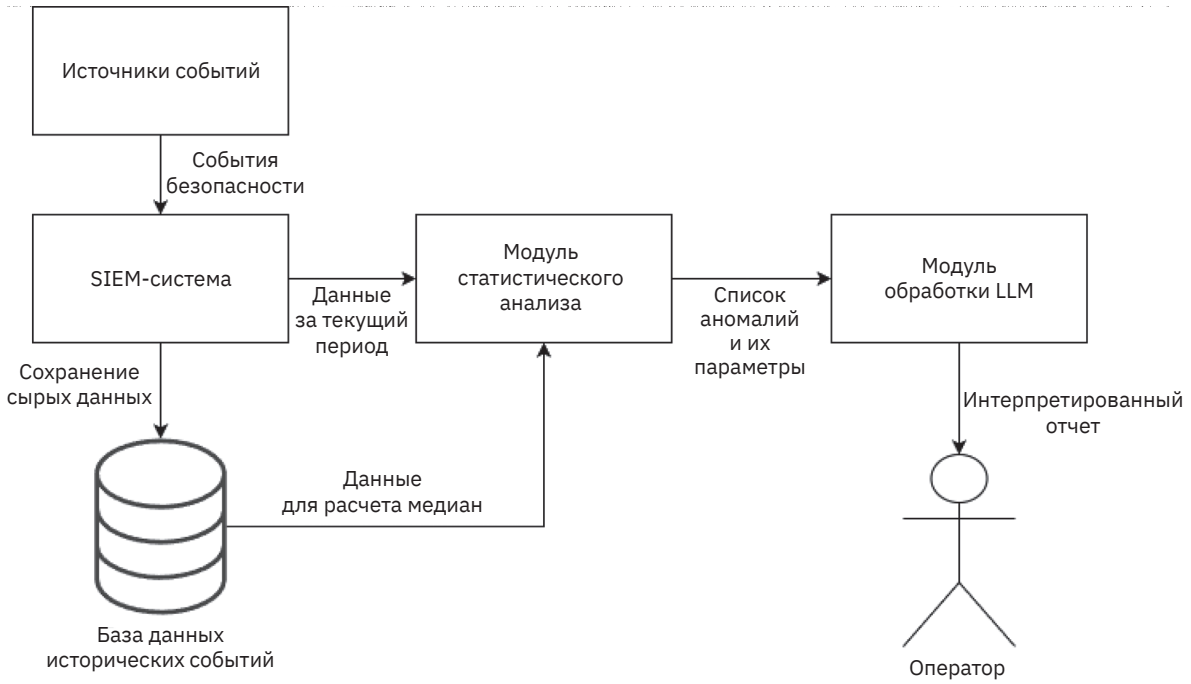


Рис. 1 | Общая схема работы метода обнаружения аномалий

Fig. 1 | General scheme of operation of the anomaly detection method

В качестве временного интервала выбраны сутки – такой промежуток будет самым оптимальным для анализа из-за периодичности активности в инфраструктуре. Для учета сезонных колебаний все дни разделяются на две категории: рабочие и выходные, так как характер и количество событий в эти дни существенно отличается [3, 9].

Для каждого источника и типа событий отдельно для рабочих и выходных дней вычисляются медиана и медианное абсолютное отклонение. Медиана M отражает типичное количество событий данного типа в сутки, а MAD характеризует естественный разброс вокруг медианны. Эти показатели станут стандартом, на который будут ссылаться показания в новый день.

При обработке очередных суточных данных для каждого текущего значения x_t его отклонение от «стандартных» значений. На основе ранее вычисленных переменных вычисляется модифицированная Z-оценка, она показывает, насколько удалено текущее значение от типичного. Однако при малых значениях медианны даже небольшое абсолютное изменение может давать высокую Z-оценку, что приведет к ложным срабатываниям. Чтобы избежать этого, решение о том является ли наблюдение аномалией принимается на основе нескольких критериев.

Первый критерий – абсолютное отклонение $\Delta = |x_t - M|$. Оно должно превышать некоторое минимальное значение $\Delta_{\min}$ – это отсеивает случайные отклонения от медианны.

Второй критерий – относительное отклонение в процентах $\delta = \frac{\Delta}{M} \cdot 100\,\%$, при $M > 0$. Если медиана равна нулю, то δ принимается равным 100 % при наличии событий. Также применяется порог $\delta_{\min}$, он зависит от величины медианны. Для малых медиан порог устанавливается выше, около 150 %, для больших и средних примерно 40 и 80 % соответственно.

Третий критерий – модифицированная Z-оценка, для которой тоже устанавливается порог $Z_{\min}$. При малых медианах $Z_{\min} = 3$, при средних – 4, при больших – 5. Это обеспечивает адаптивность к разным масштабам данных и позволяет избежать ложных срабатываний из-за естественной активности.

Наблюдение фиксируется как аномалия только при выполнении перечисленных условий. Количество аномалий может быть велико даже при таком отсеивании статистического шума. Поэтому аномалии также ранжируются по уровню критичности, который определяется весовым коэффициентом. Он учитывает величину Z-оценки, относительное отклонение и уровень критичности событий, присвоенной SIEM-системой. Например, правила с уровнем критичности 10 и выше получают повышающий множитель. На рис. 2 приведен алгоритм статистического анализа выявления аномалий.

Отобранные аномалии структурируются и передаются большой языковой модели. Языковая модель способна оценить семантический контекст происходящего [1, 2] и анализирует совокупность передаваемых параметров: идентификатор источника, тип события, текущее количество, медиану, величину отклонения и критичность события. На их основе языковая модель формирует список аномалий, которые с наибольшей вероятностью связаны с инцидентами безопасности.

4. РЕАЛИЗАЦИЯ ПРОГРАММНОГО КОМПЛЕКСА И ОЦЕНКА ЭФФЕКТИВНОСТИ

Разработанный программный комплекс реализован на языке Python и интегрирован с SIEM-системой Wazuh. Он состоит из трех модулей, взаимодействующих через файловую систему и API Telegram.

1. **Сбор и агрегация.** Ежедневно запускается по расписанию. Выполняет аутентификацию в Wazuh Indexer и выгружает события за последние сутки с помощью scroll-запросов. Это позволяет обрабатывать большие объемы событий.

Все сырые события сохраняются в сжатом формате gzip в каталоге raw_events,

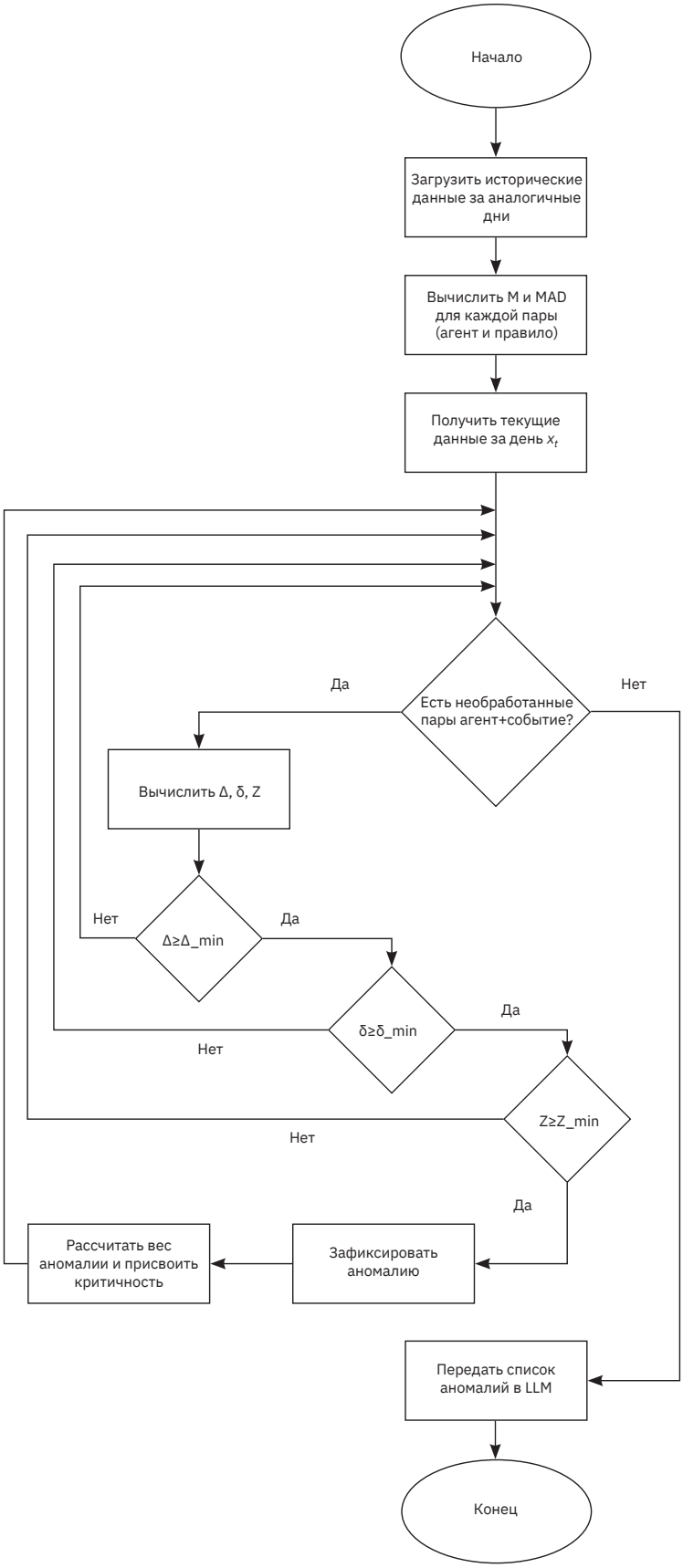


Рис. 2 | Алгоритм статистического выявления аномалий

Fig. 2 | An algorithm for statistical detection of anomalies

что дает возможность при необходимости пересчитать статистику или изменить алгоритм агрегации без повторного обращения к Wazuh Indexer. Также формируется агрегированная статистика для каждого агента и правила, подсчитывается количество срабатываний. Результат записывается в json-файл в каталоге stats_events.

2. **Обнаружение аномалий.** Реализуется метод обнаружения аномалий. На вход принимает статистику за прошедшие сутки. Сначала определяется тип дня – рабочий или выходной. Затем выбираются аналогичные предшествующие дни, количество определяется по параметру HISTORY_DAYS, по умолчанию пять дней. Для каждого дня загружается соответствующий статистический файл. Для каждой пары «агент – правило» формируется временной ряд: по выбранным предыдущим дням собираются значения количества срабатываний соответствующего правила. На основе этого ряда вычисляются медиана M и медианное абсолютное отклонение MAD. Если срабатывание правила у агента встречалось менее чем в двух днях, то MAD принимается равным нулю.

Далее загружается статистика текущего дня и последовательно проверяются все записи. Для каждой записи вычисляются абсолютное отклонение $\Delta = |x_t - M|$, относительное отклонение $\delta = \frac{\Delta}{M} \cdot 100\ \%$, модифицированная Z-оценка $Z = \frac{|x_t - M|}{1{,}4826\,\mathrm{MAD}}$. Запись признается аномалией, если одновременно все перечисленные параметры проходят пороги (табл. 1).

Каждой аномалии присваивается вес:

$$w = Z + \frac{\delta}{100} f,$$

где f – коэффициент важности правила; Z – модифицированная Z-оценка; δ –относительное отклонение.

Коэффициент важности определяется так: 2 для уровня правила ≥ 10; 1,5 для уровня 7–9; 1 для уровня меньше 7. Это позволяет выделить наиболее важные отклонения одновременно по уровню опасности события и по величине отклонения. Аномалии сортируются по убыванию веса, и в итоговый список попадают только самые весомые.

Все пороговые значения задаются через переменные окружения, это позволяет гибко настраивать алгоритм под конкретную инфраструктуру. Полный список настраиваемых параметров приведен в табл. 2.

Результатом работы второго модуля является json-файл today_stat. Он содержит дату, общее число найденных аномалий и детальную информацию про каждую. Этот файл сохраняется и одновременно копируется в отдельную директорию для передачи третьему модулю.

Третий модуль – интеграция с LLM и оповещение. Формируется подробное сообщение с найденными аномалиями для оператора на основании файла today_stat. Аномалиям присваивается уровень опасности исходя из рассчитанного веса: критический при $w \geq 20$, высокий при $10 \leq w < 20$, средний при $5 \leq w < 10$ и низкие при $w < 5$. Для каждой аномалии пишется имя агента, его ip и id в Wazuh, номер правила, уровень опасности,

Таблица 1 | Адаптивные пороги в зависимости от медианы

Table 1 | Adaptive thresholds depending on the median

Диапазон медиан	Порог абсолютного отклонения Δ_{min}	Порог относительного отклонения δ_{min},%	Порог Z-оценки z_{min}
$M < 20$	10	150	3,0
$20 \leq M < 50$		80	4,0
$50 \leq M < 200$		40	5,0
$M \geq 200$		Не используется	6,0

Таблица 2 | Основные параметры конфигурации модуля обнаружения аномалий

Table 2 | The main configuration parameters of the anomaly detection module

Переменная	Значение по умолчанию	Описание
HISTORY_DAYS	5	Количество похожих дней
MIN_ABSOLUTE_DIFF	10	Минимальная абсолютное отклонение
MIN_PERCENT_DIFF_SMALL	150	Порог относительного отклонения при медиане <20
MIN_PERCENT_DIFF_SMALL_PLUS	80	Порог относительного отклонения при медиане 20–50
MIN_PERCENT_DIFF_MEDIUM	40	Порог относительного отклонения при медиане 50–200
Z_SCORE_THRESHOLD_SMALL	3,0	Порог *Z* при медиане <20
Z_SCORE_THRESHOLD_SMALL_PLUS	4,0	Порог *Z* при медиане 20–50
Z_SCORE_THRESHOLD_MEDIUM	5,0	Порог *Z* при медиане 50–200
Z_SCORE_THRESHOLD_LARGE	6,0	Порог *Z* при медиане >200
MAX_ANOMALIES_PER_AGENT	5	Максимум аномалий для агента
MAX_TOTAL_ANOMALIES	50	Общее число аномалий в сообщении

текущее количество событий, вес и относительное отклонение. Сформированное сообщение отправляется оператору.

Далее содержимое файла today_stat передается в большую языковую модель DeepSeek с использованием API. Для наиболее точного ответа используется промпт, который предписывает модели выбирать аномалии на основе числовых показателей и семантики событий, переводить описание событий на русский язык и формировать краткое сообщение. Это сообщение также отправляется оператору, как краткая выжимка наиболее опасных аномалий.

Помимо модуля обнаружения аномалий реализован модуль ежедневной статистики. Он формирует общую сводку по состоянию SIEM-системы. В сводку включаются количество активных и отключившихся агентов, топ наиболее часто встречающихся событий за сутки, число исправленных и новых уязвимостей на основе данных сканера уязвимостей Wazuh. На основе данных за последние 10 дней строятся два графика, показывающих изменение количества событий и уязвимостей, а также их распределение по уровням критичности. Пример графиков можно увидеть на рис. 3.

Тестирование разработанного комплекса проводилось в реальной инфраструктуре, включающей 60 активных агентов SIEM. Ежедневно в Wazuh регистрировалось около миллиона событий. Модуль статистического обнаружения аномалий выявлял в среднем 14 аномалий в сутки. После обработки результатов большой языковой моделью сообщение для оператора содержало информацию о 2–3 наиболее значимых аномалиях. Все события, которые впоследствии были квалифицированы как инциденты безопасности, вошли в число отобранных LLM. Это свидетельствует об эффективности данного подхода и разработанного программного комплекса.

5. ЗАКЛЮЧЕНИЕ

В результате выполнения работы исследованы методы статистического обна-

ружения аномалий в событиях безопасности, проведен их анализ. Обоснован выбор робастных статистик в сочетании с модифицированной *Z*-оценкой как наиболее устойчивого к выбросам и интерпретируемого подхода. Разработан метод обнаружения аномалий, учитывающий сезонные колебания активности и использующий адаптивные пороги. Предложен весовой коэффициент, позволяющий ранжировать аномалии с учетом всех числовых величин.

На основе предложенного метода реализован программный комплекс, интегрированный с Wazuh и Telegram. Комплекс состоит из трех модулей: сбора и агрегации событий, обнаружения аномалий с помощью статистических методов,

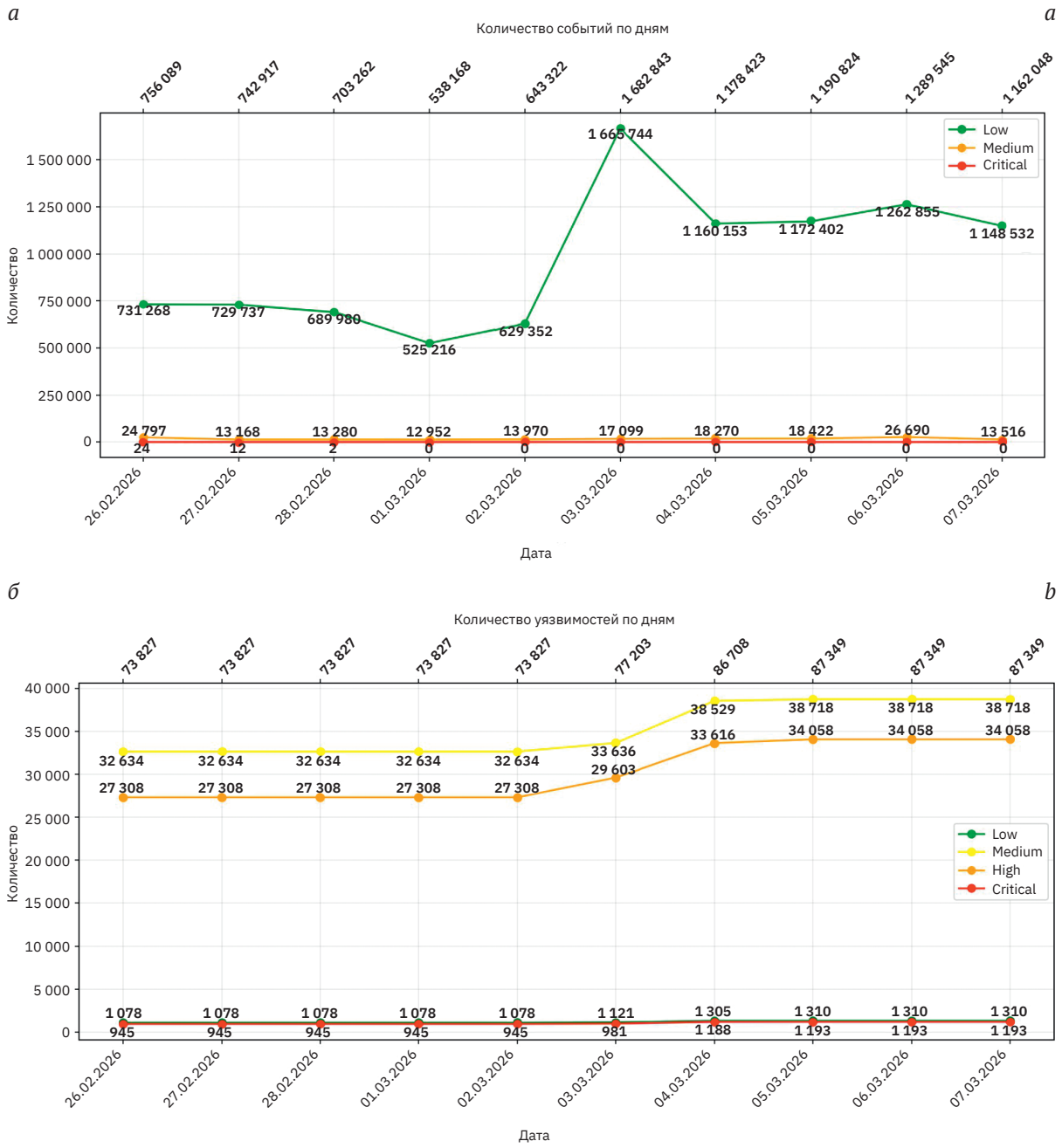


Рис. 3 | График распределения событий (*а*) и уязвимостей (*б*) по 10 последним дням

Fig. 3 | Schedule of distribution of events (*a*) and vulnerabilities (*b*) for the last 10 days

а также модуля интеграции с большой языковой моделью DeepSeek для фильтрации и интерпретации выявленных аномалий. Разработан дополнительный модуль ежедневной статистики, помогающий отслеживать активность в SIEM, предоставляя оператору контекстную информацию.

Экспериментальная проверка проведена в условиях реальной эксплуатации и подтвердила работоспособность и эффективность предложенного подхода. Применение LLM позволило значительно сократить число оповещений и расширить понимание аномального поведения. При этом сохраняется полнота выявленных инцидентов. Дальнейшее развитие комплекса предполагает автоматическое обновление эталонных значений медиан и MAD по мере накопления новых данных, учет многомерных корреляций между событиями, а также расширение функциональности языковой модели для написания рекомендаций по реагированию на выявленные аномалии.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Chagna Y., Goldschmidt A.** Next-generation cyberattack detection with large language models: anomaly analysis across heterogeneous logs // arXiv preprint arXiv:2602.06777. 2026.
2. **da Silva G. de J. C., Westphall C. B.** A survey of large language models in cybersecurity // arXiv preprint arXiv:2402.16968. 2024.
3. **Hochenbaum J., Vallis O. S., Kejariwal A.** Automatic anomaly detection in the cloud via statistical learning // arXiv preprint arXiv: 1704.07706. 2017.
4. **Stetsyuk M., Anikin V., Pyrch O. et al.** Method of detecting anomalies in IoT device traffic based on statistical analysis using the modified Z score // CEUR Workshop Proceedings. 2025. Vol. 3963. P. 284–298.
5. **Minaee S., Mikolov T., Nikzad N. et al.** Large language models: A Survey // arXiv preprint arXiv:2402.06196. 2024.
6. **Стельмах Н. Е., Козачков А. В.** Обзор методов выявления аномалий при аудите системных вызовов в ОС // International Journal of Open Information Technologies. 2025. Т. 13. № 9. С. 25–33.
7. **Iglewicz B., Hoaglin D. C.** How to detect and handle outliers. Milwaukee: ASQC Quality Press, 1993. 87 p.
8. **Roberts S. W.** Control chart tests based on geometric moving averages // Technometrics. 1959. Vol. 1. № 3. P. 239–250.
9. **Vallis O. S., Hochenbaum J., Kejariwal A.** A novel technique for long-term anomaly detection in the cloud // 6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14). 2014.
10. **Page E. S.** Continuous inspection schemes // Biometrika. 1954. Vol. 41. № 1–2. P. 100–115.
11. **Leys C., Ley C., Klein O. et al.** Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median // Journal of Experimental Social Psychology. 2013. Vol. 49. № 4. P. 764–766.
12. **Rousseeuw P. J., Croux C.** Alternatives to the median absolute deviation // Journal of the American Statistical Association. 1993. Vol. 88. № 424. P. 1273–1283.

REFERENCES

1. **Chagna Y., Goldschmidt A.** Next-generation cyberattack detection with large language models: anomaly analysis across heterogeneous logs. *arXiv preprint arXiv:2602.06777*. 2026.

2. **da Silva G. de J. C., Westphall C. B.** A survey of large language models in cybersecurity. *arXiv preprint arXiv:2402.16968*. 2024.
3. **Hochenbaum J., Vallis O. S., Kejariwal A.** Automatic anomaly detection in the cloud via statistical learning. *arXiv preprint arXiv: 1704.07706*. 2017.
4. **Stetsyuk M., Anikin V., Pyrch O. et al.** Method of detecting anomalies in IoT device traffic based on statistical analysis using the modified Z score. *CEUR Workshop Proceedings*. 2025. Vol. 3963, pp. 284–298.
5. **Minaee S., Mikolov T., Nikzad N. et al.** Large language models: A Survey. *arXiv preprint arXiv:2402.06196*. 2024.
6. **Stelmach N. E., Kozachok A. V.** Review of anomaly detection methods during system call auditing in OS. *International Journal of Open Information Technologies*. 2025. Vol. 13. No. 9, pp. 25–33. (In Russian)
7. **Iglewicz B., Hoaglin D. C.** How to detect and handle outliers. Milwaukee: ASQC Quality Press, 1993, 87 p.
8. **Roberts S. W.** Control chart tests based on geometric moving averages. *Technometrics*. 1959. Vol. 1. No. 3, pp. 239–250.
9. **Vallis O. S., Hochenbaum J., Kejariwal A.** A novel technique for long-term anomaly detection in the cloud. 6th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 14). 2014.
10. **Page E. S.** Continuous inspection schemes. *Biometrika*. 1954. Vol. 41. No. 1–2, pp. 100–115.
11. **Leys C., Ley C., Klein O. et al.** Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*. 2013. Vol. 49. No. 4, pp. 764–766.
12. **Rousseeuw P. J., Croux C.** Alternatives to the median absolute deviation. *Journal of the American Statistical Association*. 1993. Vol. 88. No. 424, pp. 1273–1283.

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

НЕМЧИНОВ Александр Владимирович – студент, Санкт-Петербургский государственный университет телекоммуникаций им. профессора М. А. Бонч-Бруевича, Россия, 193232, Санкт-Петербург, пр-т Большевиков, д. 22, к. 1
E-mail: sasha01082004@gmail.com
ORCID: 0009-0001-3348-5038

NEMCHINOV Alexander V. – Student, St. Petersburg State University of Telecommunication Named After Professor M. A. Bonch-Bruevich, Russia, 193232, St. Petersburg, Bolshevikov ave., 22, bldg. 1

ОВАСАПЯН Тигран Джаникович – канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: otd@ibks.spbstu.ru
ORCID: 0000-0002-2009-5460

OVASAPYAN Tigran D. – Candidate of Engineering Sciences, Associate Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

ЖУКОВСКИЙ Евгений Владимирович – канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: ezhukovsky@ibks.spbstu.ru
ORCID: 0009-0006-9013-1750

ZHUKOVSKY Evgeny Vladamirovich. – Candidate of Engineering Sciences, Associate Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья
DOI 10.66424/2071-8217-2026-2-5
УДК 004.056

ПСИХОЛОГИЧЕСКИЕ ПОСЛЕДСТВИЯ РАБОТЫ С СИСТЕМАМИ SECURITY OPERATIONS CENTER (SOC): ВЫГОРАНИЕ, КОГНИТИВНАЯ НАГРУЗКА И РОЛЬ ИИ-АССИСТЕНТОВ

Е. В. Ларионова[1], И. Л. Бунас[1], А. Ю. Гарькушев[1], А. Ф. Супрун[2*]

[1]Санкт-Петербургский государственный морской технический университет, Санкт-Петербург, Россия
[2]Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия
✉ *afs54@inbox.ru*

ДЛЯ ЦИТИРОВАНИЯ

Ларионова Е. В., Бунас И. Л., Гарькушев А. Ю., Супрун А. Ф. Психологические последствия работы с системами Security Operations Center (SOC): выгорание, когнитивная нагрузка и роль ИИ-ассистентов // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 60–69.
DOI: 10.66424/2071-8217-2026-2-5

ПОСТУПИЛА 28.01.2026
ПРИНЯТА 08.05.2026
ОПУБЛИКОВАНА 15.06.2026

АННОТАЦИЯ

Рассматриваются психологические последствия работы аналитиков в Security Operations Center (SOC): хронический стресс, профессиональное выгорание и связанные с ними ошибки в обеспечении кибербезопасности. Показано, что выгорание следует рассматривать как операционный риск, влияющий на вероятность инцидентов и снижающий эффективность решений. Анализируются современные ИИ-решения для SOC (LLM-ассистенты, агентные системы, поведенческие модели) и их ограничения. Рассматриваются принципы мониторинга функционального состояния оператора и структуры анализа инцидентов, механизмы ограничения влияния ИИ в целях сохранения управляемости и ответственности. Показана практическая применимость предложенного подхода, где надежность принятия решений оператором поддерживается ИИ-контуром.

КЛЮЧЕВЫЕ СЛОВА

SOC, профессиональное выгорание, усталость от оповещений, когнитивная нагрузка, кибербезопасность, ИИ-ассистенты, человеческий фактор

Original article
DOI 10.66424/2071-8217-2026-2-5

PSYCHOLOGICAL EFFECTS OF WORK IN SECURITY OPERATIONS CENTER (SOC) SYSTEMS: BURNOUT, COGNITIVE LOAD, AND THE ROLE OF AI-ASSISTANTS

E. V. Larionova[1], I. L. Bunas[1], A. Yu. Garkushev[1], A. F. Suprun[2*]

[1]Saint-Petersburg State Marine Technical University, St. Petersburg, Russia
[2]Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia
✉ *afs54@inbox.ru*

FOR CITATION

Larionova E. V., Bunas I. L., Garkushev A. Yu., Suprun A. F. Psychological effects of work in Security Operations Center (SOC) systems: burnout, cognitive load, and the role of AI-assistants. *Problems of information security. Computer systems*. 2026. No. 2, pp. 60–69. DOI: 10.66424/2071-8217-2026-2-5 (In Russian)

RECEIVED 28.01.2026
ACCEPTED 08.05.2026
PUBLICATION 15.06.2026

ABSTRACT

The article examines the psychological consequences of analysts' work in a Security Operations Center (SOC), including chronic stress, professional burnout, and related cybersecurity errors. Burnout is considered as an operational risk that increases the likelihood of incidents and reduces decision effectiveness. Modern AI solutions for SOCs (LLM assistants, agent-based systems, and behavioral models) and their limitations are analyzed. The paper discusses monitoring of the operator's functional state, the structure of incident analysis, and mechanisms for limiting AI influence in order to preserve manageability and responsibility. The practical applicability of the proposed approach is demonstrated, where the reliability of operator decision-making is supported by an AI-assisted loop.

KEYWORDS

SOC, professional burnout, alert fatigue, cognitive load, cybersecurity, AI-assistants, human factors

1. ВВЕДЕНИЕ

Современный SOC (центр управления безопасностью) функционирует в условиях постоянной перегрузки: рост поверхности атак, гибридная инфраструктура, дефицит кадров и десятки тысяч оповещений в сутки. Для оператора центра безопасности это означает практически непрерывный режим многозадачности, дефицит времени на глубокий анализ и постоянное давление высокой цены ошибки.

По данным отраслевых опросов 83 % специалистов по информационной безопасности связывают ошибки, приведшие к инцидентам, со стрессом и профессиональным выгоранием. Аналогичные выводы содержатся и в обзорах по кибербезопасности в целом: значимая доля утечек и инцидентов обусловлена не только техническими отказами, но и влиянием человеческого фактора на фоне хронической усталости, напрямую отражающейся на качестве принимаемых решений [1–3].

На рынке параллельно присутствуют ИИ-ассистенты для SOC: от крупных решений уровня Microsoft Security Copilot и Google Sec-PaLM до агентных систем, автоматически собирающих контекст и формирующих черновики расследований. Публикуемые бенчмарки показывают рост скорости и полноты анализа инцидентов [4], однако в открытых описаниях таких решений основной акцент делается на ускорении triage, сокращении MTTR и автоматизации типовых операций. Вопрос, как подобные инструменты влияют на устойчивость принятия решений оператором в условиях неполных данных, пиковых нагрузок и длительного стресса, остается открытым.

В российских разработках также используются ML-модули поведенческого анализа для выявления аномалий и атак, а также интеграция результатов с механизмами SIEM. В описаниях таких решений заявляется снижение объема рутинных операций и нагрузки на аналитиков [5], однако влияние подобных подходов на надежность решений человека в контуре управления, как правило, не рассматривается системно. Роль человеческого фактора и когнитивных ограничений оператора в системах информационной безопасности рассматривается и в ряде отечественных работ [6–11]. Тем не менее вопрос о проектировании ИИ-ассистента как средства стабилизации процесса анализа, а не только ускорения обработки, практически не рассматривается.

В рамках данной работы предложено понятие «функциональное состояние оператора», под которым понимаются не клинические или эмоциональные характеристики, а наблюдаемые признаки дестабилизации процесса анализа в человеко-машинном контуре. К ним относятся: зацикливание рассуждения, потеря

связности, смешение контекстов задач, рост повторных перепроверок и снижение устойчивости движения к цели расследования.

Цель исследования – очертить психологические и операционные механизмы стресса в SOC, обозначить ограничения существующих ИИ-подходов и предложить концептуальный контур ИИ-ассистента, ориентированного на поддержание когнитивной стабильности операторов в условиях стрессовой и критической нагрузки, а не только на метрики MTTR и объем обработанных инцидентов.

Дополнительно обозначаются классы признаков, по которым может фиксироваться дестабилизация анализа, и базовые направления дальнейшей оценки эффективности подобных систем.

2. СНИЖЕНИЯ СТАБИЛЬНОСТИ РЕШЕНИЙ В УСЛОВИЯХ ХРОНИЧЕСКОЙ КОГНИТИВНОЙ НАГРУЗКИ

Факторы нагрузки. Рабочая среда SOC комбинирует несколько типов давления (рис. 1). Кратко рассмотрим их.

Постоянная мультизадачность. Оператор одновременно ведет несколько рас-

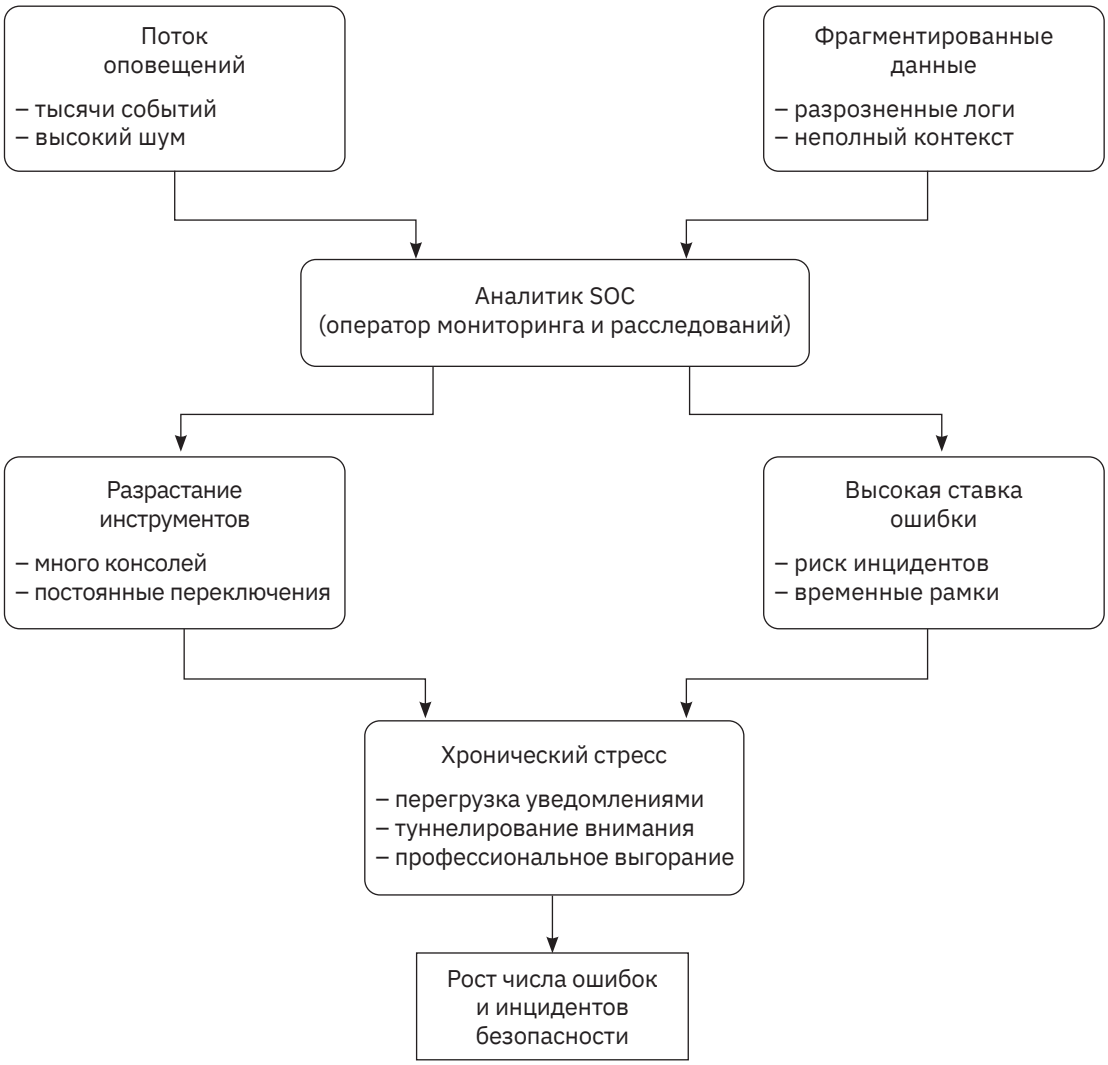


Рис. 1 | Основные источники когнитивной нагрузки в SOC

Fig. 1 | The main sources of cognitive load in SOC

следований, отслеживает очередь оповещений и дежурные каналы связи, а также вынужден регулярно переключаться между задачами. Каждое такое переключение требует концентрации внимания, расходует ресурсы рабочей памяти и снижает качество принимаемых решений.

Фрагментированные данные. Инцидент собирается из разрозненных источников отчетов, событий, внешних источников сведений об угрозах, записей обращений. Необходимость держать в голове несогласованные фрагменты усиливает нагрузку и делает анализ угрозы более уязвимым к шаблонному реагированию и риску пропустить значимые данные.

Высокая цена ошибки. Ошибочная классификация оповещения может привести к утечке, простоям и репутационным потерям, что формирует устойчивый стрессовый фон.

«Разрастание инструментов». Постоянное переключение между многочисленными консолями SIEM, EDR (обнаружение и реагирование на конечных устройствах), TI-порталами (централизованная точка доступа к сведениям о киберугрозах) и системами учета приводит к феномену контекстной усталости, требованию постоянных перепроверок и увеличению механических ошибок.

Усталость от оповещений и профессиональное выгорание. Усталость от оповещений описывается как состояние десенситизации: при тысячах оповещений в день, значительная часть которых может оказывается ложной тревогой, оператор перестает своевременно реагировать даже на действительно важные сигналы [3–5, 12]. На практике это обычно выражается в увеличении времени реакции на критические события, росте числа пропущенных инцидентов, субъективном ощущении нехватки времени у оператора.

Хроническая перегрузка приводит к профессиональному выгоранию: эмоциональному истощению, деперсонализации и снижению чувства профессиональной компетентности. Это повышает риск ошибок, усиливает текучесть кадров и разрушает накопленную экспертизу команды [1, 2, 13, 14].

Таким образом, функциональное и психологическое состояние аналитиков прямо влияет на качество принимаемых решений и устойчивость SOC как системы и рекомендуется рассматривать как отдельный класс операционных рисков.

В этом контексте задача ИИ-ассистента состоит не в диагностике психологического состояния оператора, а в выявлении признаков дестабилизации процесса анализа и поддержании надежности решений в человеко-машинном контуре.

3. СОВРЕМЕННЫЕ ИИ-РЕШЕНИЯ ДЛЯ SOC: ВОЗМОЖНОСТИ И ОГРАНИЧЕНИЯ

Если абстрагироваться от маркетинговых формулировок, ИИ-решения для SOC можно условно разделить на три группы [4, 15]:

1. LLM-ассистенты. Генеративные модели, встроенные в экосистемы SIEM и SOAR (оркестрация и автоматизация реагирования): резюмируют инциденты, помогают формировать запросы, предлагают шаги реагирования.

2. Агентные системы. Автоматически собирают контекст, запускают типовые проверки, формируют черновой таймлайн инцидента и рекомендации по реагированию.

3. Модули поведенческого анализа и приоритизации. Используют ML/Big Data для выявления аномалий и оценки риска активов, снижая шум и перераспределяя внимание аналитиков на более критичные события.

Эти системы хорошо решают задачи снятия рутины, ускорения триажа и улучшения документирования инцидентов. При этом в открытых описаниях практически не освещаются несколько критически важных аспектов [4, 6]:

1. Отсутствие работы с состоянием анализа. Метрики фокусируются на MTTR и количестве обработанных кейсов. При этом влияние ИИ на выгорание и когнитивную нагрузку не измеряется систематически. Более корректной постановкой

задачи представляется не прямая «диагностика состояния оператора», а фиксация признаков дестабилизации процесса анализа в человеко-машинном контуре. К таким признакам могут относиться зацикливание рассуждения, потеря связности, смешение контекстов задач и рост повторных перепроверок без прироста результата.

2. Смещение автоматизации и деквалификация. Чем «умнее» ведет себя ассистент, тем выше риск, что уставшие аналитики будут следовать его рекомендациям без критической проверки. Одновременно сокращается поле для практики глубокого анализа, что ведет к постепенному обеднению навыков. В этом смысле ИИ может ускорять обработку инцидентов без повышения надежности решений, особенно в условиях неполных и противоречивых данных.

3. Техностресс и смещение ответственности. Формально человек остается в контуре принятия решений, фактически же значительная часть логики расследования переходит к ИИ, тогда как ответственность за ошибки сохраняется за оператором. Это усиливает стресс и чувство неконтролируемости системы. В результате критичной становится не только скорость реакции, но и сохранение управляемости контура принятия решений: ИИ-ассистент должен не подменять решение оператора, а поддерживать структуру анализа, не навязывая конкретного действия при неполноте данных.

Отдельной проблемой является различение высокой сложности самого инцидента и признаков дестабилизации анализа. Сложный инцидент сам по себе может порождать длительные циклы проверки, рост числа гипотез и возвраты к ранее просмотренным данным. Поэтому корректная архитектура ассистента должна учитывать не единичные признаки, а их устойчивые сочетания и контекст задачи, не сводя любой рост сложности к «перегрузке оператора».

Также необходимо учитывать организационные и этические рамки использования таких систем, включая разграничение режима помощи оператору и уровень управленческого мониторинга. Наличие у системы доступа к признакам дестабилизации анализа не должно автоматически означать возможность их использования в целях контроля производительности без отдельного регламентирования и разграничения прав доступа.

Таким образом, ИИ, будучи мощным усилителем аналитических возможностей SOC, остается ограниченным инструментом поддержки людей, если не дополняется контуром, ориентированным на надежность процесса анализа и допустимые границы влияния на решения оператора. Важно отметить, что ИИ-ассистенты, оптимизируя темп реагирования и предлагая готовые решения, могут снижать надежность принятия решений оператором за счет сглаживания неопределенности и ослабления критической проверки.

4. КОНЦЕПЦИЯ ИИ-АССИСТЕНТА, ОРИЕНТИРОВАННОГО НА ПОДДЕРЖКУ СТАБИЛЬНОСТИ РЕШЕНИЙ

Основные принципы. Предлагаемый подход исходит из трех базовых принципов:

1. Фокус на функциональных признаках дестабилизации анализа, а не на эмоциях. Искусственный интеллект не оценивает эмоции аналитика и не решает задачу психологической диагностики. Его задача заключается в фиксации наблюдаемых признаков дестабилизации процесса анализа в человеко-машинном контуре: зацикливания рассуждения, потери связности, смешения контекстов разных задач, роста повторных перепроверок без прироста результата и дрейфа относительно цели расследования. При этом такие признаки не должны интерпретироваться изолированно от характеристик самого инцидента.

2. Стабилизация процесса анализа, а не только ускорение принятия решений. Ассистент контролирует структуру рассуждения, согласованность анализа с целью и распределение внимания между смыс-

ловыми ветками. При необходимости он не навязывает готовое решение, а переводит взаимодействие в более консервативный режим поддержки: замедляет темп, дробит задачу на последовательные шаги, фиксирует контекст, выделяет зону неопределенности и помогает различать обратимые и необратимые действия на ранних стадиях расследования.

3. Этическое ограничение влияния. Использование данных о процессе анализа должно быть ограничено явно заданными правилами допустимого воздействия. Ассистент может предлагать паузу, изменение формата работы, пересборку плана или уточнение контекста, но не должен скрыто подталкивать оператора к конкретному решению. Под этическим контуром в работе понимается не свод моральных предписаний, а слой функциональных ограничений, разграничивающий режимы помощи оператору, порядок использования данных о признаках дестабилизации анализа и границы перехода от поддержки к управленческому контролю.

Концептуальная архитектура ИИ-ассистента. На концептуальном уровне ИИ-ассистент [16] строится поверх классического LLM-ядра и интеграции с SOC-инфраструктурой, включая источники событий, в том числе данные из CMDB (базы данных управления конфигурациями) и UEBA (анализа поведения пользователей и сущностей) (рис. 2). При этом архитектура включает не только технический контур обработки инцидента, но и когнитивно-аналитический слой поддержки надежности решений оператора.

В предлагаемой постановке учитываются два взаимосвязанных, но не тождественных контура:

• контур инцидента, отражающий сложность самой задачи: объем входящих данных, число затронутых активов, противоречивость признаков, количество допустимых гипотез и цену ошибки;

• контур дестабилизации анализа, отражающий особенности протекания процесса рассуждения в человеко-машинном взаимодействии: рост повторных проверок, потерю связности, смешение контекстов, циклические возвраты к уже просмотренным данным и дрейф относительно цели расследования.

Высокая сложность инцидента сама по себе может приводить к росту числа гипотез, возвратов к данным и длительным циклам проверки. Поэтому ассистент не должен трактовать любой рост сложности как признак перегрузки оператора. Переход к более консервативному режиму поддержки должен определяться не единичным наблюдаемым параметром, а устойчивым сочетанием признаков, не объясняемых только сложностью самой задачи.

Когнитивно-аналитический слой в этой архитектуре включает три функциональных блока:

• слой наблюдения функционального состояния анализа: выполняет анализ текста и паттернов взаимодействия оператора с системой и выявляет признаки дестабилизации процесса анализа без постановки «диагноза» оператору;

• слой стабилизации совместного рассуждения: контролирует смысловую плотность, когерентность и связность анализа, управляет темпом и объемом выдачи, структурирует расследование на последовательные шаги и помогает сохранять контекст при росте нагрузки;

• этический контур: определяет допустимые типы вмешательства, разграничивает контур помощи оператору и контур управленческого мониторинга, предотвращает скрытое навязывание действий и ограничивает использование информации о признаках дестабилизации анализа вне согласованного домена.

Такая архитектура позволяет рассматривать ИИ в SOC не только как ускоритель процессов, но и как стабилизатор когнитивной функции оператора, уменьшая риск выгорания и связанных с ним. Вклад ИИ-ассистента в сокращение времени реагирования в данной постановке связан не только с ускорением поиска ответа, но и навигацией оператора по полю допустимых решений, включая различение обратимых, условно обратимых и необратимых действий на ранних этапах расследования. Тем самым повышается качество решений при скачкообразном росте пиковой нагрузки.

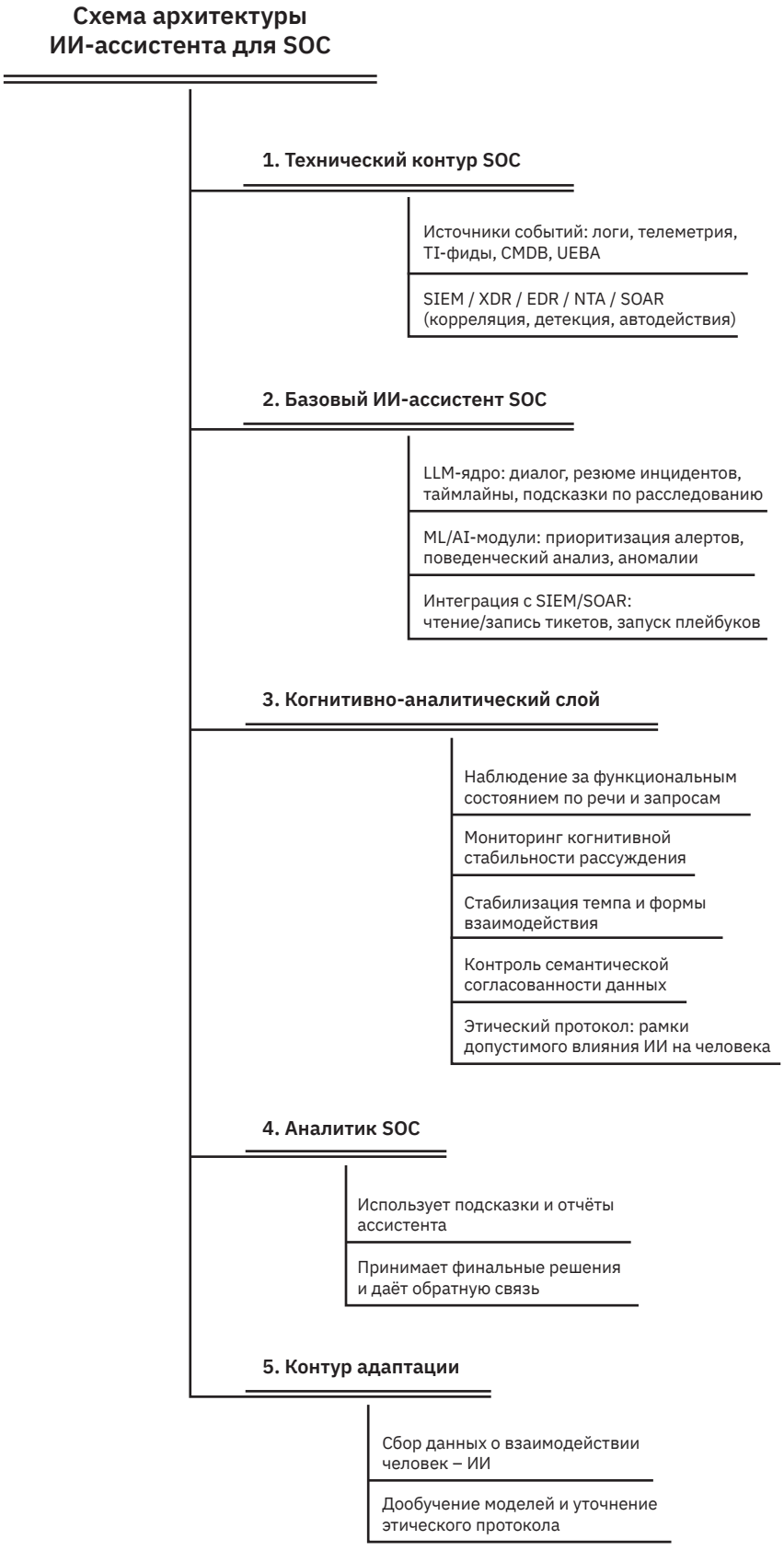


Рис. 2 | Концептуальная архитектура ИИ-ассистента для SOC

Fig. 2 | Conceptual architecture of an AI assistant for SOC

Эффективность подобного ассистента целесообразно оценивать не только по сокращению MTTR, но и по метрикам устойчивости анализа под нагрузкой. К ним могут относиться: снижение вариативности времени реакции при пиковых нагрузках; уменьшение числа пропущенных индикаторов компрометации; снижение частоты контекстных срывов и повторных перепроверок без прироста результата; сокращение доли необратимых действий, предпринятых до подтверждения рабочей гипотезы.

5. ВЫВОД

Работа в SOC характеризуется высокой когнитивной нагрузкой, утомлением от избыточных уведомлений и высоким уровнем профессионального выгорания, что напрямую влияет на частоту инцидентов и устойчивость команд. Современные ИИ-решения уже демонстрируют преимущества в скорости и полноте расследований, но в основном игнорируют психологическое состояние аналитиков, а в ряде случаев могут усиливать нагрузку за счет роста темпов работы и склонности к автоматизации.

Предложенный концептуальный контур ИИ-ассистента предполагает дополнение классического подхода функциональным слоем, ориентированным на наблюдение состояния оператора, стабилизацию совместного рассуждения и этическое ограничение влияния ИИ.

Такие системы могут рассматриваться как переход от преимущественно технологической оптимизации SOC к архитектурам, направленным на повышение надежности принятия решений ключевого элемента – человеческого мышления в условиях повышенной и нестационарной нагрузки.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Tines.** Voice of the SOC Analyst. 2022. URL: https://www.tines.com/reports/Tines%20Report%20-%20Voice%20of%20the%20SOC%20Analyst.pdf (дата обращения: 15.01.2026).
2. **Devo Technology.** 83 % of IT Security Professionals Say Burnout Causes Data Breaches. 2023. URL: https://www.devo.com/company/newsroom/it-security-professionals-say-burnout-causes-data-breaches (дата обращения: 15.01.2026).
3. **Карпова И. Л., Курилов А. В., Супрун А. Ф., Иванова Л. А.** Учет влияния человеческого фактора в моделях кибербезопасности // Проблемы информационной безопасности. Компьютерные системы. 2023. № 2 (54). С. 27–36.
4. **Cloud Security Alliance.** SOC Analyst Fatigue: What Our Data Says About Sustaining Investigation Speed and Quality. URL: https://cloudsecurityalliance.org/blog/2025/10/10/soc-analyst-fatigue-what-our-data-says-about-sustaining-investigation-speed-and-quality (дата обращения: 15.01.2026).
5. **ГК «Солар».** «Солар» выходит на рынок SIEM: две технологии в одном продукте и до 40 % экономии на внедрении. URL: https://rt-solar.ru/events/news/6173/ (дата обращения: 15.01.2026).
6. **Cyber Sierra.** What Is Alert Fatigue and How to Combat It in Your SOC. URL: https://cybersierra.co/blog/alert-fatigue-in-soc/ (дата обращения: 15.01.2026).
7. **Гарькушев А. Ю., Липис А. В., Карпова И. Л. и др.** Оценка компетентности интеллектуальной системы управления информационной безопасностью // Проблемы

информационной безопасности. Компьютерные системы. 2024. № 1(58). С. 18–27. DOI: 10.48612/jisp/92vv-6m6t-7tmh.

8. **Ведерников Ю. В., Гарькушев А. Ю., Липис А. В., Супрун А. Ф.** Реконфигурация модели развития системы управления информационной безопасностью: взаимодействие базовых модулей с оператором // Проблемы информационной безопасности. Компьютерные системы. 2024. № 2(59). С. 9–19. DOI: 10.48612/jisp/amv1-kdnf-zaae.
9. **Suprun A. F., Gar'kushev A. Yu., Lipis A. V. et al.** Assessment of the Competence of an Intelligent Information Security Management System // Automatic Control and Computer Sciences. 2024. Vol. 58. № 8. P. 1429–1435. DOI: 10.3103/S0146411624701220.
10. **Гарькушев А. Ю., Сазыкин А. М., Шалковская А. А.** Учет обоснованности в моделях оценки эффективности информационно-управляющих систем // Вопросы оборонной техники. Серия 16: Технические средства противодействия терроризму. 2024. № 3–4 (189–190). С. 40–44. DOI: 10.53816/23061456_2024_3-4_40.
11. **Гарькушев А. Ю., Липис А. В., Карпова И. Л., Супрун А. Ф.** Моделирование работы сотрудника службы информационной безопасности промышленного предприятия // Проблемы информационной безопасности. Компьютерные системы. 2023. № 3(56). С. 148–153. DOI: 10.48612/jisp/1vmb-73pk-5e9e.
12. **Dropzone AI.** Alert Fatigue in Cybersecurity: AI-Powered SOC Solutions Guide. URL: https://www.dropzone.ai/blog/how-to-address-cybersecurity-alert-fatigue-with-ai (дата обращения: 15.01.2026).
13. **Bria M., Spânu F., Băban A., Dumitraşcu D. L.** Maslach Burnout Inventory – General Survey: Factorial Validity and Invariance among Romanian Healthcare Professionals // Burnout Research. 2014. Vol. 1. Iss. 3. P. 103–111. DOI: 10.1016/j.burn.2014.09.001.
14. **Maslach C., Leiter M. P.** Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry // World Psychiatry. 2016. Vol. 15. № 2. P. 103–111. DOI: 10.1002/wps.20311.
15. ИИ-технологии в SIEM-системе KUMA. URL: https://www.kaspersky.ru/blog/ai-technology-in-kaspersky-siem/39252 (дата обращения: 15.01.2026).
16. **Глушко А.** Введение: современные вызовы для центров реагирования. URL: https://ptsecurity.com/research/analytics/autonomous-socs-future-of-cybersecurity-monitoring-and-incident-response/?ysclid=mp56tgtis9263489058 (дата обращения: 15.01.2026).

REFERENCES

1. **Tines.** Voice of the SOC Analyst. 2022. URL: https://www.tines.com/reports/Tines%20Report%20-%20Voice%20of%20the%20SOC%20Analyst.pdf (accessed: 15.01.2026).
2. **Devo Technology.** 83 % of IT Security Professionals Say Burnout Causes Data Breaches. 2023. URL: https://www.devo.com/company/newsroom/it-security-professionals-say-burnout-causes-data-breaches (accessed: 15.01.2026).
3. **Karpova I. L., Kurilov A. V., Suprun A. F., Ivanova L. A.** Accounting for the impact of the human factor in cyber security models. *Problems of information security. Computer systems*. 2023. No. 2 (54), pp. 27–36. (In Russian)
4. **Cloud Security Alliance.** SOC Analyst Fatigue: What Our Data Says About Sustaining Investigation Speed and Quality. URL: https://cloudsecurityalliance.org/blog/2025/10/10/soc-analyst-fatigue-what-our-data-says-about-sustaining-investigation-speed-and-quality (accessed: 15.01.2026).
5. **Solar Group.** Solar enters the SIEM market: two technologies in one product and up to 40 % cost savings. URL: https://rt-solar.ru/events/news/6173/ (accessed: 15.01.2026).
6. **Cyber Sierra.** What Is Alert Fatigue and How to Combat It in Your SOC. URL: https://cybersierra.co/blog/alert-fatigue-in-soc/ (accessed: 15.01.2026).
7. **Garkushev A. Yu., Lipis A. V., Karpova I. L. et al.** Assessment of the competence of the intelligent information security management system. *Problems of information security. Computer systems*. 2024. No. 1, pp. 18–27. DOI: 10.48612/jisp/92vv-6m6t-7tmh. (In Russian)
8. **Vedernikov Yu. V., Garkushev A. Yu., Lipis A. V., Suprun A. F.** Reconfiguration of the

system development model information security management: interaction of base modules with the operator. *Problems of information security. Computer systems*. 2024. No. 2, pp. 9–19. DOI: 10.48612/jisp/amv1-kdnf-zaae. (In Russian)
9. **Suprun A. F., Gar'kushev A. Yu., Lipis A. V. et al.** Assessment of the Competence of an Intelligent Information Security Management System. *Automatic Control and Computer Sciences*. 2024. Vol. 58. No. 8, pp. 1429–1435. DOI: 10.3103/S0146411624701220.
10. **Garkushev A. Yu., Sazykin A. M., Shalkovskaya A. A.** Accounting for justification in models for evaluating the effectiveness of information and control systems. *Defense Technology Issues. Series 16: Counter-Terrorism Technical Means*. 2024. No. 3–4(189–190), pp. 40–44. DOI: 10.53816/23061456_2024_3-4_40. (In Russian)
11. **Garkushev A. Yu., Lipis A. V., Karpova I. L., Suprun A. F.** Modeling of work of an employee of the information security service of an industrial enterprise. *Problems of information security. Computer systems*. 2023. No. 3(56), pp. 148–153. DOI: 10.48612/jisp/1vmb-73pk-5e9e. (In Russian)
12. **Dropzone AI.** Alert Fatigue in Cybersecurity: AI-Powered SOC Solutions Guide. URL: https://www.dropzone.ai/blog/how-to-address-cybersecurity-alert-fatigue-with-ai (accessed: 15.01.2026).
13. **Bria M., Spânu F., Băban A., Dumitraşcu D. L.** Maslach Burnout Inventory – General Survey: Factorial Validity and Invariance among Romanian Healthcare Professionals. *Burnout Research*. 2014. Vol. 1. Iss. 3, pp. 103–111. DOI: 10.1016/j.burn.2014.09.001.
14. **Maslach C., Leiter M. P.** Understanding the Burnout Experience: Recent Research and Its Implications for Psychiatry. *World Psychiatry*. 2016. Vol. 15. No. 2, pp. 103–111. DOI: 10.1002/wps.20311.
15. Use of artificial intelligence technologies in Kaspersky SIEM. URL: https://www.kaspersky.ru/blog/ai-technology-in-kaspersky-siem/39252 (accessed: 15.01.2026). (In Russian)
16. **Glushko A.** Introduction: modern challenges for response centers. URL: https://ptsecurity.com/research/analytics/autonomous-socs-future-of-cybersecurity-monitoring-and-incident-response/?ysclid=mp56tgtis9263489058 (accessed: 15.01.2026). (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ЛАРИОНОВА Екатерина Владимировна – канд. техн. наук, преподаватель, Санкт-Петербургский государственный морской технический университет, Россия, 190121, Санкт-Петербург, ул. Лоцманская, д. 3
E-mail: cf.82@mail.ru
ORCID: 0009-0009-1677-8355

LARIONOVA Ekaterina V. – Candidate of Engineering Sciences, Lecturer, Saint-Petersburg State Marine Technical University, Russia, 190121, St. Petersburg, Lotsmanskaya str., 3

БУНАС Ирина Леонидовна – преподаватель, Санкт-Петербургский государственный морской технический университет, Россия, 190121, Санкт-Петербург, ул. Лоцманская, д. 3
E-mail: ik070889@gmail.ru

BUNAS Irina L. – Lecturer, Saint-Petersburg State Marine Technical University, Russia, 190121, St. Petersburg, Lotsmanskaya str., 3

ГАРЬКУШЕВ Александр Юрьевич – канд. техн. наук, доцент, заведующий кафедрой, Санкт-Петербургский государственный морской технический университет, Россия, 190121, Санкт-Петербург, ул. Лоцманская, д. 3
E-mail: sangark@mail.ru
ORCID: 0000-0001-6695-2328

GARKUSHEV Alexander Yu. – Candidate of Engineering Sciences, Associate Professor, Head of Department, Saint-Petersburg State Marine Technical University, Russia, 190121, St. Petersburg, Lotsmanskaya str., 3

СУПРУН Александр Федорович – канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: afs54@inbox.ru
ORCID: 0000-0001-9665-0128

SUPRUN Alexander F. – Candidate of Engineering Sciences, Associate Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья
DOI 10.66424/2071-8217-2026-2-6
УДК 004.056

ДЕОБФУСКАЦИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ С ИСПОЛЬЗОВАНИЕМ ПРОМЕЖУТОЧНОГО ПРЕДСТАВЛЕНИЯ LLVM

Н. А. Милютин, Т. Д. Овасапян*, Д. В. Иванов

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *otd@ibks.spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Милютин Н. А., Овасапян Т. Д., Иванов Д. В. Деобфускация вредоносного программного обеспечения с использованием промежуточного представления LLVM // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 70–81.
DOI: 10.66424/2071-8217-2026-2-6

ПОСТУПИЛА 05.03.2026
ПРИНЯТА 05.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Рассматривается задача автоматизации деобфускации вредоносного программного обеспечения. Предложен метод, основанный на промежуточном представлении LLVM, объединяющий динамическую распаковку с трассировкой, гибридное (trace-assisted) восстановление графа потока управления и итеративную девиртуализацию. Разработан программный прототип, реализующий предложенный метод. Проведена экспериментальная оценка, подтвердившая применимость метода к снятию обфускации классов: упаковка, искажение потока управления, обфускация инструкций и виртуализация кода.

КЛЮЧЕВЫЕ СЛОВА

Обфускация, деобфускация, LLVM IR, девиртуализация, распаковка, восстановление графа потока управления

Original article
DOI 10.66424/2071-8217-2026-2-6

DEOBFUSCATION OF MALICIOUS SOFTWARE USING LLVM INTERMEDIATE REPRESENTATION

N. A. Milyutin, T. D. Ovasapyan*, D. V. Ivanov

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *otd@ibks.spbstu.ru

FOR CITATION

Milyutin N. A., Ovasapyan T. D., Ivanov D. V. Deobfuscation of malicious software using LLVM intermediate representation. *Problems of information security. Computer systems*. 2026. No. 2, pp. 70–81. DOI: 10.66424/2071-8217-2026-2-6 (In Russian)

RECEIVED 05.03.2026
ACCEPTED 05.05.2026
PUBLICATION 15.06.2026

ABSTRACT

The problem of automating deobfuscation of malicious software is considered. A method based on the LLVM intermediate representation is proposed that combines dynamic unpacking with tracing, hybrid (trace-assisted) restoration of the control flow graph and iterative devirtualization. A software prototype has been developed that implements the proposed method. An experimental evaluation was carried out, confirming the applicability of the approach to removing class obfuscation: packaging, control flow distortion, instruction obfuscation, and code virtualization.

KEYWORDS

Obfuscation, deobfuscation, LLVM IR, devirtualization, unpacking, control flow graph recovery

1. ВВЕДЕНИЕ

Одним из способов сокрытия логики работы программ является обфускация кода – процесс преднамеренного усложнения структуры исходного кода или исполняемого файла для затруднения анализа. Первоначально обфускация применялась для защиты интеллектуальной собственности, но в текущих реалиях широко используется во вредоносном программном обеспечении [1].

Разработка методов и инструментов, позволяющих автоматически проводить деобфускацию – восстановление исходной или приближенной к исходной структуре программ, – приобретает особую значимость, поскольку позволяет ускорить реверс-инжиниринг при анализе потенциально вредоносных объектов.

В мировой и отечественной практике представлено множество инструментов для статического и динамического анализа программ, однако автоматизация процесса деобфускации остается недостаточно решенной задачей. Существующие инструменты обычно требуют глубоких экспертных знаний, ручной настройки и не обеспечивают высокой степени восстановления кода. Более того, из-за разнообразия техник обфускации (изменение графа потока управления, вставка ложных операций, шифрование строк, виртуализация кода), универсального решения, охватывающего широкий класс случаев, до сих пор не существует.

Целью работы является автоматизация деобфускации вредоносного программного обеспечения за счет его анализа на уровне промежуточного представления LLVM.

2. СОВРЕМЕННЫЕ РЕШЕНИЯ

Классификация техник обфускации. Техники обфускации, применяемые во вредоносном ПО, классифицируются по четырем основным категориям [2]. Обфускация данных направлена на предотвращение извлечения критически важной информации методами статического анализа. Данные преобразуются в нечитаемую форму посредством шифрования строк, констант, разбиения данных с поздней инициализацией, использования вычислимых констант. Восстановление исходных значений происходит только в момент использования данных при работе программы.

Искажение графа потока управления направлено на увеличение цикломатической сложности программы, вследствие чего декомпиляторы не могут корректно восстановить структуру кода. Ключевой техникой является control-flow flattening [3] – преобразование иерархической структуры кода в «плоскую» структуру, где базовые блоки помещаются внутрь цикла с оператором switch, а порядок исполнения определяется переменной состояния.

К данному классу также относятся: инъекция ложных путей (opaque predicates), встраивание функций (inlining) и вынесение фрагментов в отдельные функции (outlining).

Упаковка и маскировка входной точки направлена на сокрытие исполняемого файла до момента исполнения [4]. Файл разделяется на полезную нагрузку (сжатую или зашифрованную) и распаковщик. Современные образцы ВПО применяют многоуровневую упаковку (multi-layer packing [5]), при которой каждый слой выполняет подготовку следующего слоя. Задача автоматической распаковки сводится к корректному воспроизведению всей цепочки зависимых исполнений.

Преобразование логики на уровне инструкций включает подстановку функционально эквивалентных инструкций, вставку мусорного кода и самомодифицирующийся код. Наиболее мощной техникой является виртуализация кода [6–8] при которой исходный машинный код трансформируется в байт-код пользовательской архитектуры, исполняемый встроенной виртуальной машиной (интерпретатором). Виртуальная машина содержит набор виртуальных команд, транслятор и интерпретатор, реализующий цикл выборки-декодирования-исполнения.

Методы распаковки. Ключевым этапом анализа упакованного ВПО является детектирование момента восстановления оригинальной точки входа. Выделяются следующие методы:

• Write-then-Execute (WxE) – эвристика, отслеживающая пары событий «запись в память – исполнение». Реализуется на двух уровнях гранулярности: побайтный мониторинг (Renovo [9], Ether [10]) и постраничный (OmniUnpack [11], GeMU [12]). Метод инвариантен к алгоритмам упаковки.

• Эвристики на основе энтропии: уменьшение энтропии участка памяти может служить индикатором завершения распаковки, однако возможны ложные срабатывания.

• Эвристики на основе API-паттернов: обнаружение последовательностей вызовов VirtualAlloc, WriteProcessMemory и подобных. Ненадежны, поскольку ВПО может скрывать импорт библиотек.

Среди рассмотренных решений наиболее полным является распаковщик GeMU, обеспечивающий полное послойное извлечение (Layer-by-Layer) и поддержку многопроцессности.

Методы восстановления графа потока управления. Граф потока управления (CFG) задает структуру программы: базовые блоки и переходы между ними. Обфускация, искажающая CFG (control-flow flattening, инъекция ложных путей), разрушает эту структуру, вследствие чего декомпиляторы выдают нечитаемый «спагетти-код», а чисто статический анализ не может однозначно определить цели косвенных переходов. Восстановление CFG – ключевой этап деобфускации, без которого невозможен последующий анализ семантики кода. Представим современные методы, решающие данную задачу:

• Статический анализ с абстрактной интерпретацией (Jakstab) [13] – метод итеративного дизассемблирования, использующий Value Set Analysis для разрешения косвенных переходов. Ограничен архитектурой x86 и подвержен проблеме потери точности (over-approximation).

• Посимвольное выполнение (Symbolic Execution) – метод, используемый в средствах KLEE, Angr, Triton, оперирующий символьными переменными с использованием SMT-решателя (Z3). Обеспечивает высокую точность, однако подвержен проблеме экспоненциального роста путей (path explosion).

• Итеративный лифтинг LLVM (SATURN) [14] – метод, объединяющий поднятие бинарного кода в LLVM IR и использование оптимизаций компилятора. Алгоритм выполняет трансляцию инструкций в LLVM IR, применяет стандартные оптимизации и SMT-оптимизатор Souper [15], разрешает переходы с помощью SMT-решателя и итеративно расширяет фронт обнаруженных адресов. Преимуществом является использование готовой инфраструктуры LLVM для автоматического устранения обфускации на этапе построения графа. В качестве недостатка можно отметить ограниченность статического подхода в обработке самомодифицирующегося кода.

Методы деобфускации виртуализированного кода. Виртуализация кода относится к числу наиболее устойчивых техник обфускации: нативный код подменяется байт-кодом авторской архитектуры, исполняемым встроенным интерпретатором. В результате скрываются и поток управления, и поток данных. Аналитику приходится сначала реконструировать саму виртуальную машину, чтобы понять ее систему команд и структуру диспетчера. Девиртуализация сводится к двум задачам – анализу структуры виртуальной машины (ВМ) и переводу ее байт-кода обратно в нативный код или промежуточное представление. Рассмотрим основные методы:

• программный синтез (Syntia): обфусцированный блок рассматривается как черный ящик, для которого синтезируется эквивалентная программа (инвариантен к сложности обфускации, но ограничен масштабируемостью);

• статический паттерн-матчинг: поиск известных сигнатур ВМ (быстр, но неустойчив к полиморфизму обфускатора);

• символьное выполнение с анализом трассы [16, 17]: запись трассы, taint-анализ, посимвольное исполнение обработчиков ВМ (позволяет точно понять семантику обработчиков, но зависит от длины трассы).

3. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ И ИНСТРУМЕНТОВ

Результаты сравнительного анализа распаковщиков представлены в табл. 1. GeMU в силу своих возможностей выбран в качестве распаковщика для дальнейшей деобфускации. В табл. 2 представлено сравнение методов восстановления графа потока управления.

Наиболее перспективным для автоматизации деобфускации является метод, используемый SATURN: промежуточное представление LLVM IR унифицирует анализ для разных архитектур и позволяет применять оптимизации компиляторов.

Таблица 1 | Сравнение распаковщиков

Table 1 | Comparison of unpackers

Инструмент	Метод обнаружения	Извлечение слоев	Поддержка многопроцессности	Основные ограничения
GeMU	WxE	Полное (Layer-by-Layer)	Да	Требует ресурсов эмуляции (QEMU)
Renovo/Ether	WxE	Частичное	Да	Устаревшие реализации, высокие накладные расходы
OmniUnpack	WxE + внешние эвристики	Нет	Ограничена	Зависимость от внешних детекторов

Таблица 2 | Сравнение методов к восстановлению графа потока управления

Table 2 | Comparison of approaches to control flow graph reconstruction

Метод	Инструменты	Точность	Скорость
Абстрактная интерпретация	Jakstab	Средняя (false-positives)	Средняя
Посимвольное исполнение	KLEE, Triton, Angr	Высокая	Низкая (path explosion)
Итеративный лифтинг LLVM	SATURN	Высокая (SMT + оптимизации)	Средняя

Однако чисто статический подход не способен обработать код, формируемый во время выполнения.

На основании проведенного анализа определен комбинированный метод:

• использование LLVM-лифтинга для перевода кода в унифицированное представление;

• расширение статического анализа динамической трассой для преодоления ограничений чисто статических методов;

• применение SMT-решателей для очистки локальных обфускаций (MBA).

Ни один из рассмотренных инструментов не обеспечивает одновременно распаковку, восстановление CFG, деобфускацию инструкций и девиртуализацию в рамках единого конвейера.

4. ОПИСАНИЕ ПРЕДЛАГАЕМОГО МЕТОДА

Предлагаемый метод автоматизации деобфускации основывается на промежуточном представлении LLVM и состоит из трех последовательных этапов.

Этап 1: распаковка и выборочная трассировка. На начальном этапе проводится динамическая распаковка кода с целью получения чистого бинарного образа для дальнейшего лифтинга. Используется метод на основе инструмента GeMU [12], реализующий принцип Write-then-Execute.

Работа этапа:

1. Эмуляция среды исполнения: образец запускается в изолированной эмулируемой среде (QEMU), что скрывает факт анализа от средств антиотладки.

2. Мониторинг страниц памяти: при записи на страницу она помечается как кандидат (dirty); при исполнении инструкций из такой страницы срабатывает триггер детектора.

3. Итеративное извлечение слоев: при каждом обнаружении нового слоя создается дамп процесса. После этого страница сбрасывает статус кандидата и эмуляция продолжается.

Дополнительно фиксируется селективная трасса исполнения, включающая:

• поток управления: адрес инструкции, адреса переходов, исходы условных ветвлений;

• контекст вычислений: значения регистров на границах базовых блоков;

• события памяти: чтения и записи, влияющие на вычисление целей переходов;

• карту адресного пространства: регионы base/size/prot с обновлениями при VirtualAlloc/VirtualProtect.

Необходимость трассы обусловлена тем, что значимая часть логики обфусцированной программы реализуется через косвенные переходы, вычисляемые во время выполнения, а также через динамически вычисляемые константы, которые не выводятся из статического контекста.

Результатом этапа является набор дампов памяти, содержащих распакованные фрагменты кода, и синхронизированная трасса исполнения.

Этап 2: лифтинг и восстановление графа потока управления. На данном этапе реализуется основной алгоритм деобфускации, объединяющий восстановление CFG и очистку кода от обфусцированных инструкций в едином цикле анализа. Это принципиально важно, поскольку точное определение адресов переходов часто невозможно без предварительного упрощения арифметики.

С учетом наличия трассы этап расширяется до гибридного (trace-assisted) восстановления CFG: статический анализ используется как базовый механизм, а трасса применяется как источник фактических целей при неоднозначности.

Алгоритм работы:

1. Динамическое расширение фронта: анализ начинается с известной точки входа; адреса новых блоков извлекаются из очереди и поднимаются в LLVM IR.

2. Оптимизации: к каждому поднятому блоку применяются стандартные оптимизации LLVM (Constant Propagation, DCE) и оптимизатор Souper (SMT-решатель Z3, символьное выполнение KLEE). При наличии записей трассы добавляются

ограничения (assumptions), фиксирующие конкретные значения регистров.

3. Отсечение невыполнимых ветвей: SMT-решатель проверяет выполнимость условий; при доказанной тривиальности ветвления граф упрощается.

4. Trace-assisted разрешение переходов: для прямых переходов цели определяются статически; для косвенных – при неоднозначности используется трасса для получения фактической цели с добавлением ограничения.

Цикл продолжается до исчерпания очереди блоков. Результатом является восстановленный модуль LLVM IR и CFG, корректный для наблюдаемого исполнения.

Этап 3: девиртуализация. Девиртуализация трактуется как процедура нормализации и редукции восстановленного IR, направленная на устранение артефактов виртуальной машины.

Для описания этапа необходимо ввести следующие определения: псевдопамять – абстрактный объект памяти в IR, от базового указателя которого вычисляются адреса обращений; псевдостек – часть псевдопамяти, рассматриваемая как стек по правилу классификации адресов.

Трасса на данном этапе используется для фиксации значений, определяющих выбор обработчика, конкретизации чтений из псевдопамяти, направленной развертки цикла диспетчеризации.

Алгоритм конвейера:

1. Инициализация: регистрация анализа через метод llvm:: PassBuilder [18].

2. Специализированная нормализация (trace-assisted):

• подстановка значений для выражений адресов и условий;

• constant propagation для чтений по детерминированным адресам;

• нормализация чтений и устранение избыточных преобразований типов;

• отделение стековой области от общей псевдопамяти;

• канонизация GEP/inttoptr и снижение ложных зависимостей.

3. Финальная оптимизация: применяется стандартный оптимизационный набор LLVM уровня O2.

5. РАЗРАБОТКА ПРОТОТИПА ПРОГРАММНОГО СРЕДСТВА АВТОМАТИЗАЦИИ ДЕОБФУСКАЦИИ

Разработанный прототип состоит из пяти компонентов (см. рисунок):

1. Распаковщик – модифицированный проект GeMU, обеспечивающий извлечение распакованных слоев (WxE) и сбор селективной трассы исполнения.

2. Дизассемблер – IDA Pro с IDAPython-скриптом для извлечения листинга по заданному адресу.

3. Модуль трансляции – модифицированный mcsema_lift, выполняющий лифтинг в LLVM IR с одновременным сопоставлением событий трассы.

4. Модуль восстановления CFG – реализует worklist-обход и построение графа.

5. Модуль девиртуализации – выполняет специализированную нормализацию и развертку цикла диспетчера.

Распаковщик построен как модифицированный GeMU с двумя целями: устойчивое извлечение полезной нагрузки при многоуровневой распаковке и формирование селективной трассы.

Архитектура включает: среду исполнения (QEMU), монитор записи в память, монитор исполнения, менеджер слоев/дампов и модуль трассировки (Trace Recorder – добавлен в рамках модификации).

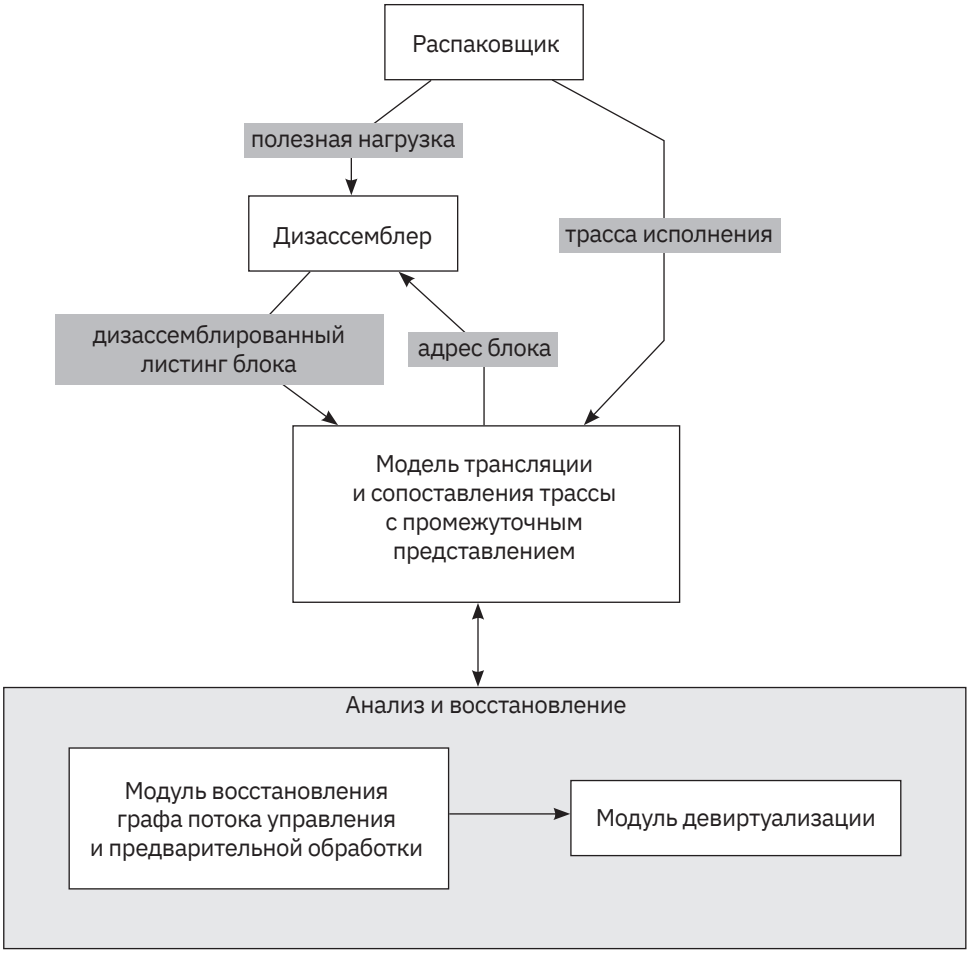
Трассировка фиксирует три типа событий:

1. Control-flow события: rip_before, kind (Jcc/JMP_indirect/CALL_indirect/RET), target_rip, taken.

2. Memory-access события: rip_before, kind (load/store), ea, size, value;

3. Карта памяти: регионы base/size/prot с обновлениями.

Модуль трансляции. Переход от бинарного слоя к LLVM IR организован как связка IDA → IDAPython – скрипт → модифицированный mcsema_lift [19]. Модуль трансляции одновременно выполняет лифтинг и сопоставляет события трассы с адресами инструкций. Для ускорения запросов к трассе вводится индекс TraceIndex, содержащий: observedTargets,



Схематическое представление прототипа

Schematic representation of the prototype

observedTaken, fallthroughRIP (control-flow), observedMemOps (memory-access) и memoryMaps (адресные пространства).

Инъекция динамических сведений разделяется на два механизма: Trace-assisted assumptions для условного перехода – фиксация наблюдаемого исхода через llvm.assume; Trace-assisted load concretization при наличии наблюдаемых memory-операций – подстановка константного значения вместо неопределенного чтения.

Для косвенных переходов наблюдаемые цели используются для развертки управления в явные ребра CFG.

Модуль девиртуализации. Модуль решает задачу развертки цикла диспетчера. После восстановления CFG косвенные переходы заменяются на switch или каскад условных переходов, но интерпретаторная модель сохраняется.

Анализ трассы и восстановление последовательности обработчиков: определяется базовый блок диспетчера (наибольшее количество control-flow событий), затем линейным проходом по трассе строится последовательность пар (virtual instruction pointer, handler).

Инъекция виртуального указателя: в тело цикла добавляется llvm.assume с фиксацией значения VIP (virtual instruction pointer), что делает диспетчеризацию детерминированной. Последующие оптимизации (Loop Unrolling, Constant Propagation, DCE) автоматически разворачивают и упрощают цикл.

Реализованы специализированные проходы оптимизации:

1. Constant propagation для чтений по константным адресам: замена чтений на константы при выполнении условий политики работы с памятью. Пример:

```
; До: чтение по детерминированному адресу управляет ветвлением
%p = getelementptr i8, ptr %memory, i64 5368725620
%flag = load i32, ptr %p
%is_zero = icmp eq i32 %flag, 0
br i1 %is_zero, label %fast, label %slow

; После constant propagation + SimplifyCFG + DCE:
%a = add i32 %x, 1
ret i32 %a
```

2. Сборка значений из сегментов: при чтении диапазона байтов строится разбиение на ref-сегменты (из предыдущих store) и mem-сегменты, итоговое значение собирается через операции извлечения байтов и битовых сдвигов.

3. Упрощение адресации: замена адресации от общей базы (GEP + offset) на прямое представление (inttoptr), что снижает число ложных зависимостей по данным. Пример:

```
; До:
%p = getelementptr i8, ptr %memory, i64 5368725620
; После:
%p = inttoptr i64 5368725620 to ptr
```

Тестирование. Для оценки разработан ряд тестовых наборов:

• dataset A (юнит-тесты): арифметика (работа с памятью), ветвления по флагам, обращение к памяти, косвенные переходы – 18 тестов;

• dataset B (виртуализация/контроль потока): бинарные файлы, собранные с Tigress Virtualize [20] и коммерческими ВМ-защитами (Themida/VMProtect);

• dataset C (упакованные семплы): упакованные PE файлы с overlay, формируемым модифицированным GeMU.

Для Dataset B верификация выполнялась сравнением с оригинальным ПО либо проверкой эквивалентности по инвариантам на фиксированном наборе входов. Для Dataset C фиксировались структурные метрики: стабилизация CFG, исчезновение VM-диспетчерного цикла, уменьшение доли псевдопамяти, рост константности адресов.

Результаты подтверждают, что предложенный конвейер обеспечивает практическую применимость для широкого класса задач деобфускации: лифтинг в LLVM IR

Таблица 3 | Сравнение реализованного прототипа с фреймворком Saturn

Table 3 | Comparison of the implemented software with the Saturn framework

Класс техники	SATURN	Разработанный прототип
Константные/арифметические маски (MBA)	+	+
Opaque predicates	+	+
Dead code insertion	+	+
Bogus control flow	+	+
Integer encoding/flattening по данным	+	+
Упакованные семплы (overlay)	–	+
Девиртуализация (устранение ВМ-диспетчеризации)	–	+

Таблица 4 | Результаты тестирования

Table 4 | Test results

Датасет	Программы	Проверка результата	Снятие обфускации
A	18 юнит-тестов (bench_add, test_branch_*, test_memory и др.)	Юнит-тест	+ (все тесты пройдены)
B	tigress_virtualize_min.exe, themida_vm_sample.exe	Сравнение потоков управления	+ (для выбранной трассы)
C	upx_packed_sample.exe, packed_vm_sample.exe	Структурные метрики	Частично

с восстановлением CFG и итеративная нормализация обеспечивают заметное снижение структурного и инструкционного шума.

6. ЗАКЛЮЧЕНИЕ

В ходе работы рассмотрены и систематизированы основные классы обфускации, существенные для анализа бинарного кода: упаковка (включая многоуровневую распаковку), обфускация графа потока управления, обфускация данных и инструкций, а также виртуализация кода. Проведен сравнительный анализ существующих методов и инструментов деобфускации.

На основе анализа предложен метод автоматизации деобфускации, основанный на промежуточном представлении LLVM и объединяющий три этапа:

• динамическая распаковка с извлечением слоев по инварианту WxE и формирование селективной трассы исполнения;

• лифтинг с гибридным (trace-assisted) восстановлением графа потока управления, где наблюдаемые цели переходов и операции с памятью используются как ограничения для стабилизации анализа;

• девиртуализация как итеративная редукция LLVM IR стандартными и специализированными проходами оптимизаций, дополненная SMT-ориентированным упрощением выражений.

Разработан программный прототип, реализующий предложенный метод. В сравнении с ближайшим аналогом (SATURN) прототип дополнительно обеспечивает: поддержку упакованных семплов за счет overlay и девиртуализацию с устранением BM-диспетчеризации.

Тестирование показало, что прототип обеспечивает автоматизацию ключевых этапов восстановления семантики обфусцированного кода: восстановление графа потока управления, упрощение базовых блоков и управляющих условий, девиртуализацию, корректную работу с распакованными слоями. Ограничения метода определяются качеством трассы и точностью классификации адресных диапазонов, влияющими на полноту конкретизации памяти и степень редукции IR.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. Отчет лаборатории Касперского за второй квартал 2024. URL: https://securelist.ru/it-threat-evolution-q2-2024-pc-statistics/110425/ (дата обращения: 21.02.2026).
2. **Collberg C. S., Thomborson C.** Watermarking, tamper-proofing, and obfuscation tools for software protection // IEEE Transactions on Software Engineering. 2002. Vol. 28 (8). P. 735–746. DOI: 10.1109/TSE.2002.1027797.
3. **Laszlo T., Kiss A.** Obfuscating C++ programs via control flow flattening // Annales Universitatis Scientarum Budapestinensis. 2009. Vol. 30. № 1. P. 3–19.
4. **Jenke T., Liessem S., Padilla E., Bruckschen L.** A Measurement Study on Interprocess Code Propagation of Malicious Software // ICDF2C 2023. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. 2023. Vol 571. P. 264–282.
5. **Behera C., Bhaskari D. L.** Different Obfuscation Techniques for Code Protection // Procedia Computer Science. 2015. Vol. 70. P. 757–763. DOI: 10.1016/j.procs.2015.10.114.
6. **Kochberger P., Schrittwieser S., Schweighofer S. et al.** SoK: Automatic Deobfuscation of Virtualization-protected Applications // ARES 2021: The 16th International Conference on Availability, Reliability and Security. 2021. P. 1–15. DOI: 10.1145/3465481.3465772.
7. **Xiao X., Wang Y., Hu Yi., Gu D.** xvmp: An llvm-based code virtualization obfuscator // IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2023. P. 738–742. DOI: 10.1109/SANER56733.2023.00082.
8. **Junod P., Rinaldini J., Wehrli J., Michielin J.** Obfuscator LLVM Software Protection for the Masses // IEEE/ACM 1st International Workshop on Software Protection (SPRO). 2015. DOI: 10.1109/SPRO.2015.10.
9. **Kang M. G., Poosankam P., Yin H.** Renovo: A hidden code extractor for packed executables // Proceedings of the 2007 ACM workshop on Recurring Malcode. 2007. P. 46–53. DOI: 10.1145/1314389.1314399
10. **Dinaburg A., Royal P., Sharif M. I., Lee W.** Ether: malware analysis via hardware virtualization extensions // Proceedings of the 2008 ACM Conference on Computer and Communications Security (CCS 2008), 27–31 October 2008, Alexandria, Virginia, USA. 2008. P. 51–62. DOI: 10.1145/1455770.1455779
11. **Martignoni L., Christodorescu M., Jha S.** OmniUnpack: Fast, Generic, and Safe Unpacking of Malware // ACSAC. 2007. P. 431–441. DOI: 10.1109/ACSAC.2007.15
12. GeMU, the generic malware unpacker based on QEMU. URL: https://github.com/fkie-cad/GeMU (дата обращения: 21.02.2026).
13. **Kinder J.** Static analysis of x86 executables. Darmstadt: Technische Universität, 2010. P. 156–178.
14. **Garba P., Favaro M.** Software Deobfuscation Framework Based on LLVM // Proceedings of the 3rd ACM Workshop on Software Protection. 2019. P. 27–38.
15. Souper (superoptimizer for LLVM IR). URL: https://github.com/google/souper (дата обращения: 21.02.2026).
16. **Coogan K., Lu G., Debray S.** Deobfuscation of Virtualization-Obfuscated Software: A Semantics-Based Approach // Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011), 17–21 October 2011, Chicago, Illinois, USA. 2011. P. 275–284. DOI: 10.1145/2046707.2046739.
17. **Ovasapyan T. D., Knyazev P. V., Moskvin D. A.** Application of taint analysis to study the safety of software of the Internet of Things devices based on the ARM architecture // Automatic Control and Computer Sciences. 2020. Vol. 54. № 8. P. 834–840
18. McSema. URL: https://github.com/lifting-bits/mcsema (дата обращения: 21.02.2026).
19. LLVM's Analysis and Transform Passes. URL: https://github.com/llvm/llvm-project/blob/main/llvm/docs/Passes.rst (дата обращения: 21.02.2026).
20. Tigress C obfuscator. URL: https://tigress.wtf/ (дата обращения: 21.02.2026).

REFERENCES

1. Kaspersky Lab report for the second quarter of 2024. URL: https://securelist.ru/it-threat-evolution-q2-2024-pc-statistics/110425/ (accessed: 21.02.2026). (In Russian)

2. **Collberg C. S., Thomborson C.** Watermarking, tamper-proofing, and obfuscation tools for software protection. *IEEE Transactions on Software Engineering*. 2002. Vol. 28 (8), pp. 735–746. DOI: 10.1109/TSE.2002.1027797.
3. **Laszlo T., Kiss A.** Obfuscating C++ programs via control flow flattening. *Annales Universitatis Scientarum Budapestinensis*. 2009. Vol. 30. No. 1, pp. 3–19.
4. **Jenke T., Liessem S., Padilla E., Bruckschen L.** A Measurement Study on Interprocess Code Propagation of Malicious Software. ICDF2C 2023. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. 2023. Vol 571, pp. 264–282.
5. **Behera C., Bhaskari D. L.** Different Obfuscation Techniques for Code Protection. Procedia Computer Science. 2015. Vol. 70, pp. 757–763. DOI: 10.1016/j.procs.2015.10.114.
6. **Kochberger P., Schrittwieser S., Schweighofer S. et al.** SoK: Automatic Deobfuscation of Virtualization-protected Applications. ARES 2021: The 16th International Conference on Availability, Reliability and Security. 2021, pp. 1–15. DOI: 10.1145/3465481.3465772.
7. **Xiao X., Wang Y., Hu Yi., Gu D.** xvmp: An llvm-based code virtualization obfuscator. IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). 2023, pp. 738–742. DOI: 10.1109/SANER56733.2023.00082.
8. **Junod P., Rinaldini J., Wehrli J., Michielin J.** Obfuscator LLVM Software Protection for the Masses. IEEE/ACM 1st International Workshop on Software Protection (SPRO). 2015. DOI: 10.1109/SPRO.2015.10.
9. **Kang M. G., Poosankam P., Yin H.** Renovo: A hidden code extractor for packed executables. Proceedings of the 2007 ACM workshop on Recurring Malcode. 2007, pp. 46–53. DOI: 10.1145/1314389.1314399
10. **Dinaburg A., Royal P., Sharif M. I., Lee W.** Ether: malware analysis via hardware virtualization extensions. Proceedings of the 2008 ACM Conference on Computer and Communications Security (CCS 2008), 27–31 October 2008, Alexandria, Virginia, USA. 2008, pp. 51–62. DOI: 10.1145/1455770.1455779
11. **Martignoni L., Christodorescu M., Jha S.** OmniUnpack: Fast, Generic, and Safe Unpacking of Malware. ACSAC. 2007, pp. 431–441. DOI: 10.1109/ACSAC.2007.15
12. GeMU, the generic malware unpacker based on QEMU. URL: https://github.com/fkie-cad/GeMU (accessed: 21.02.2026).
13. **Kinder J.** Static analysis of x86 executables. Darmstadt: Technische Universität, 2010, pp. 156–178.
14. **Garba P., Favaro M.** Software Deobfuscation Framework Based on LLVM. Proceedings of the 3rd ACM Workshop on Software Protection. 2019, pp. 27–38.
15. Souper (superoptimizer for LLVM IR). URL: https://github.com/google/souper (accessed: 21.02.2026).
16. **Coogan K., Lu G., Debray S.** Deobfuscation of Virtualization-Obfuscated Software: A Semantics-Based Approach. Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS 2011), 17–21 October 2011, Chicago, Illinois, USA. 2011, pp. 275–284. DOI: 10.1145/2046707.2046739.
17. **Ovasapyan T. D., Knyazev P. V., Moskvin D. A.** Application of taint analysis to study the safety of software of the Internet of Things devices based on the ARM architecture. *Automatic Control and Computer Sciences*. 2020. Vol. 54. No. 8, pp. 834–840
18. McSema. URL: https://github.com/lifting-bits/mcsema (accessed: 21.02.2026).
19. LLVM's Analysis and Transform Passes. URL: https://github.com/llvm/llvm-project/blob/main/llvm/docs/Passes.rst (accessed: 21.02.2026).
20. Tigress C obfuscator. URL: https://tigress.wtf/ (accessed: 21.02.2026).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

МИЛЮТИН Никита Алексеевич – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: milyutin.na@edu.spbstu.ru
ORCID: 0009-0002-7398-5069

MILYUTIN Nikita A. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

ОВАСАПЯН Тигран Джаникович – канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: otd@ibks.spbstu.ru
ORCID: 0000-0002-2009-5460

OVASAPYAN Tigran D. – Candidate of Engineering Sciences, Associate Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

ИВАНОВ Денис Вадимович – канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: ivanov@ibks.spbstu.ru
ORCID: 0000-0001-8206-2915

IVANOV Denis V. – Candidate of Engineering Sciences, Associate Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья
DOI 10.66424/2071-8217-2026-2-7
УДК 004.032.26

АУГМЕНТАЦИЯ ТРАФИКА ИНТЕРНЕТА ВЕЩЕЙ С ИСПОЛЬЗОВАНИЕМ ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНЫХ СЕТЕЙ

В. В. Платонов[1*], Д. А. Скиба[2]

[1]Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия
[2]АО «ИнфоТеКС», Москва, Россия

✉ *plato@ibks.spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Платонов В. В., Скиба Д. А. Аугментация трафика Интернета вещей с использованием генеративно-состязательных сетей // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 82–91.
DOI: 10.66424/2071-8217-2026-2-7

ПОСТУПИЛА 09.02.2026
ПРИНЯТА 04.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Исследуется проблема критического дисбаланса классов в системах обнаружения вторжений (IDS) для сетей Интернета вещей (IoT). Проведено сравнительное исследование результативности различных методов аугментации данных: пяти архитектур генеративно-состязательных сетей (CopulaGAN, CTGAN, CTAB-GAN+ и модификаций MC-WGAN-GP и TMG-GAN) в сопоставлении с традиционными подходами (SMOTE, случайное пересемплирование). Показано, что применение аугментации (как этапа подготовки данных) позволяет восстановить работоспособность классификатора LightGBM в сценариях с критическим дисбалансом, увеличивая показатель F1-macro с 0,03 до 0,81.

КЛЮЧЕВЫЕ СЛОВА

Аугментация данных, генеративно-состязательные сети, дефицит данных, система обнаружения вторжений, Интернет вещей

Original article
DOI 10.66424/2071-8217-2026-2-7

IOT DATA AUGMENTATION USING GENERATIVE ADVERSARIAL NETWORKS

V. V. Platonov[1*], D. A. Skiba[2]

[1]Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia
[2]JSC “InfoTeX”, Moscow, Russia

✉ *plato@ibks.spbstu.ru

FOR CITATION

Platonov V. V., Skiba D. A. IoT data augmentation using generative adversarial networks. *Problems of information security. Computer systems*. 2026. No. 2, pp. 82–91.
DOI: 10.66424/2071-8217-2026-2-7
(In Russian)

ABSTRACT

The article investigates the problem of critical class imbalance in intrusion detection systems (IDS) for Internet of Things (IoT) networks. A comparative study of data augmentation methods was conducted, evaluating five generative adversarial network (GAN) architectures (CopulaGAN, CTGAN, CTAB-GAN+ and modified versions of MC-WGAN-GP and TMG-GAN) against traditional approaches (SMOTE, random oversampling). The study shows that data augmentation (as a

RECEIVED 09.02.2026
ACCEPTED 04.05.2026
PUBLICATION 15.06.2026

data preprocessing stage) enables the restoration of the LightGBM classifier's performance in critical imbalance scenarios, increasing the F1-macro score from 0.03 to 0.81.

KEYWORDS

Data augmentation, generative adversarial networks, data deficiency, intrusion detection system, Internet of Things

1. ВВЕДЕНИЕ

В эпоху цифровизации, когда сети Интернета вещей (Internet of Things – IoT) проникают во все сферы жизни – от промышленных систем и умных городов до бытовых устройств, – объем генерируемого трафика достигает триллионов пакетов ежедневно, создавая новые вызовы для обеспечения кибербезопасности. По оценкам аналитиков, в 2025 г. количество подключенных IoT-устройств превысит 20 млрд с прогнозируемым ростом на 14% ежегодно [1]. Это неизбежно ведет к эскалации угроз: ежедневно фиксируется около 820 тыс. атак [2] на IoT-инфраструктуру, включая DDoS, ботнеты и сетевую разведку. Согласно отчету Nozomi Networks Labs, в первой половине 2025 г. вредоносная активность в отношении критической инфраструктуры значительно возросла, при этом преобладают brute-force атаки и эксплуатация устаревших уязвимостей [3].

Эффективная защита таких сетей невозможна без современных систем обнаружения вторжений (Intrusion Detection System – IDS), использующих алгоритмы машинного обучения для идентификации аномального поведения. Однако на практике разработчики IDS сталкиваются с фундаментальной проблемой дефицита и дисбаланса данных (class imbalance). В реальных условиях эксплуатации доля атакующего трафика ничтожно мала по сравнению с фоновым (нормальным трафиком), что приводит к дисбалансу классов в обучающих выборках и, как следствие, к деградации классификаторов – модели склонны игнорировать минорные классы, что недопустимо для систем безопасности.

Большинство открытых наборов данных для обучения IDS, таких как актуальный CIC-IoT-2023 [4, 5], характеризуются высокой плотностью вредоносного трафика, что упрощает задачу обучения, но не соответствует реальным условиям функционирования IoT-сетей. Для имитации эксплуатации системы в реальной среде необходимо создание условий критического дисбаланса путем искусственного увеличения минорных классов. В таких сценариях традиционные методы борьбы с дисбалансом, такие как случайное дублирование (ROS) или синтетическая генерация на основе ближайших соседей (SMOTE [6]), часто оказываются недостаточно эффективными для восстановления сложной структуры NetFlow-данных.

Перспективным решением является применение генеративно-состязательных сетей (Generative Adversarial Networks – GAN), способных моделировать многомерные распределения признаков. Приведены результаты исследования эффективности различных архитектур GAN (включая CopulaGAN, CTGAN [7] и авторские модификации) для синтеза табличных данных IoT-трафика в условиях моделируемого критического дефицита. Представлен сравнительный анализ влияния аугментации на качество классификации при различных параметрах дисбаланса, объема данных и интенсивности синтеза.

Методы борьбы с дисбалансом можно разделить на алгоритмические (настройка весов и гиперпараметров) и методы уровня данных (пересемплирование). В данной работе акцент сделан на модификации данных, так как этот подход является универсальным и позволяет подготовить качественный обучающий набор, пригодный для использования с любыми архитектурами классификаторов без необходимости их специфической настройки под каждый тип атаки.

2. МЕТОДЫ

В качестве исходного набора данных использовался набор CIC-IoT-2023. В нем данные представлены в виде потоков трафика (формат netflow). Набор данных содержит восемь классов: семь вредоносных и один легитимный.

Параметры аугментации. В качестве параметров, контролирующих условия аугментации, выбраны:

1. Объем данных N_B – количество экземпляров в легитимном классе.

2. Степень дисбаланса (ir – imbalance ratio):

$$ir = \frac{N_M}{N_B},$$

где N_M – количество экземпляров в одном вредоносном классе обучающего (несбалансированного) набора данных.

3. Интенсивность аугментации (tr – target ratio):

$$tr = \frac{\tilde{N}_M}{N_B},$$

где $\tilde{N}_M$ – количество экземпляров в одном вредоносном классе после аугментации (в аугментированном наборе данных).

Наборы параметров аугментации разделены на четыре группы:

1. Основной сценарий. Сильный дефицит и дисбаланс данных. Параметры:

• N_B (объем данных) ∈ {5000, 10000, 20000};

• ir (степень дисбаланса) ∈ {0,0075; 0,015};

• tr (интенсивность аугментации) ∈ {0,05; 0,1; 0,15; 0,2}.

2. Высокая интенсивность аугментации ($tr > 0{,}25$).

3. Слабый дисбаланс ($ir > 0{,}05$).

4. Большой объем данных ($N_B > 30000$).

Наборы данных. Для проведения экспериментов сформированы три выборки (без учета наборов данных для аугментации):

1. Обучающий (несбалансированный) набор данных. Используется для обучения моделей аугментации, а также при классификации для получения «базового» результата в условиях дисбаланса.

2. Дополнительный набор данных. Содержит реальные данные в таком же количестве, в котором будут генерироваться синтетические данные. Используется для сравнения прироста качества классификации при добавлении синтетических и реальных данных в несбалансированный набор.

3. Тестовый набор данных. Предназначен для сбора метрик результатов классификации.

Схема формирования наборов данных представлена на рис. 1.

Методы аугментации. Использовались следующие методы аугментации:

• традиционные: SMOTE, SMOTE-Tomek, случайное пересемлирование, GaussianCopula;

• GAN: CTGAN [7], CTAB-GAN+ [8], MC-WGAN-GP [9], TMG-GAN [10], CopulaGAN [7];

• TVAE [7], используется в качестве генеративной альтернативы GAN.

Использованы реализации CTGAN, TVAE и CopulaGAN из библиотеки SDV [11–13].

Изначально в группу традиционных методов аугментации планировалось вклю-

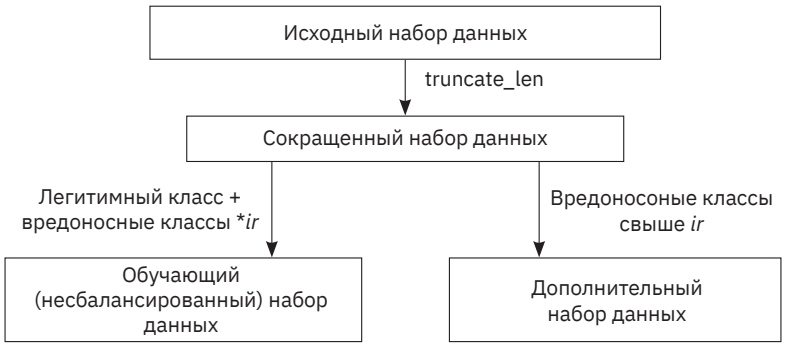


Рис. 1 | Схема формирования наборов данных

Fig. 1 | Data set formation scheme

чить также алгоритм ADASYN (Adaptive Synthetic Sampling). Однако в ходе экспериментов выявлена его принципиальная неприменимость к условиям, наиболее характерным для реальных систем обнаружения вторжений в сетях Интернета вещей.

Модификации моделей. Оригинальные архитектуры MC-WGAN-GP и TMG-GAN обладают узкой специализацией: первая ориентирована исключительно на категориальные признаки, вторая – на непрерывные численные величины. Для преодоления этого ограничения реализована модификация выходного слоя генератора на основе многопоточной архитектуры (multi-head architecture). В рамках данного подхода для каждого атрибута формируется отдельный выходной слой с семантически зависимой функцией активации: Softmax или Gumbel-Softmax для дискретных категорий и сигмоидальная или тождественная функция для аппроксимации численных значений.

Классификаторы. В качестве классификаторов использовались ансамблевые модели: LightGBM [14], RandomForest [15] и XGBoost [16]. При обучении использовались гиперпараметры по умолчанию, кроме параметра n_estimators, для которого задано значение 200 для каждого классификатора.

Работа реализованного прототип включает три этапа:

1. Подготовка несбалансированного набора данных для обучения (с учетом объема данных N_B и степени дисбаланса ir). Сначала количество экземпляров в каждом классе равно N_B, затем количество вредоносных экземпляров уменьшается до значения $N_B ir$.

2. Аугментация данных различными методами (с учетом интенсивности аугментации tr). После аугментации сохраняется аугментированный набор для каждого метода аугментации.

3. Классификация. Обучение классификаторов проводится: на исходных несбалансированных данных; аугментированных данных (10 наборов данных, один для каждого метода аугментации); дополненных данных (в несбалансированный набор добавлено такое количество экземпляров вредоносных классов, чтобы достичь показателя tr).

Схема программного прототипа представлена на рис. 2.

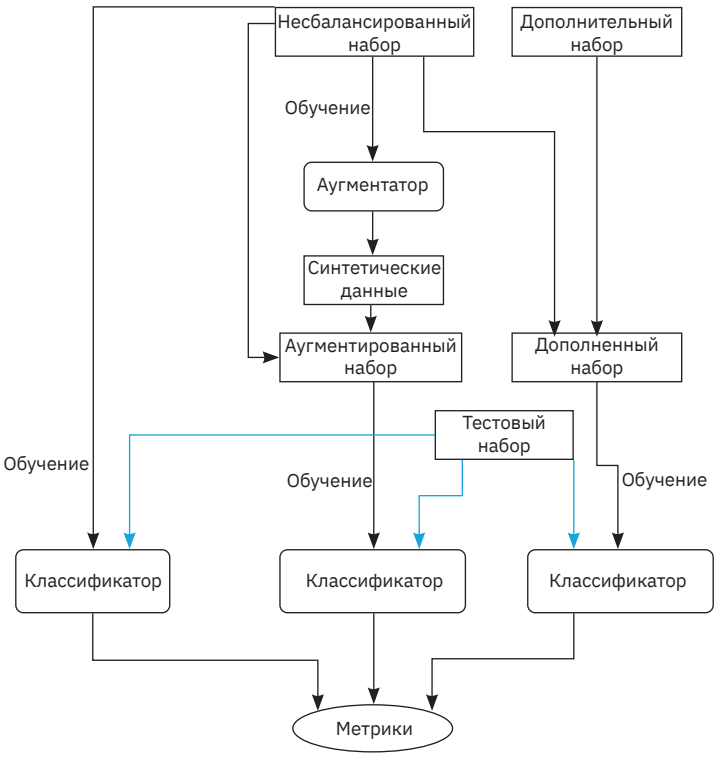


Рис. 2 | Схема разработанного прототипа

Fig. 2 | Diagram of the developed prototype

Метрика. Для оценки качества классификации использовалась макроусредненная F1-мера (далее F1-macro). Она подходит для оценки качества классификации в условиях дисбаланса данных, так как каждый класс имеет одинаковый вес. Прирост F1-macro далее будет указываться в процентах, величина абсолютная.

3. ОСНОВНЫЕ РЕЗУЛЬТАТЫ

В табл. 1 приведены метрики качества классификации в зависимости от метода аугментации, а также результаты при дополнении данных реальными данными. Сравнительный анализ различных генеративных подходов позволил установить преимущество архитектуры CopulaGAN, которая по ключевым метрикам превзошла альтернативную модель на основе вариационных автокодировщиков (TVAE). Тем не менее в ходе экспериментов зафиксировано, что классические методы пересемплирования, такие как SMOTE, SMOTE-Tomek и случайное пересемплирование, демонстрируют более высокие показатели эффективности, чем CopulaGAN.

Данный результат находит теоретическое обоснование в специфике обучения глубоких нейронных сетей: для адекватной настройки параметров GAN-моделей требуется значительный объем репрезентативной выборки минорных классов. В то же время алгоритм SMOTE, основанный на линейной интерполяции признакового пространства, сохраняет устойчивость в условиях экстремального дефицита данных. Относительно низкие показатели GAN-архитектур на табличных данных формата NetFlow объясняются тем, что структура последних зачастую характеризуется менее сложными нелинейными зависимостями по сравнению с изображениями или текстами. Это подтверждает гипотезу

Таблица 1 | Сравнение методов аугментации

Table 1 | Comparison of augmentation methods

Метод	Средний ΔF1-macro, %	Медиана ΔF1-macro, %	Win-rate (улучшение), %	Максимум, %	Минимум, %
Дополненные данные	+14,0	+11,5	100	+84,2	+0,8
SMOTE	+5,6	+2,6	92	+80,8	–2,7
SMOTETomek	+5,5	+2,3	93	+80,4	–2,7
Случайное пересемплирование	+4,9	+2,3	85	+81,2	–10,2
CopulaGAN	+4,2	+1,1	72	+79,3	–5,5
GaussianCopula	+3,9	+0,9	76	+80,2	–3,3
CTGAN	+2,8	+0,2	54	+77,7	–40,2
TVAE	+2,5	+0,2	54	+80,2	–9,2
MCWGANGP	+1,6	–0,9	23	+75,0	–6,6
CTABPlus	+1,4	–1,0	21	+74,5	–7,0
TMGGAN	+1,2	–0,7	29	+77,7	–13,7

Примечания: Win-rate – это доля экспериментов, в которых модель, обученная на аугментированных данным методом данных, превзошла по метрике F1-macro контрольную модель, обученную на исходных (несбалансированных) данных. Максимум – максимальный прирост F1-macro среди всех экспериментов. Минимум – минимальный прирост F1-macro.

о том, что превосходство состязательных сетей над статистическими методами проявляется лишь при достижении определенного порога объема и сложности входных данных.

Контрольный эксперимент с добавлением реальных данных показал предсказуемо высокий результат (ΔF1-macro = 14 %), превзошедший все методы синтеза. Однако сложность масштабирования стендов и трудоемкость экспертной разметки трафика в реальных IoT-инфраструктурах делают методы генеративной аугментации (в частности, CopulaGAN) более приоритетными для оперативного обучения IDS в условиях априорного дефицита информации об атаках.

Анализ эффективности аугментации в зависимости от условий эксперимента (табл. 2) позволил выявить ряд ключевых закономерностей. В качестве репрезентативной модели для оценки влияния параметров выбран CopulaGAN, как показавший наиболее стабильные результаты среди методов аугментации с использованием GAN.

На основе полученных данных (табл. 2) можно сделать следующие выводы:

1. Влияние интенсивности синтеза. Сравнение основного сценария (строка 1) и сценария с высокой интенсивностью аугментации (строка 2) показывает, что избыточная генерация синтетических данных ($tr > 0{,}25$) приводит к снижению эффективности классификации. Прирост F1-macro падает с 4,8 до 2,2 %, что свидетельствует о накоплении «статистического шума» и переобучении моделей на искусственных паттернах.

2. Порог целесообразности. В условиях слабого дисбаланса (строка 3, $tr > 0{,}05$) применение аугментации нецелесообразно: зафиксировано незначительное снижение качества (–0,1 %). Это подтверждает гипотезу о том, что современные ансамблевые классификаторы обладают достаточной внутренней устойчивостью к умеренному дисбалансу классов.

3. Эффект объема данных. Особый интерес представляет сравнение групп 1 и 4. Увеличение объема исходной несбалансированной выборки само по себе не ведет к росту качества (показатель в столбце «Дисбаланс» остается на уровне 0,7). Однако именно на больших объемах данных аугментация демонстрирует максимальную эффективность (ΔF1-macro = +9,1 %). Это объясняется тем, что увеличение обучающей выборки для генератора позволяет GAN-модели точнее аппроксимировать распределение признаков, минимизируя ошибки при синтезе минорных классов.

Оценка чувствительности различных алгоритмов машинного обучения к аугментации методом CopulaGAN (табл. 3, 4) выявила значительные различия в их архитектурной устойчивости.

Основные выводы:

1. Random Forest демонстрирует исходную устойчивость к дисбалансу благодаря механизму бэггинга. Дополнительная аугментация не приводит к росту целевой метрики, а в ряде случаев вызывает незначительную деградацию (–1,6 %).

2. Градиентный бустинг (XGBoost, LightGBM): в условиях дефицита данных модели склонны к переобучению, что критично

Таблица 2 | Изменение качества классификации (ΔF1-macro) по группам параметров

Table 2 | Change in classification quality (ΔF1-macro) by parameter groups

Номер строки	Группа параметров	Дисбаланс	Δ CopulaGAN, %
1	Основной сценарий	0,69	+4,8
2	Высокая интенсивность аугментации	0,7	+2,2
3	Слабый дисбаланс	0,84	–0,1
4	Большой объем данных	0,7	+9,1

Таблица 3 | Результаты классификаторов при использовании набора параметров «Основной сценарий»

Table 3 | Classifier results when using the "Main Scenario" parameter set

Классификатор	Дисбаланс	ΔCopulaGAN, %	CopulaGAN
RandomForest	0,69	–0,1	0,69
XGBoost	0,73	+2,4	0,75
LightGBM	0,65	+12,2	0,77

Таблица 4 | Результаты классификаторов при использовании набора параметров «Большой объем данных»

Table 4 | Classifier results when using a set of "Large data volume" parameters

Классификатор	Дисбаланс	ΔCopulaGAN, %	CopulaGAN
RandomForest	0,76	–1,6	0,74
XGBoost	0,79	+3,3	0,82
LightGBM	0,57	+25,7	0,83

Таблица 5 | F1-мера каждого класса после аугментации

Table 5 | F1 is the measure of each class after augmentation

Класс	Дисбаланс	CopulaGAN
Легитимный	0,54	0,62
DDoS	0,76	0,99
DoS	0,83	0,99
Mirai	0,9	0,99
Recon	0,53	0,71
Spoofing	0,52	0,76
BruteForce	0,5	0,69
Web	0,42	0,6

для несбалансированных выборок. До аугментации LightGBM показывает наихудшую устойчивость.

3. Синергетический эффект: применение CopulaGAN наиболее эффективно для LightGBM, обеспечивая максимальный прирост (ΔF1-macro до +25,7 %). После аугментации LightGBM превосходит остальные модели, что делает связку CopulaGAN + LightGBM наиболее эффективным решением для задач IDS в условиях экстремального дисбаланса.

При использовании связки LightGBM и CopulaGAN для наборов параметров «Основной сценарий» и «Большой объем данных» F1-мера каждого класса растет после аугментации (табл. 5).

Наибольшая эффективность предложенного подхода зафиксирована в сценарии с максимальным объемом выборки и критическим уровнем дисбаланса ($N_B = 8000$, $ir = 0{,}0075$, $tr = 0{,}05$). В данных условиях классификатор LightGBM без предварительной аугментации находился

в вырожденном состоянии, демонстрируя F1-macro = 0,03 (фактическое игнорирование минорных классов). Применение CopulaGAN позволило восстановить функциональность модели, подняв значение целевой метрики до 0,81 (абсолютный прирост 0,78). Данный результат доказывает, что при достижении определенного порога объема данных, нейросетевая аугментация способна полностью компенсировать структурные недостатки классификатора, переводя его из состояния неработоспособности в режим эффективного обнаружения атак.

4. ЗАКЛЮЧЕНИЕ

В результате проведенного исследования эффективности генеративно-состязательных сетей для аугментации IoT-трафика сделаны следующие выводы:

1. Оптимальные архитектуры. Установлено, что для табличных данных формата NetFlow наиболее эффективной является модель CopulaGAN, превзошедшая другие нейросетевые подходы (TVAE, CTGAN). При этом выявлено, что классические методы (SMOTE, ROS) сохраняют преимущество на сверхмалых выборках из-за меньшей требовательности к объему обучающих данных.

2. Границы применимости. Экспериментально определен «порог целесообразности» аугментации: она критически важна при высоком дисбалансе ($ir \leq 0{,}015$) и умеренной интенсивности синтеза ($tr \leq 0{,}25$). Показано, что увеличение объема исходных данных само по себе не решает проблему дисбаланса, но создает необходимую базу для качественного обучения GAN-генератора.

3. Восстановление работоспособности моделей. Выявлен значительный синергетический эффект при использовании связки CopulaGAN + LightGBM. В условиях экстремального дефицита данных аугментация позволила восстановить функциональность классификатора, подняв показатель F1-macro с вырожденного уровня 0,03 до эксплуатационного значения 0,81.

4. Практическая значимость. Предложенная методика нейросетевой аугментации позволяет эффективно обучать системы обнаружения вторжений (IDS) в условиях невозможности сбора реального вредоносного трафика, обеспечивая надежную классификацию за счет переноса «бремени дисбаланса» с модели классификатора на этап подготовки данных.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. State of IoT 2025: Number of connected IoT devices growing 14% to 21.1 billion globally. URL: https://iot-analytics.com/number-connected-iot-devices/ (дата обращения: 01.02.2026).
2. IoT Hacking Statistics 2025: The Definitive Report on Threats, Risks & Regulations. URL: https://deepstrike.io/blog/iot-hacking-statistics (дата обращения: 01.02.2026).
3. Nozomi Networks Labs. OT/IoT Cybersecurity Trends & Insights 2025. URL: https://www.nozominetworks.com/ot-iot-cybersecurity-trends-insights-february-2025 (дата обращения: 01.02.2026).
4. Canadian Institute for Cybersecurity. CIC IoT dataset 2023. URL: https://www.unb.ca/cic/datasets/iotdataset-2023.html (дата обращения: 01.02.2026).
5. Kaggle. UNB CIC IOT 2023 Dataset. URL: https://www.kaggle.com/datasets/madhavmalhotra/unb-cic-iot-dataset (дата обращения: 01.02.2026).
6. imbalanced-learn. SMOTE. URL: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html (дата обращения: 01.02.2026).
7. **Xu L., Skoularidou M., Cuesta-Infante A., et al**. Modeling tabular data using conditional

GAN // Advances in neural information processing systems. 2019. Vol. 32. arXiv: 1907.00503.
8. **Zhao Z., Kunar A., Birke R., Chen L. Y.** CTAB-GAN+: Enhancing tabular data synthesis // Frontiers in Big Data. 2024. Vol. 6. P. 1296508.
9. **Camino R., Hammerschmidt C., State R.** Generating multi-categorical samples with generative adversarial networks // arXiv preprint arXiv: 1807.01202. 2018.
10. **Hongwei Ding, Yu Sun, Nana Huang, Xiaohui Cui.** TMG-GAN: Generative adversarial networks-based imbalanced learning for network intrusion detection // IEEE Transactions on Information Forensics and Security. 2023. Vol. 19. P. 1156–1167. DOI: 10.1109/TIFS.2023.3331240
11. Synthetic Data Vault. CopulaGANSynthesizer. URL: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/copulagansynthesizer (дата обращения: 01.02.2026).
12. Synthetic Data Vault. CTGANSynthesizer. URL: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/ctgansynthesizer (дата обращения: 01.02.2026).
13. Synthetic Data Vault. TVAESynthesizer. URL: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/tvaesynthesizer (дата обращения: 01.02.2026).
14. LightGBM's documentation. URL: https://lightgbm.readthedocs.io/en/stable/ (дата обращения: 01.02.2026).
15. scikit-learn. RandomForestClassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (дата обращения: 01.02.2026).
16. XGBoost Documentation. URL: https://xgboost.readthedocs.io/en/stable/ (дата обращения: 01.02.2026).

REFERENCES

1. State of IoT 2025: Number of connected IoT devices growing 14% to 21.1 billion globally. URL: https://iot-analytics.com/number-connected-iot-devices/ (accessed: 01.02.2026).
2. IoT Hacking Statistics 2025: The Definitive Report on Threats, Risks & Regulations. URL: https://deepstrike.io/blog/iot-hacking-statistics (accessed: 01.02.2026).
3. Nozomi Networks Labs. OT/IoT Cybersecurity Trends & Insights 2025. URL: https://www.nozominetworks.com/ot-iot-cybersecurity-trends-insights-february-2025 (accessed: 01.02.2026).
4. Canadian Institute for Cybersecurity. CIC IoT dataset 2023. URL: https://www.unb.ca/cic/datasets/iotdataset-2023.html (accessed: 01.02.2026).
5. Kaggle. UNB CIC IOT 2023 Dataset. URL: https://www.kaggle.com/datasets/madhavmalhotra/unb-cic-iot-dataset (accessed: 01.02.2026).
6. imbalanced-learn. SMOTE. URL: https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html (accessed: 01.02.2026).
7. **Xu L., Skoularidou M., Cuesta-Infante A., et al**. Modeling tabular data using conditional GAN. *Advances in neural information processing systems*. 2019. Vol. 32. arXiv: 1907.00503.
8. **Zhao Z., Kunar A., Birke R., Chen L. Y.** CTAB-GAN+: Enhancing tabular data synthesis. *Frontiers in Big Data*. 2024. Vol. 6, pp. 1296508.
9. **Camino R., Hammerschmidt C., State R.** Generating multi-categorical samples with generative adversarial networks. *arXiv preprint arXiv: 1807.01202*. 2018.
10. **Hongwei Ding, Yu Sun, Nana Huang, Xiaohui Cui.** TMG-GAN: Generative adversarial networks-based imbalanced learning for network intrusion detection. *IEEE Transactions on Information Forensics and Security*. 2023. Vol. 19, pp. 1156–1167. DOI: 10.1109/TIFS.2023.3331240
11. Synthetic Data Vault. CopulaGANSynthesizer. URL: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/copulagansynthesizer (accessed: 01.02.2026).
12. Synthetic Data Vault. CTGANSynthesizer. URL: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/ctgansynthesizer (accessed: 01.02.2026).
13. Synthetic Data Vault. TVAESynthesizer. URL: https://docs.sdv.dev/sdv/single-table-data/modeling/synthesizers/tvaesynthesizer (accessed: 01.02.2026).

14. LightGBM's documentation. URL: https://lightgbm.readthedocs.io/en/stable/ (accessed: 01.02.2026).
15. scikit-learn. RandomForestClassifier. URL: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed: 01.02.2026).
16. XGBoost Documentation. URL: https://xgboost.readthedocs.io/en/stable/ (accessed: 01.02.2026).

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ПЛАТОНОВ Владимир Владимирович – канд. техн. наук, доцент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: plato@ibks.spbstu.ru
ORCID: 0000-0002-9899-2778

PLATONOV Vladimir V. – Candidate of Engineering Sciences, Associate Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

СКИБА Дарослав Александрович – программист, АО «ИнфоТеКС», Россия, 125167, Москва, ул. Викторенко, д. 9, стр. 1
E-mail: daroslav.skiba@yandex.ru
ORCID: 0009-0004-9032-4961

SKIBA Daroslav A. – Programmer, JSC "InfoTeX", Russia, 125167, Moscow, Viktorenko str., 9, build.1

Научная статья
DOI 10.66424/2071-8217-2026-2-8
УДК 004.056

РАСПОЗНАВАНИЕ НАЧАЛ ФУНКЦИЙ В БИНАРНЫХ ФАЙЛАХ С ИСПОЛЬЗОВАНИЕМ РЕКУРРЕНТНЫХ НЕЙРОННЫХ СЕТЕЙ

А. С. Шайханов*

Московский государственный технический университет имени Н. Э. Баумана, Москва, Россия

✉ *artem.shaykhanov@gmail.com

ДЛЯ ЦИТИРОВАНИЯ

Шайханов А. С. Распознавание начал функций в бинарных файлах с использованием рекуррентных нейронных сетей // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 92–112. DOI: 10.66424/2071-8217-2026-2-8

ПОСТУПИЛА 01.04.2026
ПРИНЯТА 04.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Рассматривается задача распознавания начал функций в бинарных файлах, которая является одной из ключевых подзадач реверс-инжиниринга программного обеспечения. В качестве подхода к решению задачи предложено использование рекуррентной нейронной сети, обрабатывающей последовательности байтов бинарного файла. Проведен сравнительный анализ существующих нейросетевых моделей распознавания функций, выявлены их преимущества и ограничения, что позволило обосновать выбор простой и воспроизводимой архитектуры RNN. Получены результаты, на основании которых сделаны выводы о влиянии гиперпараметров модели, соответствующих особенностям машинной архитектуры и форматов бинарных файлов, на качество распознавания модели. Эксперименты выполнены на бинарных файлах микроконтроллеров ESP32 архитектуры Xtensa Little Endian и STM32WBA6 с ядром Cortex-M33 архитектуры ARMv8-M с использованием как стандартного, так и случайного выравниваний, что также позволило оценить устойчивость модели к изменению структуры бинарных данных. На основе разработанной модели реализовано расширение для дизассемблера IDA Pro, демонстрирующее практическую применимость предложенного подхода в реальных задачах реверс-инжиниринга.

КЛЮЧЕВЫЕ СЛОВА

Реверс-инжиниринг, распознавание, бинарный файл, начало функции, рекуррентная нейронная сеть, IDA Pro

Original article
DOI 10.66424/2071-8217-2026-2-8

RECOGNIZING FUNCTION PROLOGUES IN BINARY FILES WITH RECURRENT NEURAL NETWORKS

A. S. Shaikhanov*

Bauman Moscow State Technical University, Moscow, Russia

✉ *artem.shaykhanov@gmail.com

FOR CITATION

Shaikhanov A. S. Recognizing function prologues in binary files with

ABSTRACT

The article discusses the problem of recognising function prologues in binary files, which is one of the key subtasks of software reverse engi-

recurrent neural networks. *Problems of information security. Computer systems*. 2026. No. 2, pp. 92–112.
DOI: 10.66424/2071-8217-2026-2-8
(In Russian)

RECEIVED 01.04.2026
ACCEPTED 04.05.2026
PUBLICATION 15.06.2026

neering. The proposed approach is to use a recurrent neural network that processes byte sequences of a binary file. A comparative analysis of existing neural network models for function recognition was conducted, and their advantages and limitations were identified, which made it possible to justify the choice of a simple and reproducible RNN architecture. The obtained results allow to make conclusions about the influence of model hyperparameters on the quality of model recognition. These hyperparameters correspond to the features of the machine architecture and binary file formats. The experiments were performed on binary files of the ESP32 microcontroller with Xtensa Little Endian architecture and STM-32WBA6 microcontroller of Cortex-M33 core with ARMv8-M architecture using both standard and random alignment, which made it possible to evaluate the model's resistance to changes in the structure of binary data. Based on the developed model, an extension for the IDA Pro disassembler has been implemented, demonstrating the practical applicability of the proposed approach in real reverse engineering tasks.

KEYWORDS

Reverse engineering, recognition, binary file, function prologue, recurrent neural network, IDA Pro

1. ВВЕДЕНИЕ

Реверс-инжиниринг программного обеспечения – процесс исследования файлов этого программного обеспечения с целью восстановления принципов работы для выполнения различных задач, например, поиска недекларированных возможностей. Реверс-инжиниринг почти всегда включает в себя анализ исполняемого кода внутри бинарных файлов, представленного в виде машинных инструкций. Одной из подзадач анализа кода является распознавание начал функций внутри бинарного файла.

Реверс-инженеры используют различные инструменты, автоматизирующие процесс распознавания начал функций. Этими инструментами могут быть простые скрипты, написанные исследователем самостоятельно, бесплатные утилиты, разработанные сообществом, или специализированное коммерческое программное обеспечение, решающее сразу несколько задач обратной разработки помимо распознавания функций. Однако большая часть этих инструментов не применяет нейронные сети, а использует детерминированные методы, основанные на эвристиках, сигнатурном анализе, построении графа потока управления. В то же время существующие инструменты, основанные на нейронных сетях, имеют множество недостатков, которые усложняют постоянное использование этих инструментов для решения широкого круга задач.

Цель исследования – частичная автоматизация решения задачи распознавания начал функций внутри бинарного файла посредством разработки прикладного инструмента реверс-инжиниринга. Данный инструмент основан на модели нейронной сети. Важной особенностью исследования также является то, что при разработке этого инструмента обязательно должна учитываться прикладная специфика задач обратной разработки.

2. МЕТОДЫ

Существуют научные исследования, решающие задачи реверс-инжиниринга посредством машинного обучения. К таким задачам можно отнести:

- выделение в бинарном файле секций кода и данных [1, 2];
- формирование корректного дизассемблерного представления бинарного кода [3–5];
- восстановление имен функций в бинарный файлах, не содержащих отладочную информацию [6–8];

• распознавание границ функций внутри бинарного файла [3, 4, 9–11];

• восстановление типов данных и аргументов функций внутри бинарного файла [12];

• восстановление имен переменных в декомпилированном представлении функций бинарного файла [13];

• поиск функций, семантически схожих между собой, в нескольких бинарных файлах [14, 15];

• сравнение разных версий бинарного файла (бинарный диффинг) [16];

• создание семантических векторных представлений инструкций ассемблера [17];

• создание больших наборов с собранными бинарными файлами для обучения моделей [18, 19].

Предложенные решения демонстрируют хорошие показатели оценивающих метрик в сравнении с традиционными методами реверс-инжиниринга. Конкретно для задачи распознавания начал функций этими традиционными методами являются различные эвристики, сигнатурный анализ, построение графа потока управления и др. Часть этих методов реализована в популярных инструментах реверс-инжиниринга – IDA Pro и Ghidra [20–22].

Опишем существующие модели нейронных сетей, распознающих начала функций внутри бинарных файлов, а также выделим преимущества и недостатки этих моделей.

RNN. В 2015 г. представлена модель, использующая двунаправленную рекуррентную нейронную сеть [10]. На вход модели подавались последовательности фиксированной длины в 1000 байт. Байты последовательности конвертировались в one-hot вектора, после чего передавались на вход рекуррентному слою. Модель имела один двунаправленный рекуррентный слой из 16 нейронов. Выход рекуррентного слоя конкатенировался для прямого и обратного направлений и подавался в softmax для создания распределения вероятностей соответствия байта началу (концу) функции.

Авторы модели утверждают, что они использовали две независимые одинаковые модели для решения двух независимых задач поиска начал и концов функций. Далее результат работы моделей объединялся посредством простой эвристики для получения точных границ функции. Многие авторы моделей, представленных после 2015 г., использовали показатели метрик качества работы RNN для сравнения.

Преимущества модели:

• высокая скорость обучения, обусловленная простотой работы модели;

• высокая скорость работы, согласно оценкам авторов статьи «Padding Matters»;

• высокие показатели метрик для Executable and Linkable Format (ELF) и Portable Executable (PE) форматов файлов, собранных под архитектуры x86 и x86–64 с разными уровнями оптимизации;

• простота реализации предполагает возможность воспроизвести архитектуру модели, а далее обучить модель для нужной машинной архитектуры.

Недостатки: отсутствие программной реализации от авторов статьи в открытом доступе; значительное снижение показателей качества модели для x86 и x86–64 при заполнении выравниваний случайными байтами.

Именно эта модель взята автором данной работы за основу для решения задачи распознавания начал функций.

XDA. В 2020 г. представлена модель, основанная на переносе различных контекстных зависимостей машинного кода для решения задач обратного проектирования, в том числе и для задачи распознавания начал функций [3].

Работа модели делится на две части:

• Предобучение модели посредством маскированного языкового моделирования (MLM) для выявления контекстных зависимостей машинного кода. Задача MLM – обучить модель предсказывать случайно замаскированные байты исходя из окружающего контекста. По словам авторов XDA, это повышает уровень понимания моделью машинного кода и устойчивость модели к различным компиляторам и уровням оптимизации.

• Обучение модели для распознавания функций. Авторы использовали слои внутреннего внимания для вычисления потока информации между каждой парой байтов. Это необходимо для установления зависимостей между удаленным байтами

тела функции, что повышает качество восстановления границ функций.

Преимущества модели:

• наличие исходного кода модели и уже предобученных весов для некоторых архитектур в открытом доступе, что позволяет настроить или предварительно обучить модель для нужной машинной архитектуры;

• возможность использования предобученной модели для решения других задач обратной разработки;

• высокие показатели метрик для ELF и PE форматов файлов, собранных под архитектуры x86 и x86–64 с разными уровнями оптимизации.

Недостатки: низкая скорость обучения из-за сложной архитектуры; низкая скорость работы согласно оценкам авторов DeepDi и Padding Matters; значительное снижение показателей качества модели для x86 и x86–64 при заполнении выравниваний случайными байтами.

DeepDi. В 2022 г. представлена модель, работающая не с байтами, а с машинными инструкциями. Для первичной разметки инструкций в DeepDi строится супермножество всевозможных инструкций. На основе этого супермножества строится граф потока инструкций. Графовая сверточная сеть анализирует этот граф для определения реальных инструкций и их связей между собой [4].

Для поиска начал функций DeepDi использует:

• Набор эвристик, применяемых к размеченным инструкциям для определения кандидатов на начало функции. По словам авторов, данный подход не только снижает количество ложных срабатываний, но и существенно сокращает входное количество байтов, подаваемое на вход нейронной сети.

• Нейронную сеть со слоем встраивания, рекуррентным слоем и двухслойным персептроном с классификатором.

Преимущества модели:

• наличие модели, собранной в динамическую библиотеку с программным интерфейсом для доступа к функциям восстановления границ;

• самая высокая скорость работы среди всех моделей согласно оценкам авторов статьи «Padding Matters»;

• самые высокие показатели метрик для ELF и PE форматов файлов, собранных под архитектуры x86 и x86–64 с разными уровнями оптимизации согласно оценкам авторов статьи «Padding Matters»;

• наименьшее снижение показателей качества модели для x86 и x86–64 при заполнении выравниваний случайными байтами согласно оценкам авторов статьи «Padding Matters».

Недостатки: отсутствие программной реализации модели, что обусловлено тем, что разработанный DeepDi принадлежит коммерческой компании DeepBits; ограниченный набор архитектур – x86 и x86–64.

Выводы о существующих моделях. RNN, XDA, DeepDi обучены на больших наборах обучающих данных. По словам авторов моделей, каждая из них в свое время демонстрировала хорошие показатели оценивающих метрик. Все они в каком-то виде имеют прикладные реализации, которые можно использовать для решения собственных задач.

Еще одной важной отличительной особенностью предыдущих исследований является то, что задачу распознавания функций исследователи рассматривали как «дополнительную». То есть авторами работ предприняты попытки создать универсальный инструмент бинарного анализа, который мог бы не только восстановить границы функций, но и разметить ассемблерные инструкции, классифицировать вредоносный бинарный файл. Это также стало причиной усложнения архитектур самих моделей с течением времени.

Тем не менее все эти модели имеют недостатки. Основной недостаток – обучение и оценка на бинарных файлах, собранных под ограниченный список архитектур (x86 и x86–64), что вызвано наличием уже готовых емких наборов данных для обучения. Эти наборы включают в себя PE и ELF файлы, собранные под Windows и Linux разными компиляторами и с разными уровнями оптимизациями машинного кода.

Таким образом, задачу распознавания функций необходимо рассмотреть для других машинных архитектур и инструментов сборки. При построении модели помимо ее метрик качества необходимо

также учитывать особенности прикладных задач по распознаванию функций. В данном случае имеется в виду ограниченное время исследователя, применяющего модель, и малый набор обучающих данный.

Постановка задачи и оценка качества решения. Входные данные B^F представляют собой последовательность байтов B_1, B_2, …, B_{F-1} некоторого бинарного файла размера F, где $B_i \in \mathbb{Z}_{256}$ – байт бинарного файла со смещением i.

Выходные данные O^F представляют собой последовательность признаков начала функции O_1, O_2, …, O_{F-1}, где $O_i \in \mathbb{Z}_2$. При этом $O_i=1 \Rightarrow$ байт B_i распознан как байт начала функции, в противном случае байт B_i не обладает признаком «начала функции». Количество ненулевых элементов в $\{O_1, O_2, \ldots, O_{F-1}\}$ – общее количество распознанных функций внутри бинарного файла.

Таким образом, рассматриваемая задача распознавания начал функций представляет собой задачу бинарной классификации. Традиционными метриками оценки качества бинарной классификации являются: Accuracy, Precision, Recall и F1 метрика:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}};$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}};$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}};$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}},$$

где TP (True Positive) – количество случаев, когда модель верно распознала байт начала функции; TN (True Negative) – количество случаев, когда модель верно распознала байт, не являющийся байтом начала функции; FP (False Positive) – ошибки первого рода (количество случаев, когда модель распознала байт, не являющийся байтом начала функции, как байт начала функции); FN (False Negative) – ошибки второго рода (количество случаев, когда модель распознала байт начала функции, как байт, не являющийся началом функции).

При выборе метрики оценки качества учитывалось то, что любой бинарный файл помимо секций с кодом функций будет содержать в себе секции с данными: строки, таблицы с адресами и т.д. В коде исполняемых функций всегда будет гораздо меньше байт, обладающих признаком «начала функции», чем байт, не являющихся началом функции. Все это в совокупности вносит сильный дисбаланс в обучающую и тестирующую выборки. Данная проблема отражается в следующем:

• Метрика Accuracy, представляющая собой отношение количества верно предсказанных ответов к общему количеству предсказанных ответов, не может быть использована для оценки качества работы модели. Тривиальный классификатор, определяющий любой байт последовательности как байт, не обладающий признаком «начала функции», будет демонстрировать высокие показатели данной метрики, что не является правильным.

• Недопустимо как уменьшение числа ошибок первого рода за счет существенного увеличения ошибок второго рода, так и обратное. Для универсальности оценки качества работы модели необходимо учитывать оба вида ошибок, поэтому отдельно взятый Precision (Recall) не может использоваться в качестве универсальной метрики.

Исходя из перечисленного, для оценки качества работы модели используется F1 метрика.

В предыдущих работах исследователи ставили перед собой разные задачи, смежные с восстановлением начал функций: восстановление конца функции, объединение начала и конца для восстановления границ всей функции с помощью каких-то дополнительных эвристик; восстановление инструкций ассемблера.

Перечисленные задачи не будут рассматриваться в данной работе из-за того, что встроенный функционал IDA Pro способен дизассемблировать инструкции и распознавать тело функции посредством анализа потока управления, если правильно предоставить ему начало функции.

Набор данных для обучения и оценки. Исходный набор данных состоял из

двух поднаборов – последовательностей байтов файлов программ микроконтроллера ESP32 архитектуры Xtensa Little Endian и последовательностей байтов файлов программ микроконтроллера STM32WBA6 с ядром Cortex-M33 архитектуры ARMv8-M. Каждый из поднаборов использовался для обучения и тестирования отдельной модели, что позволило получить независимые результаты для двух разных машинных архитектур.

Исходный код файлов программ микроконтроллера ESP32 разрабатывался на языке Rust с использованием C/C++ библиотек. Сборка файла проекта осуществлялась с использованием фреймворка сборки Rust проектов от Espressif Systems.

Исходный код файлов программ микроконтроллера STM32WBA6 разрабатывался на языках C/C++. Сборка файла проекта осуществлялась с использованием фреймворка сборки проектов от STMicroelectronics.

В предыдущих исследованиях авторы ограничили обучающую и тестирующую выборки. Они использовали секции с кодом, мотивируя это необходимостью минимизации дисбаланса классов. В данном исследовании для обучения и тестирования используются все секции бинарного файла, включающие в себя и данные, и код, и служебные заголовки. Это сделано исходя из следующей гипотезы: «Обучающая и тестирующая выборки должны максимально соответствовать реальным данным, на которых впоследствии будет применяться модель».

Все секции исходного бинарного файла считываются и разбиваются на последовательности фиксированной длины. Длина последовательности является гиперпараметром. Далее последовательности случайно перемешиваются между собой с фиксированным параметром рандомизации для воспроизводимости, после чего разбиваются на обучающую и тестирующую выборки. Для разбиения используется стандартное отношение 80 к 20 %.

Необходимо отметить, что идея применения моделей нейронных сетей XDA и RNN для распознавания начал функций в Rust бинарных файлах уже реализована в работе «RustBound: Function Boundary Detection over Rust Stripped Binaries». Авторы продемонстрировали, что от конфигураций сборки Rust проектов сильно зависят байты прологов и эпилогов функций в собираемом бинарном файле. Поэтому для формирования обучающего набора данных авторы собирали бинарные файлы с разными уровнями оптимизации. Данная работа имеет недостаток оригинальных работ XDA и RNN – обучение модели для архитектуры x86–64. При этом в своей работе авторы ограничились экспериментами с разными подходами к обучению моделей, не проделав никаких экспериментов по изменению гиперпараметров архитектур оригинальных моделей [23].

Выбор модели нейронной сети. За основу архитектуры нейронной сети взята RNN модель. Основные причины:

• Универсальность модели: минимальное количество дополнительных эвристик, зависящих от конкретной машинной архитектуры. Модель принимает на вход последовательности байт, а не инструкции, напрямую зависящие от конкретной архитектуры.

• Модели со сложной архитектурой требуют большего набора данных для обучения. Простые модели, наоборот, не требовательны к большим наборам обучающих данных. В связи с ограничениями количества входных данных, связанными с прикладными особенностями решаемой задачи, принято решение рассматривать простые архитектуры нейронных сетей.

• Высокая скорость обучения, обусловленная простотой архитектуры модели. Высокая скорость обучения позволяет проверить большее количество гипотез с изменением гиперпараметров модели для поиска их оптимальных значений. Скорость обучения также важна в случае, если независимый исследователь захочет переобучить модель со своими значениями гиперпараметров.

• Высокая скорость работы модели. Анализ байтов исследуемого бинарного файла посредством разработанного расширения для IDA Pro должен быстро выполняться, поскольку время исследователя на анализ одного бинарного файла может быть сильно ограничено. В статье

«Padding Matters» авторы измерили время работы моделей RNN, XDA и DeepDi, а также инструментов реверс-инжиниринга Ghidra и IDA Pro на файлах динамической библиотеки Chromium и файлах вредоносной программы-вымогателя Conti. Результаты измерений показали, что время работы DeepDi составило секунды, время работы RNN – минуты, а XDA, Ghidra и IDA Pro – часы. Поскольку использование модели DeepDi невозможно на бинарных файлах, собранных не под архитектуру x86, то RNN остается быстрейшим вариантом для распознавания начал функций из доступных.

Несмотря на все достоинства модели RNN, она нуждается в некоторых изменениях перед исследованием гиперпараметров.

В работе «Explaining deep learning based security applications» авторы предлагают модификации модели RNN от 2015 г. Основное изменение – уменьшение количества нейронов выходного слоя и изменение правила выбора класса [24].

Основные причины:

• В оригинальной версии модели выходной слой имел два нейрона, которые выдавали вероятности того, что байт является или не является началом функции. Функция выбора максимального значения из двух вероятностей определяла класс байта.

• В адаптированной версии модели выходной слой имеет всего один нейрон, выдающий одну вероятность. Далее эта вероятность сравнивается с пороговым значением для определения класса.

• Реализация с одним выходным нейроном имеет следующие преимущества:

– задача бинарной классификации требует всего одну вероятность (принадлежность байта к классам 1 и 0 – противоположные события);

– адаптивное пороговое значение позволяет балансировать метрики качества, по сути представляя собой еще один гиперпараметр.

Пороговое значение принимается равным 0,5. Авторы работы утверждают, что такие простые изменения позволили им повысить значения показателей метрик качества.

Формальное описание модели. Входными данными является последовательность байт B^F бинарного файла размера F, где $B_i \in \mathbb{Z}_{256}$. Эта последовательность разбивается на подпоследовательности длины L, а далее каждый байт B_i подпоследовательности посредством one-hot преобразования приводится к вектору действительных чисел, в котором элемент с индексом, равным значению байта B_i, является единичным, а все остальные нулевыми:

$$\text{one-hot} : \mathbb{Z}_{256}^{L} \to \mathbb{R}^{L \times 256}.$$

One-hot преобразование необходимо, поскольку в наборах инструкций микроконтроллеров байты с близкими значениями могут обладать совершенно различными семантическими свойствами.

После one-hot преобразования подпоследовательности передаются в двунаправленный рекуррентный слой размерности N. Под двунаправленностью в данном случае понимается использование двух независимых рекуррентных слоев, один из которых получает one-hot вектора в прямом порядке, а второй – в обратном. Выходы двух этих слоев объединяются для передачи в следующий слой. Формально рекуррентный слой можно определить так:

$$\text{Recurrent layer} : \mathbb{R}^{L \times 256} \times \mathbb{R}^{N} \to \mathbb{R}^{L \times 2N}.$$

При этом для каждого входного вектора $\text{one-hot}_j \in \mathbb{R}_{256}$, $j \in [0, L-1]$, подаваемого на вход рекуррентному слою прямого порядка, справедливо соотношение:

$$\begin{aligned} &\text{Forward layer}(\text{one-hot}_j) = \text{state}_j = \\ &= \text{Relu}(W_{\text{state,one-hot}} \times \text{one-hot}_j + \\ &+ W_{\text{state,state}} \times \text{state}_{j-1} + W_{\text{state}}), \end{aligned}$$

где $\text{state}_{j-1}, \text{state}_j \in \mathbb{R}^N$ – векторы внутреннего состояния рекуррентного слоя для $j-1$ и j входных one-hot векторов соответственно; $W_{\text{state,one-hot}} \in \mathbb{R}^{N \times 256}$, $W_{\text{state,state}} \in \mathbb{R}^{N \times N}$, $W_{\text{state}} \in \mathbb{R}^{N}$ – матрицы весов рекуррентного слоя; $W_{\text{state,one-hot}} \times \text{one-hot}_j$ – вклад текущего байта подпоследовательности; $W_{\text{state,state}} \times \text{state}_{j-1}$ – вклад предыдущих байтов подпоследовательности (вклад внутреннего состояния рекуррентного слоя); Relu – функция активации в рекуррентном слое.

Приведенное соотношение аналогично для слоя обратного порядка с поправкой на очередность подаваемых векторов. Выход с рекуррентного слоя передается в слой активации, а после полученный вектор вероятностей обрабатывается решающим правилом для получения подпоследовательности признаков «начала функции»:

$$\text{Activation layer}: \mathbb{R}^{L\times 2N} \to \mathbb{Z}_2^L,$$

где Sigmoid – используемая функция активации в слое.

Как было упомянуто, наличие признака «начала функции» для i-го байта подпоследовательности определяется из сравнения полученной вероятности для i-го байта с пороговым значением 0,5.

Подпоследовательности признаков объединяются в последовательность O^F, являющуюся выходом решения задачи распознавания функции в бинарных файлах.

Исследуемые параметры. Приведем список исследуемых гиперпараметров модели:

• Длина входной последовательности байт. Исходный файл размером в 10000 байт можно разделить большим количеством способов. Например, на 10 последовательностей длиной 1000 или 100 последовательностей с длиной в 100 байт. В исследовании от 2015 г. этому не уделяется необходимое внимание. Авторы принимают длину последовательности равной 1000 байт и, в отличие от количества нейронов в рекуррентном слое, не приводят никакой информации об исследованиях. Значение параметра длины входной последовательности определяется длинами прологов, эпилогов и выравниваний функций.

• Количество нейронов в рекуррентном слое. Исходная модель содержит в рекуррентном слое 16 нейронов. Количество нейронов неявным образом влияет на способность слоя обрабатывать закономерности между байтами инструкций (данных) входных последовательностей.

• Веса функции потерь. Уже упоминалась проблема дисбаланса классов. Отношение количества байт начала функции к количеству байт, не обладающих признаком «начала функции», связано с оптимальным значением весов классов функции потерь.

В рамках работы исследовалась зависимость между качеством распознавания и только теми гиперпараметрами модели, которые соответствуют особенностям машинной архитектуры или структуре самого бинарного файла. Поэтому в главе с результатами не приведены эксперименты, например, для функции активации в рекуррентном слое (tanh или различные вариации ReLU: Leaky ReLU, PReLU, ELU) или архитектуры рекуррентного слоя (GRU и LSTM). Для этих гиперпараметров не определяется прямая связь с машинной архитектурой, под которую собран бинарный файл, или форматом самого бинарного файла.

Необходимо отметить, что необоснованное усложнение архитектуры рекуррентного слоя увеличивает количество обучаемых параметров нейронной сети, что отражается на времени обучения одной эпохи и количестве эпох, требуемых для сходимости. Данный вывод согласуется с точкой зрения авторов модели RNN. Опираясь на результаты своих экспериментов, авторы делают вывод о том, что применение LSTM и GRU в качестве архитектуры рекуррентного слоя не дает значительного увеличения показателей метрик качества в сравнении с обычным SimpleRNN в задаче распознавания начал функций.

Эксперименты с исследуемыми параметрами необходимо провести для случайных байтов выравнивания, как это предложено в работе «Padding Matters» [25].

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Представим графики метрик качества для исследуемых параметров модели. В каждом эксперименте изменялся только один параметр, если о другом не написано явно. По умолчанию значения параметров устанавливались либо рекомендованными значениями от авторов RNN, либо некоторыми значениями, эмпирически подобранными авторами данной работы. Параметры эмпирически подбирались, если авторы модели RNN явно их не описывали. Значения гиперпараметров модели по умолчанию:

• длина one-hot вектора – 256; единица в ячейке с номером, равным значению байта, остальное – нули;

• длина входной последовательности – 1000 байт;
• оптимизатор – rmsprop;
• значение весов в бинарной кросс-энтропии – 0,9 для 1 класса, 0,1 для 0 класса;
• количество нейронов в рекуррентном слое – 16;
• архитектура рекуррентного слоя – SimpleRNN;
• функция активации в рекуррентном слое – ReLU;
• функция активации в нейроне выходного слоя – Sigmoid;
• порог определения принадлежности к классу – 0,5;
• количество эпох обучения – 30.

Все графики приведены для тестирующей выборки.

Обучение со стандартным выравниванием. На рис. 1 изображены графики метрики F1 для класса 1 в зависимости от эпохи обучения для ESP32 и STM32WBA6. Изменяемый гиперпараметр модели – количество нейронов в рекуррентном слое.

Уменьшение количества нейронов существенно ухудшает качество работы модели, что особенно заметно при критическом значении в 6 нейронов в слое. Увеличение количества нейронов в рекуррентном слое незначительно улучшает качество

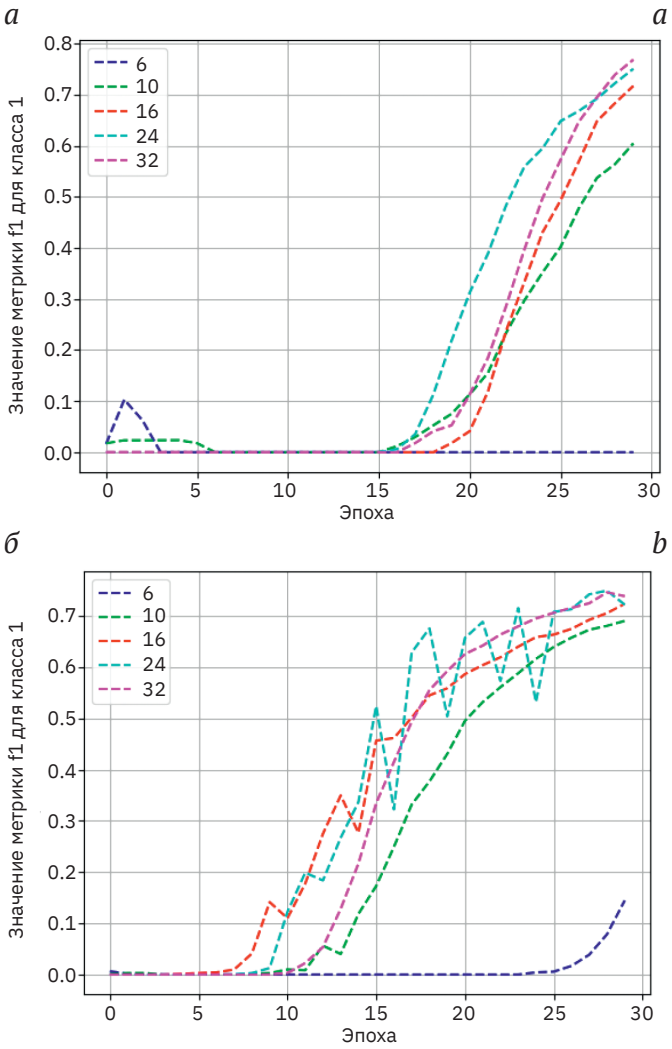


Рис. 1 | Графики метрики F1 класса 1 при изменении количества нейронов в рекуррентном слое для ESP32 (*а*) и STM32WBA6 (*б*)

Fig. 1 | Graphs of Class 1 F1 metrics with changes in the number of neurons in the recurrent layer for ESP32 (*a*) and STM32WBA6 (*b*)

работы модели. На графике для ESP32 значение метрики F1 для класса 1 с оригинальным значением гиперпараметра в 16 нейронов составило 0,72, а для 32 нейронов это значение 0,77. В то же время на графике для STM32WBA6 увеличение числа нейронов с 16 до 32 позволило повысить значение F1 всего лишь с 0,72 до 0,74. В обоих случаях графики имеют тенденцию к сходимости, что ограничивает возможность в существенном увеличении качества работы модели посредством увеличения числа нейронов.

Исходя из этого, можно сделать вывод, что увеличение количества нейронов в рекуррентном слое позволяет улучшить качество работы модели со стандартным выравниванием при фиксированных значениях остальных гиперпараметров, однако у этого улучшения есть верхний предел.

Длина входной последовательности. На рис. 2 изображены графики метрики F1 для класса 1 в зависимости от эпохи обучения для ESP32 и STM32WBA6. Изменяемый гиперпараметр модели – длина входной последовательности.

Увеличение длин входных последовательностей приводит к избытку контекста у нейронной сети. Для распознавания начал функций методами машинного обучения нет необходимости в анализе полного кода тел функций. Вместо этого необходимо выделять важные признаки в потоке байтов: прологи, эпилоги и выравнивания. Длинные последовательности затрудняют выделение этих признаков. На графиках проблема

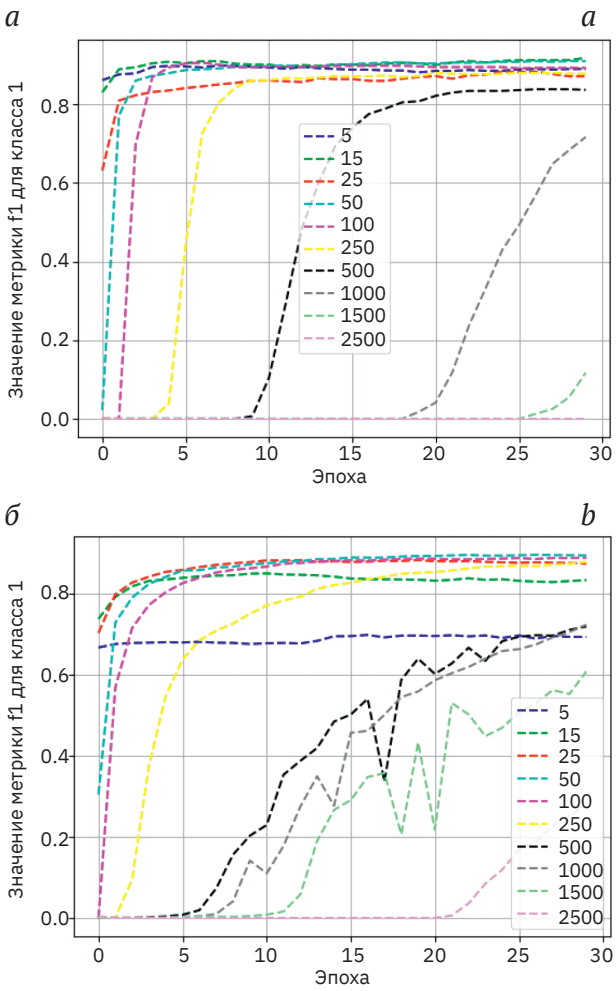


Рис. 2 | Графики метрики F1 класса 1 при изменении длины входной последовательности для ESP32 (*а*) и STM32WBA6 (*б*)

Fig. 2 | Graphs of Class 1 F1 metrics when changing the length of the input sequence for ESP32 (*a*) and STM32WBA6 (*b*)

избытка контекста проявляется в увеличении числа эпох, необходимого для сходимости обучения модели. Метрика F1 у длинных входных последовательностей после 30 эпох значительно ниже, чем у средних последовательностей. Аналогичное справедливо для средних и коротких последовательностей, но по достижении меньшего числа эпох.

С другой стороны, чрезмерное уменьшение длины входной последовательности может приводить к нехватке контекста у нейронной сети. На графике для STM32WBA6 это заметно для длин последовательностей в 15 и особенно 5 байт. На рис. 3 отражены типичные эпилог функции *X* и пролог функции *X*+1 для STM32WBA6. Если посчитать байты инструкций: эпилог (6–8+ байт), пролог (6–8+ байт) и возможное выравнивание функций по адресам, кратным четырем (1–3 байта), то получится, что длины последовательности менее 15 байт недостаточно для распознавания начал функций. Из-за разделения на слишком короткие последовательности контекст может чаще размазываться по нескольким входным последовательностям. В данном случае имеется в виду то, что эпилог функции *X* с выравниванием может попасть в *i*-ю последовательность, а пролог функции *X*+1 попадет в *i*+1 последовательность. Тогда при определении класса у байта начала функции *X*+1 нейросеть на входе не будет иметь эпилог и выравнивание.

Описанная закономерность для малых входных последовательностей не проявляется на графиках для ESP32, поскольку, в отличие от STM32WBA6, подавляющее большинство функций файла программы микроконтроллера ESP32 начинается с трехбайтовой инструкции выделения памяти на стеке. Тем не менее для некоторых машинных архитектур существенное уменьшение длины входной последовательности будет критичным. В частности, для архитектур с длинными инструкциями.

Исходя из всех приведенных суждений, можно сделать вывод, что оптимальная длина входной последовательности для решения поставленной задачи распознавания – это некоторое среднее значение. Этому среднему значению для обеих машинных архитектур в случае стандартного выравнивания соответствует длина 50.

Вес класса 1 в функции потерь бинарной кросс-энтропии. Графики показателей метрик для весовых коэффициентов функции потерь из списка наложились друг на друга. При обучении брались следующие весовые коэффициенты для класса 1: 0,5; 0,8; 0,95; 0,99; 0,998. Изменение данного гиперпараметра не изменяет показатели метрик в случае стандартного выравнивания.

Совпадение графиков можно объяснить слишком большим дисбалансом классов в обучающей и тестирующей выборках. Для исходных данных это соотношение примерно равно 1 к 400.

Обучение со случайным выравниванием. Случайное выравнивание означает, что в исходных последовательностях обучающей и тестирующей выборок все нулевые байты выравнивания заменяются на случайные байты из 0,255.

```
.text:0803195C 83 20                    MOVS    R0, #0x83
.text:0803195E E0 75                    STRB    R0, [R4,#0x17]
.text:08031960 00 20                    MOVS    R0, #0
.text:08031962 B0 BD                    POP     {R4,R5,R7,PC}
.text:08031962                  ; End of function SMP_MI_Send_Pairing_Random
.text:08031962
.text:08031964
.text:08031964                  ; =============== S U B R O U T I N E =======================================
.text:08031964
.text:08031964
.text:08031964                  SMP_LP_Compute_Confirm_Value            ; CODE XREF: SMP_Process_Rx_Packet+182↑p
.text:08031964                                                          ; SMP_Process_Rx_Packet+2EA↑p ...
.text:08031964
.text:08031964                  var_38          = -0x38
.text:08031964                  var_31          = -0x31
.text:08031964                  var_2A          = -0x2A
.text:08031964                  var_23          = -0x23
.text:08031964
.text:08031964 2D E9 F0 43              PUSH.W  {R4-R9,LR}
.text:08031968 8B B0                    SUB     SP, SP, #0x2C
.text:0803196A 6F 46                    MOV     R7, SP
.text:0803196C 89 46                    MOV     R9, R1
.text:0803196E 90 46                    MOV     R8, R2
```

Рис. 3 | Эпилог и пролог функций для STM32WBA6

Fig. 3 | Epilogue and prologue of functions for STM32WBA6

В работе [26] приводятся примеры преднамеренных и непреднамеренных атак на модели нейронных сетей, распознающих границы функций внутри бинарного файла. В контексте преднамеренных атак авторы в том числе упоминают использование необычных настроек стандартных систем сборки, которые на практике часто используются разработчиками вредоносного программного обеспечения. Одной из атак, повлиявших на качество работы моделей, является замена байтов заполнения на байты исполняемых инструкций. Идею подобных атак развивают авторы статьи «Padding Matters». По словам авторов, на данный момент уже встречаются обфускаторы бинарного кода, изменяющие байты выравниваний. Тем не менее при проверке работы RNN модели на x86 и x86–64 бинарных файлах со случайным выравниванием метрики качества для обеих архитектур сильно упали. Это говорит о том, что модель сильно полагалась на выравнивания при распознавании [26].

В рамках данной работы предлагается рассматривать случайные выравнивания именно как способ проверки качества обработки моделью прологов функций в отсутствие стандартных выравниваний.

Количество нейронов в рекуррентном слое. На рис. 4 представлены графики

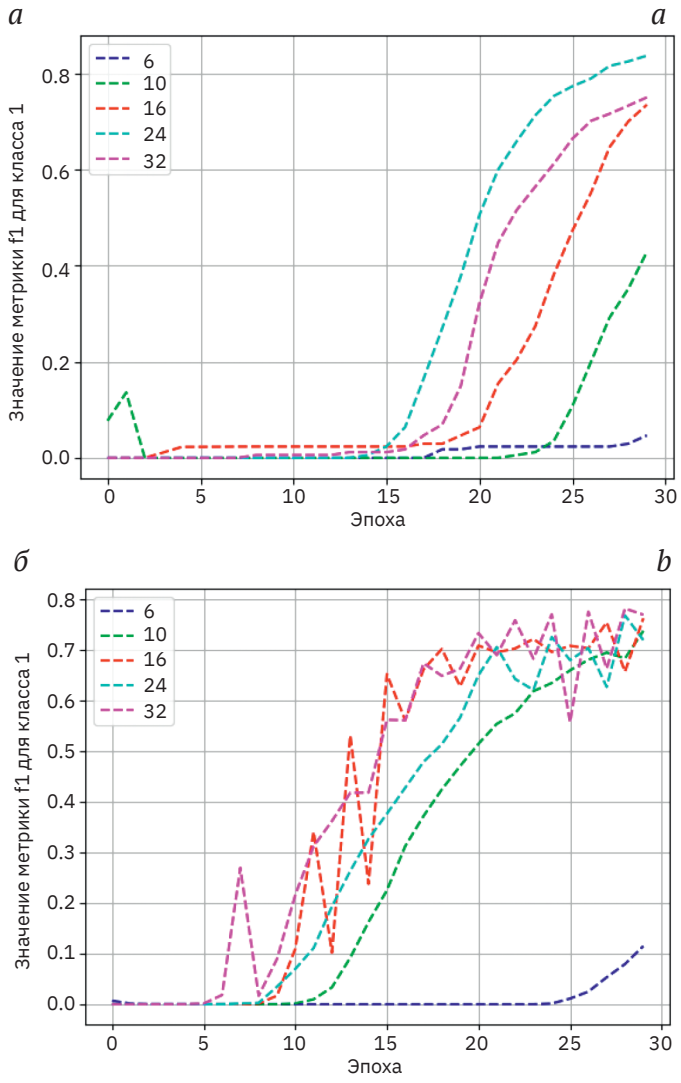


Рис. 4 | График метрики F1 класса 1 для случайного выравнивания при изменении количества нейронов в рекуррентном слое для ESP32 (*а*) и STM32WBA6 (*б*)

Fig. 4 | Graph of Class 1 F1 metric for random alignment when changing the number of neurons in the recurrent layer for ESP32 (*a*) and STM32WBA6 (*b*)

метрики F1 для класса 1 в зависимости от эпохи обучения для ESP32 и STM32WBA6. Изменяемый гиперпараметр модели – количество нейронов в рекуррентном слое. При этом в обучающей и тестирующей выборках стандартные нулевые выравнивания заменены на случайные значения.

Поведение графиков метрики качества F1 для класса 1 у разного количества нейронов в рекуррентном слое почти аналогично графикам модели со стандартным выравниванием за исключением единственной аномалии в случае для ESP32 с 24 нейронами в рекуррентном слое.

В ходе проведения эксперимента возникло предположение об обобщаемости задачи распознавания начал функций в бинарных файлах с различными видами выравниваний. Под обобщаемостью задачи подразумевается возможность работы с оптимальными значениями метрик качества рассматриваемой нейронной сети, обученной со случайным заполнением, на данных со случайным или стандартным заполнением.

Для полноты исследования стоит рассмотреть обратную ситуацию – стандартное заполнение в обучающей выборке и случайное или стандартное заполнение в тестирующей выборках. Таким образом, необходимо рассмотреть четыре возможных случая:

- стандартное выравнивание и в обучающей, и в тестирующей выборках;
- стандартное выравнивание в обучающей и случайное выравнивание в тестирующей;
- случайное выравнивание в обучающей и стандартное заполнение в тестирующей;
- случайное выравнивание и в обучающей, и в тестирующей выборках.

Не рассматриваются случаи с одновременным использованием стандартного и случайного выравниваний в выборках исходя из предположения о том, что в одном или нескольких схожих бинарных файлах разработчик будет использовать одинаковые инструменты сборки и обфускации.

На рис. 5 показаны графики метрики F1 для класса 1 для четырех случаев для ESP32 и STM32WBA6.

Исходя из полученных графиков, можно сделать вывод о необобщаемости задачи распознавания функций для модели со случайным выравниванием. Особенно это заметно для графика метрики F1 класса 1 для ESP32. Модель, обученная на случайном выравнивании, показывает низкие показатели оценивающих метрик при работе на стандартном выравнивании. Аналогичный вывод делается и для модели, обученной на стандартном выравнивании.

Таким образом, исследователю необходимо выбирать одну из двух моделей для распознавания функций в новом бинарном файле. Выравнивание данных в обучающей выборке должно совпадать с выравниванием в рассматриваемом бинарном файле. В противном случае может сильно ухудшиться качество распознавания.

Длина входной последовательности. На рис. 6 изображены графики метрики F1 для класса 1 в зависимости от эпохи обучения для ESP32 и STM32WBA6. Изменяемый гиперпараметр модели – длина входной последовательности. При этом в обучающей и тестирующей выборках стандартные нулевые выравнивания заменены на случайные значения.

Поведение графиков метрики качества F1 для класса 1 для разных длин входной последовательности почти аналогично графикам модели со стандартным выравниванием. Оптимальная длина входной последовательности все так же соответствует среднему значению длины. Для случайного выравнивания это длина 100 байт для обеих машинных архитектур.

Вес класса 1 в функции потерь бинарной кросс-энтропии. Графики показателей метрик для весовых коэффициентов функции потерь из списка 0,8; 0,95; 0,99; 0,998 наложились друг на друга. Изменение данного гиперпараметра не изменяет показатели метрик и в случае случайного выравнивания.

Ошибки распознавания. *Ошибки первого рода.* Типичная ошибка первого рода

распознавания начала функции приведена на рис. 7.

Ошибка возникает в служебных отладочных секциях файла программы микроконтроллера, содержащих адресные, размерные, индексные и флаговые константы. В данном случае в подпоследовательности байтов «00 00 F8 B5 03 08 04 00» распознаются признаки наличия функции по адресу 0xE54C5:

• Байты «00 00» ошибочно определяются как выравнивание функций по адресам, кратным 4.

• Байты «F8 B5» определяются как инструкция PUSH {R3-R7, LR} загрузки значений регистров в стек из набора Thumb-2 архитектуры ARMv8-M. При этом младший бит байта «B5» определяет загрузку регистра LR адреса возврата в стек, что свойственно прологам функций.

• Последующие байты определяются моделью как типовые инструкции работы с данными, переданными в качестве параметров функции.

В качестве дополнительного эксперимента поочередно заменены байты «00 00»

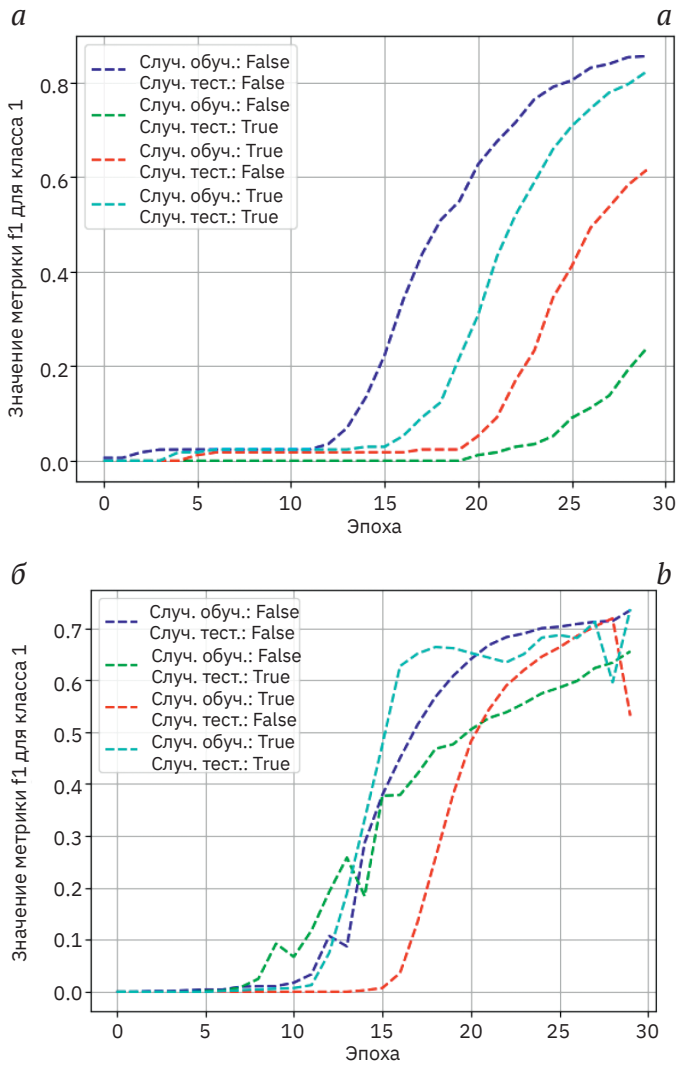


Рис. 5 | График метрики F1 класса 1 для комбинаций случайного и стандартного выравнивания в обучающей и тестирующей выборках для ESP32 (*а*) и STM32WBA6 (*б*)

Fig. 5 | Graph of Class 1 F1 metrics for combinations of random and standard alignment in training and testing samples for ESP32 (*a*) and STM32WBA6 (*b*)

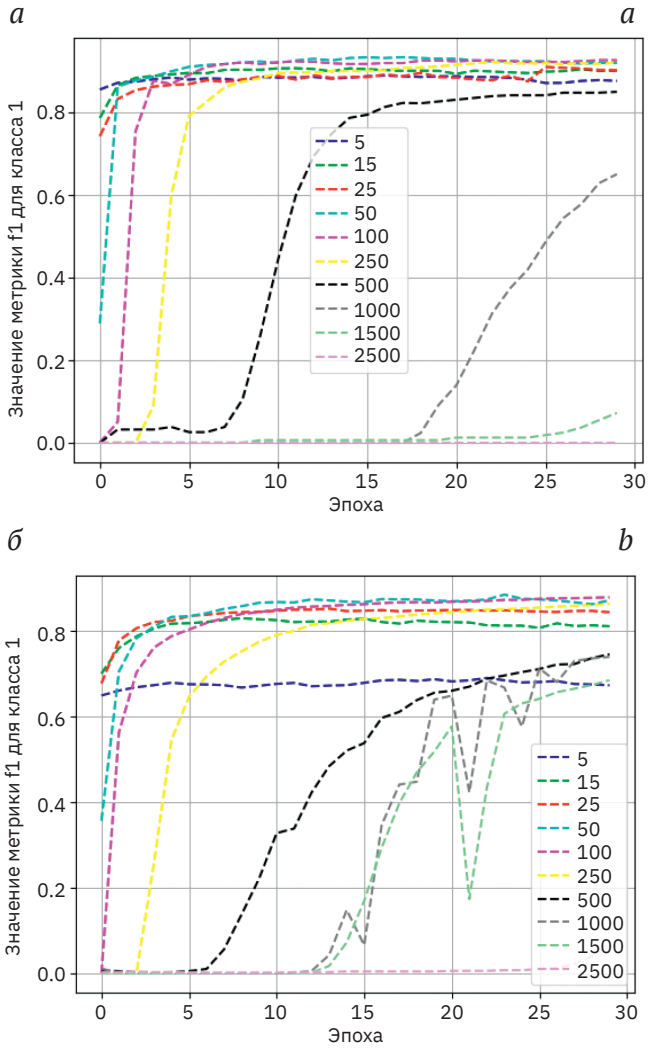


Рис. 6 | График метрики F1 класса 1 для случайного выравнивания при изменении длины входной последовательности для ESP32 (*а*) и STM32WBA6 (*б*)

Fig. 6 | Graph of Class 1 F1 metric for random alignment when changing the length of the input sequence for ESP32 (*a*) and STM32WBA6 (*b*)

```
:000E54C0 B1                            DCB 0xB1
:000E54C1 1A                            DCB 0x1A
:000E54C2 00                            DCB   0
:000E54C3 00                            DCB   0
:000E54C4
:000E54C4                       ; =============== S U B R O U T I N E ===============
:000E54C4
:000E54C4
:000E54C4                       sub_E54C4
:000E54C4 F8 B5                         PUSH        {R3-R7,LR}
:000E54C6 03 08                         LSRS        R3, R0, #0x20 ; ' '
:000E54C8 04 00                         MOVS        R4, R0
:000E54CA 00 00                         MOVS        R0, R0
:000E54CC 00 00                         MOVS        R0, R0
```

Рис. 7 | Ошибка первого рода распознавания начала функции для STM32WBA6

Fig. 7 | Error of the first kind of recognition of the function start for STM32WBA6

на байты случайной инструкции Thumb-2, и инвертирован младший бит байта «B5», после чего было еще раз запущено распознавание. В обоих случаях модель определила байт по адресу 0xE54C5 как байт, не обладающий признаком «начала функции», т. е. модель отработала верно.

Ошибки второго рода. Типичная ошибка второго рода распознавания начала функции приведена на рис. 8.

В данном случае модель не смогла распознать функцию по адресу 0x425B0, а автоанализ IDA Pro посчитал нераспознанный блок кода продолжением распознанной ранее функции по адресу 0x4258C. Причиной возникновения ошибки является отсутствие явных типовых признаков наличия функции в последовательности байт:

• Вместо выравнивания функции компилятор разместил четырехбайтовые константы «A8E40020» и «53B30808», используемые инструкциями LDR загрузки данных из памяти в регистры. Эти константы ошибочно воспринимаются моделью как последовательность инструкций обработки данных в регистрах и условного перехода из Thumb-2.

• Система сборки скомпилировала нетипичные эпилог функции 0x4258C, заканчивающийся на байте по адресу 0x425A5 включительно, и пролог функции 0x425B0 для оптимизации генерируемого кода. Однако нетипичность пролога и эпилога усложняют распознавание функции.

Обсуждение полученных результатов. По результатам обучения моделей сделаны следующие выводы о влиянии гиперпараметров, соответствующих особенностям машинной архитектуры и форматов бинарных файлов, на показатели метрик качества:

• Количество нейронов в рекуррентном слое – параметр средней значимости. Увеличение количества нейронов может дополнительно незначительно улучшить качество работы модели, однако при увеличении более чем в два раза показатели метрик сходятся. Это справедливо для обоих видов выравниваний.

• Длина входной последовательности байт – наиболее значимый гиперпараметр.

```
:0004258C                 ; =============== S U B R O U T I N E =======================================
:0004258C
:0004258C
:0004258C
:0004258C                 sub_4258C                               ; CODE XREF: sub_3FEE8+A↑p
:0004258C                                                         ; sub_42666+4↓j ...
:0004258C 80 B5                           PUSH            {R7,LR}
:0004258E 06 48                           LDR             R0, =0x2000E4A8
:00042590 01 88                           LDRH            R1, [R0]
:00042592 21 B1                           CBZ             R1, loc_4259E
:00042594 04 30                           ADDS            R0, #4
:00042596 BD E8 80 40                     POP.W           {R7,LR}
:0004259A 00 F0 09 B8                     B.W             loc_425B0
:0004259E                 ; ---------------------------------------------------------------------------
:0004259E
:0004259E                 loc_4259E                               ; CODE XREF: sub_4258C+6↑j
:0004259E 03 48                           LDR             R0, =0x808B353
:000425A0 55 21                           MOVS            R1, #0x55 ; 'U'
:000425A2 CA F7 F7 FF                     BL              sub_D594
:000425A2                 ; ---------------------------------------------------------------------------
:000425A6 00                              DCB    0
:000425A7 BF                              DCB 0xBF
:000425A8 A8 E4 00 20     off_425A8       DCD 0x2000E4A8          ; DATA XREF: sub_4258C+2↑r
:000425AC 53 B3 08 08     dword_425AC     DCD 0x808B353           ; DATA XREF: sub_4258C:loc_4259E↑r
:000425B0                 ; ---------------------------------------------------------------------------
:000425B0
:000425B0                 loc_425B0                               ; CODE XREF: sub_4258C+E↑j
:000425B0 01 68                           LDR             R1, [R0]
:000425B2 48 F2 4E 32                     MOVW            R2, #0x834E
:000425B6 A1 FB 02 12                     UMULL.W         R1, R2, R1, R2
:000425BA 02 EB 51 01                     ADD.W           R1, R2, R1,LSR#1
:000425BE 21 F0 00 42                     BIC.W           R2, R1, #0x80000000
:000425C2 00 29                           CMP             R1, #0
:000425C4 48 BF                           IT MI
:000425C6 51 1C                           ADDMI           R1, R2, #1
:000425C8 01 60                           STR             R1, [R0]
:000425CA 08 46                           MOV             R0, R1
:000425CC 70 47                           BX              LR
:000425CC                 ; End of function sub_4258C
```

Рис. 8 | Ошибка второго рода распознавания начала функции для STM32WBA6

Fig. 8 | Error of the second kind of recognition of the beginning of the function for STM32WBA6

Разделение потока байт программы на длинные входные последовательности приводит к избытку бесполезного контекста. В результате увеличивается число эпох, необходимое для сходимости обучения модели. В то же время разделение на малые последовательности может привести к нехватке полезного контекста в последовательностях байтов, обладающих свойством «начала функции». Под полезным контекстом в данном случае подразумеваются выравнивания, прологи и эпилоги, а под бесполезным – тело функции без пролога и эпилога, строки и другие константы. Оптимальным значением гиперпараметра длины входной последовательности для достижения наилучших значений показателей метрик качества будет некоторое среднее значение. Для ESP32 и STM32WBA6 это значение составило 50 для стандартного выравнивания и 100 для случайного.

• Веса функции потерь – незначимый гиперпараметр.

Значения оценивающих метрик для стандартного и случайного выравниваний, а также соответствующие им значения гиперпараметров исходной и улучшенной моделей приведены в табл. 1 и 2.

В процессе обучения моделей сделан вывод о необобщаемости задачи распознавания функций. Модель, обученная на случайных выравниваниях, плохо справлялась с распознаванием начал функций со стандартными выравниваниями. Аналогично плохо работала модель, обученная на стандартных выравниваниях, на тестирующей выборке со случайными выравниваниями.

Для выбора используемой модели (для установления типа выравнивания) исследователю необходимо вручную проанализировать фрагмент бинарного файла с исполняемым машинным кодом.

Расширение для IDA Pro. Репозиторий проекта содержит в себе две функциональные части [27]:

Таблица 1 | Значения оценивающих метрик и соответствующие им значения гиперпараметров исходной RNN модели

Table 1 | Values of the estimating metrics and their corresponding hyperparameter values of the initial RNN model

Микроконтроллер	Выравнивание	Длина последовательности	Кол-во нейронов	Веса функции потерь	F1 для класса 1	Взвешенное F1
ESP32	Стандартное	1000	16	–	0,717	0,998
	Случайное	1000	16	–	0,735	0,998
STM32WBA6	Стандартное	1000	16	–	0,724	0,995
	Случайное	1000	16	–	0,739	0,995

Таблица 2 | Значения оценивающих метрик и соответствующие им значения гиперпараметров улучшенной RNN модели

Table 2 | Values of the evaluation metrics and their corresponding hyperparameter values of the improved RNN model

Микроконтроллер	Выравнивание	Длина последовательности	Кол-во нейронов	Веса функции потерь	F1 для класса 1	Взвешенное F1
ESP32	Стандартное	50	32	–	0,904	0,999
	Случайное	100	32	–	0,930	0,999
STM32WBA6	Стандартное	50	32	–	0,897	0,998
	Случайное	100	32	–	0,862	0,997

• Консольная утилита, разработанная на Python3 с использованием библиотеки машинного обучения tensorflow. Данная утилита содержит в себе программную реализацию модели. Назначение утилиты – обучение модели и сохранение обученных весов для последующего использования непосредственно с самим расширением внутри IDA Pro.

• Расширение для одной из последних версии IDA Pro 9.1. Расширение разработано на Python3 с использованием IDA Python фреймворка от компании Hexrays.

4. ЗАКЛЮЧЕНИЕ

Представлено решение одной из ключевых подзадач реверс-инжиниринга – задачи распознавания начал функций в бинарном файле. Предлагаемое решение автоматизирует распознавание начал функций, используя модель нейронной сети.

В работе представлены:

• Описание модели нейронной сети с двунаправленным рекуррентным слоем. Модель представляет собой оптимизированную RNN модель.

• Результаты экспериментов с гиперпараметрами модели для случайного и стандартного выравниваний функций в бинарных файлах. На основе результатов сделаны выводы о влиянии на качество распознавания гиперпараметров модели (длина входной последовательности, количество нейронов в рекуррентном слое и веса функции потерь), соответствующих особенностям машинной архитектуры или форматов бинарных файлов.

• Сводная таблица оптимальных значений гиперпараметров для бинарных файлов, собранных под микропроцессоры ESP32 и STM32WBA6 архитектур Xtensa Little Endian и ARMv8-M соответственно.

• Описание и исходный код разработанного расширения для дизассемблера IDA Pro – одного из самых используемых инструментов обратной разработки.

Реверс-инжиниринг бинарных файлов редких машинных архитектур иногда подразумевает отсутствие разнообразия бинарных файлов для сравнительного анализа и отсутствие возможности динамического анализа. Ключевой особенностью предлагаемого решения является то, что оно учитывает описанную прикладную специфику задач реверс-инжиниринга, проявляющуюся в малой выборке обучающих данных. Автор предлагает два сценария использования результатов исследования:

• Автоматизированный перенос разметки функций из одного разобранного бинарного файла в другой, если перед исследователем стоит задача анализа нескольких схожих по структуре бинарных файлов. Данными для обучения в этом случае станет уже ранее вручную разобранный исследователем бинарный файл.

• Автоматизированная разметка анализируемого бинарного файла посредством сборки собственного бинарного файла и обучения модели на нем. Исходный код проектов для сборки может быть взят из открытых репозиториев. Важным условием является определение окружения и инструментов сборки анализируемого бинарного файла. Репозиторий расширения для IDA Pro содержит в себе веса моделей для ESP32 и STM32WBA6, собранных посредством стандартных фреймворков разработки от компаний производителей этих микроконтроллеров.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Wartell R., Zhou Y., Hamlen K. W. et al.** Differentiating code from data in x86 binaries // Machine Learning and Knowledge Discovery in Databases. 2011. P. 522–536. DOI: 10.1007/978-3-642-23808-6_34.
2. **Benkraouda H., Diwan N., Wang G.** You Can't Judge a Binary by Its Header: Data-Code Separation for Non-Standard ARM Binaries Using Pseudo Labels // IEEE Symposium on Security and Privacy (SP). 2025. P. 3727–3745. DOI: 10.1109/SP61157.2025.00036.
3. **Pei K., Guan J., Williams-King D. et al.** XDA: accurate, robust disassembly with transfer learning // Network and Distributed Systems Security (NDSS) Symposium 2021, 21–24 February 2021, San Diego, CA, USA. 2021. DOI: 10.14722/ndss.2021.23112.
4. **Yu S., Qu Y., Hu X., Yin H.** DeepDi: Learning a relational graph convolutional network model on instructions for fast and accurate disassembly // 31st USENIX Security Symposium (USENIX Security 22). 2022. P. 2709–2725.
5. **Siliang Qin1, Fengrui Yang, Hao Wang et al.** Tady: A Neural Disassembler without Structural Constraint Violations. URL: https://arxiv.org/pdf/2506.13323 (дата обращения: 03.12.2025).
6. **David Y., Alon U., Yahav E.** Neural Reverse Engineering of Stripped Binaries using Augmented Control Flow Graphs. Proceedings of the ACM on Programming Languages. 2020. Vol. 4 (OOPSLA). 28 p. DOI: 10.1145/3428293.
7. **Patrick-Evans J., Cavallaro L., Kinder J.** Probabilistic Naming of Functions in Stripped Binaries // Annual Computer Security Applications Conference (ACSAC 2020), 7–11 December 2020, Austin, USA. 2020. P. 373–385. DOI: 10.1145/3427228.3427265.
8. **Jiang L., Jin X., Lin Z.** Beyond Classification: Inferring Function Names in Stripped Binaries via Domain Adapted LLMs. URL: https://www.ndss-symposium.org/wp-content/uploads/2025-797-paper.pdf (дата обращения: 03.12.2025).
9. **Bao T., Burket J., Woo M., Turner R., Brumley D.** BYTEWEIGHT: Learning to Recognize Functions in Binary Code // 23rd USENIX Security Symposium. 2014. P. 845–860.
10. **Shin E.C.R., Song D., Moazzezi R.** Recognizing Functions in Binaries with Neural Networks // 24th USENIX Security Symposium (USENIX Security 15). 2015. P. 611–626.
11. **He J., Li S., Wang X., Yang J.** Neural-FEBI: Accurate Function Identification in Ethereum Virtual Machine Bytecode // Journal of Systems and Software. 2023. Vol. 199. № 111627. DOI: 10.1016/j.jss.2023.111627.
12. **Pei K., Guan J., Broughton M. et al.** StateFormer: Fine-Grained Type Recovery from Binaries using Generative State Modeling // Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2021. P. 690–702. DOI: 10.1145/3468264.3468607.
13. **Nitin V., Saieva A., Ray B., Kaiser G.** A Transformer-based Model for Decompiled Identifier Renaming // Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021), 1–6 August 2021. 2021. P. 48–57.
14. **Wang H., Qu W., Katz G. et al.** jTrans: Jump-Aware Transformer for Binary Code Similarity // Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA). 2022. P. 1–13. DOI: 10.1145/3533767.3534367.
15. **Yu Z., Cao R., Tang Q. et al.** OrderMatters: Semantic-Aware Neural Networks for Binary Code Similarity Detection // Proceedings of the AAAI Conference on Artificial Intelligence. 2020. Vol. 34. P. 1145–1152. DOI: 10.1609/aaai.v34i01.5466.
16. **Duan Y., Li X., Wang J., Yin H.** DeepBinDiff: Learning Program-Wide Code Representations for Binary Diffing // Network and Distributed System Security Symposium (NDSS). 2020. DOI: 10.14722/ndss.2020.24311.
17. **Li X., Qu Y., Yin H.** PalmTree: Learning an Assembly Language Model for Instruction Embedding // ACM SIGSAC Conference on Computer and Communications Security (CCS '21). 2021. P. 3236–3251. DOI: 10.1145/3460120.3484587.
18. **Gao Z., Wang H., Wang Y., Zhang C.** Virtual Compiler Is All You Need For Assembly Code Search // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024. P. 3040–3051. DOI: 10.18653/v1/2024.acl-long.167.

19. **Liu C., Saul R., Sun Y. et al.** ASSEMBLAGE: Automatic Binary Dataset Construction for Machine Learning. URL: https://openreview.net/pdf?id=dsK5EmmomU (дата обращения: 03.12.2025).
20. **Andriesse D., Slowinska A., Bos H.** Compiler-agnostic function detection in binaries // IEEE European Symposium on Security and Privacy. 2017. P. 251–276.
21. **Александров Я. А., Сафин Л. К., Чернов А. В., Трошина К. Н.** Определение границ подпрограмм при статическом анализе бинарных образов // Вопросы кибербезопасности. 2016. № 1 (14). С. 53–60.
22. **Flores-Montoya A., Schulte E.** Datalog disassembly // 29th USENIX Security Symposium (USENIX Security 20). 2020. P. 1075–1092.
23. **Evans R., Hawkins W., Wang B.** RustBound: Function Boundary Detection over Rust Stripped Binaries // Security and Privacy in Cyber-Physical Systems and Smart Vehicles. 2025. Vol. 622. P. 237–256.
24. **Guo W., Mu D., Xu J. et al.** LEMNA: Explaining Deep Learning based Security Applications // ACM SIGSAC Conference on Computer and Communications Security (CCS '18). 2018. P. 364–379.
25. **Springer R., Schmitz A., Leinweber A. et al.** Padding Matters – Exploring Function Detection in PE Files // arXiv:2504.21520. 2025.
26. **Bundt J., Davinroy M., Agadakos I. et al.** Black-box Attacks Against Neural Binary Function Detection // The 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23). 2023. 16 p. DOI: 10.1145/3607199.3607200.
27. RNN-Function-Finder. URL: https://github.com/Tinkerrer/RNN-Function-Finder (дата обращения: 03.12.2025).

REFERENCES

1. **Wartell R., Zhou Y., Hamlen K. W. et al.** Differentiating code from data in x86 binaries. *Machine Learning and Knowledge Discovery in Databases*. 2011, pp. 522–536. DOI: 10.1007/978-3-642-23808-6_34.
2. **Benkraouda H., Diwan N., Wang G.** You Can't Judge a Binary by Its Header: Data-Code Separation for Non-Standard ARM Binaries Using Pseudo Labels. IEEE Symposium on Security and Privacy (SP). 2025, pp. 3727–3745. DOI: 10.1109/SP61157.2025.00036.
3. **Pei K., Guan J., Williams-King D. et al.** XDA: accurate, robust disassembly with transfer learning. Network and Distributed Systems Security (NDSS) Symposium 2021, 21–24 February 2021, San Diego, CA, USA. 2021. DOI: 10.14722/ndss.2021.23112.
4. **Yu S., Qu Y., Hu X., Yin H.** DeepDi: Learning a relational graph convolutional network model on instructions for fast and accurate disassembly. 31st USENIX Security Symposium (USENIX Security 22). 2022, pp. 2709–2725.
5. **Siliang Qin1, Fengrui Yang, Hao Wang et al.** Tady: A Neural Disassembler without Structural Constraint Violations. URL: https://arxiv.org/pdf/2506.13323 (accessed: 03.12.2025).
6. **David Y., Alon U., Yahav E.** Neural Reverse Engineering of Stripped Binaries using Augmented Control Flow Graphs. Proceedings of the ACM on Programming Languages. 2020. Vol. 4 (OOPSLA), 28 p. DOI: 10.1145/3428293.
7. **Patrick-Evans J., Cavallaro L., Kinder J.** Probabilistic Naming of Functions in Stripped Binaries. Annual Computer Security Applications Conference (ACSAC 2020), 7–11 December 2020, Austin, USA. 2020, pp. 373–385. DOI: 10.1145/3427228.3427265.
8. **Jiang L., Jin X., Lin Z.** Beyond Classification: Inferring Function Names in Stripped Binaries via Domain Adapted LLMs. URL: https://www.ndss-symposium.org/wp-content/uploads/2025-797-paper.pdf (accessed: 03.12.2025).
9. **Bao T., Burket J., Woo M., Turner R., Brumley D.** BYTEWEIGHT: Learning to Recognize Functions in Binary Code. 23rd USENIX Security Symposium. 2014, pp. 845–860.
10. **Shin E.C.R., Song D., Moazzezi R.** Recognizing Functions in Binaries with Neural Networks. 24th USENIX Security Symposium (USENIX Security 15). 2015, pp. 611–626.
11. **He J., Li S., Wang X., Yang J.** Neural-FEBI: Accurate Function Identification in Ethereum Virtual Machine Bytecode. Journal of Systems and Software. 2023. Vol. 199. No. 111627. DOI: 10.1016/j.jss.2023.111627.

12. **Pei K., Guan J., Broughton M. et al.** StateFormer: Fine-Grained Type Recovery from Binaries using Generative State Modeling. Proceedings of the 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 2021, pp. 690–702. DOI: 10.1145/3468264.3468607.
13. **Nitin V., Saieva A., Ray B., Kaiser G.** A Transformer-based Model for Decompiled Identifier Renaming. Proceedings of the 1st Workshop on Natural Language Processing for Programming (NLP4Prog 2021), 1–6 August 2021. 2021, pp. 48–57.
14. **Wang H., Qu W., Katz G. et al.** jTrans: Jump-Aware Transformer for Binary Code Similarity. Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA). 2022, pp. 1–13. DOI: 10.1145/3533767.3534367.
15. **Yu Z., Cao R., Tang Q. et al.** OrderMatters: Semantic-Aware Neural Networks for Binary Code Similarity Detection. Proceedings of the AAAI Conference on Artificial Intelligence. 2020. Vol. 34, pp. 1145–1152. DOI: 10.1609/aaai.v34i01.5466.
16. **Duan Y., Li X., Wang J., Yin H.** DeepBinDiff: Learning Program-Wide Code Representations for Binary Diffing. Network and Distributed System Security Symposium (NDSS). 2020. DOI: 10.14722/ndss.2020.24311.
17. **Li X., Qu Y., Yin H.** PalmTree: Learning an Assembly Language Model for Instruction Embedding. ACM SIGSAC Conference on Computer and Communications Security (CCS '21). 2021, pp. 3236–3251. DOI: 10.1145/3460120.3484587.
18. **Gao Z., Wang H., Wang Y., Zhang C.** Virtual Compiler Is All You Need For Assembly Code Search. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2024, pp. 3040–3051. DOI: 10.18653/v1/2024.acl-long.167.
19. **Liu C., Saul R., Sun Y. et al.** ASSEMBLAGE: Automatic Binary Dataset Construction for Machine Learning. URL: https://openreview.net/pdf?id=dsK5EmmomU (accessed: 03.12.2025).
20. **Andriesse D., Slowinska A., Bos H.** Compiler-agnostic function detection in binaries. IEEE European Symposium on Security and Privacy. 2017, pp. 251–276.
21. **Aleksandrov Ya. A., Safin L. K., Chernov A. V., Troshina K. N.** Determining subroutine boundaries in static analysis of binary images. *Voprosy kiberbezopasnosti – Cybersecurity Issues*. 2016. No. 1 (14), pp. 53–60. (In Russian)
22. **Flores-Montoya A., Schulte E.** Datalog disassembly. 29th USENIX Security Symposium (USENIX Security 20). 2020, pp. 1075–1092.
23. **Evans R., Hawkins W., Wang B.** RustBound: Function Boundary Detection over Rust Stripped Binaries. *Security and Privacy in Cyber-Physical Systems and Smart Vehicles*. 2025. Vol. 622, pp. 237–256.
24. **Guo W., Mu D., Xu J. et al.** LEMNA: Explaining Deep Learning based Security Applications. ACM SIGSAC Conference on Computer and Communications Security (CCS '18). 2018, pp. 364–379.
25. **Springer R., Schmitz A., Leinweber A. et al.** Padding Matters – Exploring Function Detection in PE Files. *arXiv:2504.21520*. 2025.
26. **Bundt J., Davinroy M., Agadakos I. et al.** Black-box Attacks Against Neural Binary Function Detection. The 26th International Symposium on Research in Attacks, Intrusions and Defenses (RAID '23). 2023, 16 p. DOI: 10.1145/3607199.3607200.
27. RNN-Function-Finder. URL: https://github.com/Tinkerrer/RNN-Function-Finder (accessed: 03.12.2025).

СВЕДЕНИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHOR

ШАЙХАНОВ Артем Серикович – студент, Московский государственный технический университет имени Н. Э. Баумана, Россия, 105005, Москва, ул. 2-я Бауманская, д. 5, стр. 1
E-mail: artem.shaykhanov@gmail.com

SHAIKHANOV Artem S. – Student, Bauman Moscow State Technical University, Russia, 105005, Moscow, 2-ya Baumanskaya str., 5, build. 1

Научная статья
DOI 10.66424/2071-8217-2026-2-9
УДК 004.056

ЗАЩИТА ОТ СОСТЯЗАТЕЛЬНЫХ АТАК НА БАЗЕ ДИНАМИЧЕСКИ ПЕРЕСТРАИВАЕМОГО АНСАМБЛЯ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Г. Д. Гавва, М. О. Калинин*

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *max@ibks.spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Гавва Г. Д., Калинин М. О. Защита от состязательных атак на базе динамически перестраиваемого ансамбля моделей машинного обучения // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 113–120. DOI: 10.66424/2071-8217-2026-2-9

ПОСТУПИЛА 27.04.2026
ПРИНЯТА 07.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Рассмотрена проблема защиты моделей машинного обучения от состязательных атак. Представлен метод защиты, основанный на динамически перестраиваемом ансамбле классификаторов с механизмом отказа, который объединяет: случайную комбинацию гетерогенных подмоделей, онлайн-анализ дисперсии прогнозов, имитацию правдоподобного ответа при атаке и механизм моделей-ловушек. Анализ согласованности выходов внутри ансамбля и отказ от выдачи наиболее вероятного прогноза снижает результативность действий нарушителя при анализе им обратной связи, получаемой от целевой модели, и генерации состязательных образцов. Экспериментальная оценка, проведенная на наборе данных UNSW-NB15, показала, что разработанный метод сохраняет высокую исходную точность защищаемой модели при воздействии состязательных атак (85–95 %) при минимальном ее снижении на 1–3 п.п. Метод позволяет устранить до 98 % атак, что значительно превосходит показатели таких широко распространенных аналогов.

КЛЮЧЕВЫЕ СЛОВА

Защита машинного обучения, ансамбль моделей, классификация, механизм отказа, состязательные атаки

Original article
DOI 10.66424/2071-8217-2026-2-9

PROTECTION AGAINST ADVERSARIAL ATTACKS BASED ON A DYNAMICALLY RECONFIGURABLE ENSEMBLE OF MACHINE LEARNING MODELS

G. D. Gavva, M. O. Kalinin*

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *max@ibks.spbstu.ru

FOR CITATION

Gavva G. D., Kalinin M. O. Protection against adversarial attacks based on a dynamically reconfigurable ensemble of machine learning models. *Problems of information security. Computer systems*. 2026. No. 2, pp. 113–120. DOI: 10.66424/2071-8217-2026-2-9 (In Russian)

RECEIVED 27.04.2026
ACCEPTED 07.05.2026
PUBLICATION 15.06.2026

ABSTRACT

The paper reviews the problem of protecting machine learning models from adversarial attacks. A protection method is presented based on a dynamically reconfigurable ensemble of classifiers with a failure mechanism that combines a random combination of heterogeneous sub-models, online analysis of forecast variance, simulation of a plausible attack response, and a decoy model mechanism. Analysis of the consistency of outputs in the ensemble and failure to issue the most probable output reduces the effectiveness of an attacker when analyzing feedback received from the target model and generating adversarial samples. An experimental evaluation conducted on the UNSW-NB15 dataset showed that the developed method maintains high initial accuracy of the protected model under adversarial attacks (85–95 %) with a minimal decrease of 1–3 percentage points. The method can eliminate up to 98 % of attacks, significantly exceeding the performance of similar widely used methods.

KEYWORDS

Protection of machine learning, ensemble of models, classification, mechanism for rejecting, adversarial attacks

1. ВВЕДЕНИЕ

Машинное обучение стало неотъемлемой частью практически всех сфер человеческой деятельности. Технологии искусственного интеллекта активно применяются в здравоохранении, транспорте, финансах, кибербезопасности и во многих других областях, где они демонстрируют результаты, сопоставимые или превосходящие человеческие.

Как показывают результаты исследований, модели глубокого обучения могут значительно улучшить оказание медицинской помощи в части выявления и мониторинга прогнозов различных заболеваний [1]. В транспортной отрасли цифровые интеллектуальные технологии лежат в основе развития автономных транспортных средств и «умных» транспортных систем, включая адаптивное управление движением и анализ дорожных событий в реальном времени [2]. В финансовом секторе модели машинного обучения применяются для инвестиционного консультирования, управления рисками и обнаружения мошенничества, а также для обслуживания клиентов с помощью многоязычных чат-ботов [3]. Повсеместная интеграция механизмов машинного обучения в критические информационные инфраструктуры усилила привлекательность новых цифровых сервисов для злоумышленников и спровоцировала рост специфических атак на эти технологии. Одними из ключевых являются атаки извлечения данных, моделей машинного обучения, а также атаки уклонения и отравления моделей [4–8]. Разнообразие новых классов атак на машинное обучение и их широкий спектр воздействия определили разработку и внедрение новых подходов и средств защиты, которые смогли бы противодействовать новой угрозе безопасности (табл. 1).

Несмотря на разнообразие существующих решений они имеют общие характерные недостатки: вынужденный компромисс между точностью, устойчивостью и вычислительной эффективностью и недостаточную адаптивность. Это обусловливает необходимость создания легких, гибких и действенных защитных механизмов, которые способны обнаруживать и нейтрализовать как известные, так и ранее не встречавшиеся атаки без существенного снижения производительности защищаемой модели [18].

Для преодоления ограничений известных подходов и повышения устойчивости к адаптивным атакам, универсальности и результирующей точности защищаемой модели машинного обучения авторами предложен метод защиты моделей машинного обучения, основанный на использовании динамически перестраиваемого ансамбля классификаторов с механизмом отказа.

Таблица 1 | Сопоставление атак на модели машинного обучения и методов защиты

Table 1 | Matching of attacks on machine learning models and protection methods

Разновидности атак	Методы защиты
Атака уклонения	1. Защитное дистиллирование [9, 10] усложняет вычисление градиентов. 2. Сжатие признаков [11], обфускация входа [12, 13], защита на основе рандомизации [9, 14], упрощение выходных данных [15] затрудняют подбор нарушителем входных данных. 3. Ограничение скорости запросов [15] замедляет нарушителя
Атака отравления	1. Дифференциальная приватность [9, 14] снижает влияние выбросов. 2. Безопасные вычисления для нескольких сторон [9, 16] при совместном обучении понижают риск целенаправленных атак на конкретные данные
Атака извлечения данных	1. Дифференциальная приватность затрудняет понимание извлеченных данных. 2. Гомоморфное шифрование [9] обеспечивает сокрытие данных даже в случае их извлечения
Атака извлечения модели	1. Ограничение скорости запросов делает процесс извлечения непрактично долгим для нарушителя. 2. Упрощение выходных данных снижает точность восстановления, значительно увеличивает необходимое число запросов нарушителем. 3. Внедрение паспортных слоев [12, 17] делает извлечение бесполезным для нарушителя, поскольку без ключа скопированная модель выдает бессмысленные результаты

2. МЕТОДЫ

Разработанный метод объединяет защитные механизмы на архитектурном уровне и реализует активную стратегию противодействия (см. рисунок).

Целевая модель машинного обучения представлена в виде пула из N подмоделей-экземпляров, каждый из которых обучен на модифицированных данных с использованием методов аугментации, добавления шума, отбора подмножеств признаков либо обладает отличной от других архитектурой. Все подмодели-экземпляры решают одну и ту же задачу. Вариативность ансамбля усложняет проведение атаки нарушителем, поскольку наличие некорректных данных среди обучающих не обеспечивает искажение каждой подмодели из ансамбля.

Для каждого легитимного запроса ансамбль случайным образом формирует подмножество из K моделей ($K<N$), агрегирует их прогнозы посредством усреднения и выдает итоговый прогноз общей модели. Случайный характер комбинации подмоделей в ансамбле и гетерогенность подмоделей в пуле существенно затрудняют для нарушителя создание универсального состязательного примера, эффективного в произвольный момент времени.

Механизм отказа функционирует на уровне обработки данных. Параллельно с основным вычислительным процессом каждый входной вектор X направляется в модуль согласованности, который пропускает его через все N подмоделей пула и вычисляет дисперсию полученных предсказаний. Для легитимных данных все подмодели, обученные на едином наборе, демонстрируют высокую согласованность. В данном случае дисперсия будет низкой, и запрос будет классифицирован как доверенный, и пользователю возвращается прогноз, полученный от быстрого ансамбля из K моделей.

При подаче состязательного примера малые, незаметные для человека искажения по-разному дестабилизируют различные подмодели анасамбля, что приводит к росту дисперсии прогнозов от подмоделей. При превышении заданного порога срабатывает механизм отказа. Система не выдает явного сообщения об ошибке, а имитирует нестандартное, но правдоподобное поведение (например, в задачах

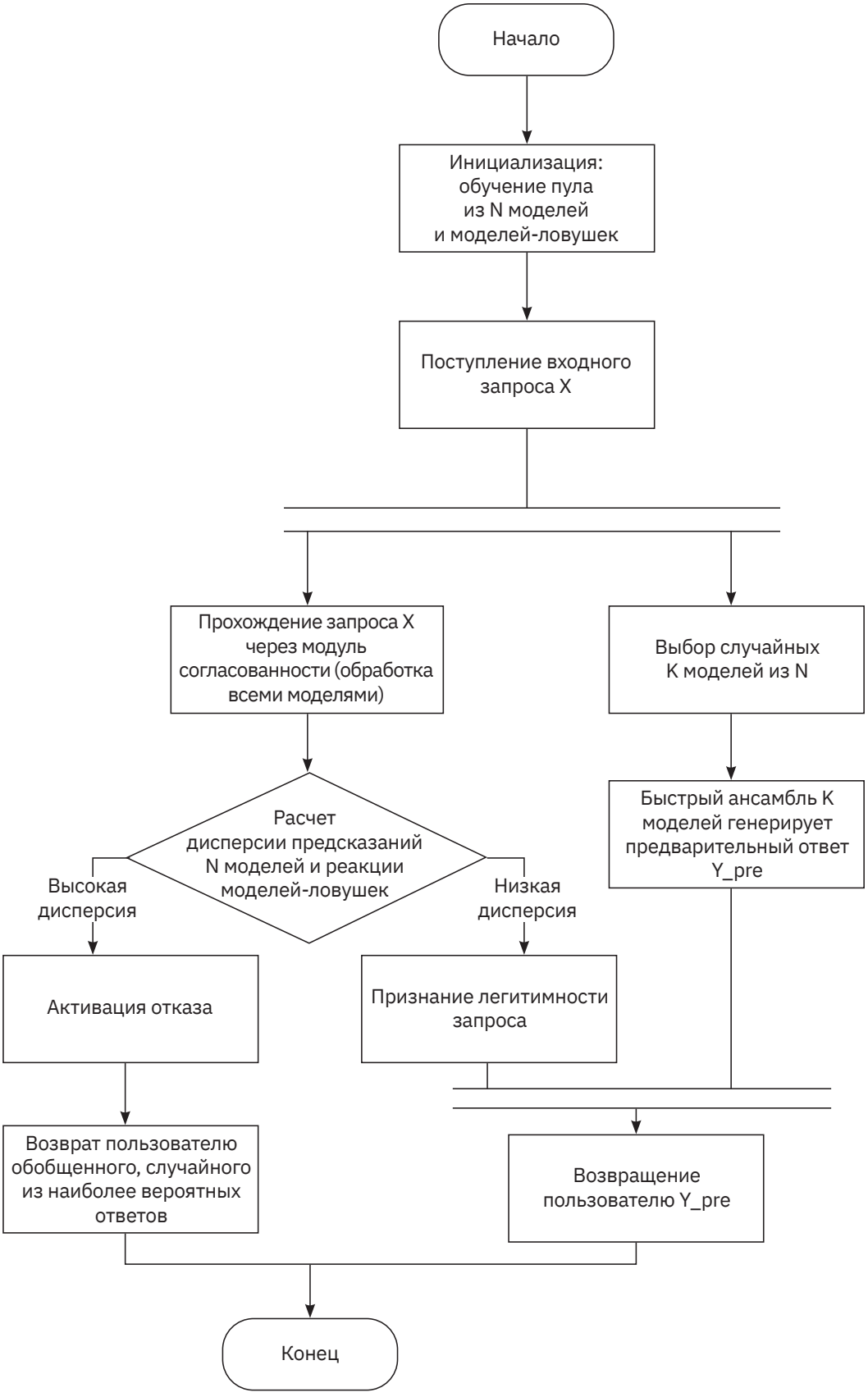


Схема разработанного метода

Scheme of the developed method

классификации ансамбль возвращает случайный, но семантически близкий ответ из числа наиболее вероятных классов). Тем самым нарушитель лишается устойчивого сигнала обратной связи (например, градиента), необходимого ему для итеративной корректировки состязательны образцов, что значительно усложняет эксплуатацию атак извлечения данных и извлечения модели.

Дополнительно в данном методе используется уровень подмоделей-ловушек. В ансамбль включаются одна или несколько специально обученных «хрупких» подмоделей-ловушек, обладающих повышенной чувствительностью к незначительным искажениям данных. Их аномальная реакция, а именно резкое отклонение выдаваемых прогнозов, служит триггером для активации механизма отказа даже в тех случаях, когда основные подмодели еще не фиксируют значимой дисперсии. Обучение подмоделей-ловушек производится либо на обратной задаче предсказания заведомо неверного, но детерминированного класса для части данных, либо на состязательных выборках данных. В результате при предъявлении нового состязательного образца подмодель-ловушка с высокой вероятностью активирует указанный атакующий класс либо выдает хаотичный прогноз, существенно расходящийся с предсказаниями основных моделей.

3. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Для экспериментальной оценки разработанного метода построен ансамбль из восьми моделей, включающий рекуррентные нейросети, многослойный персептрон, трансформеры и модели-ловушки по две модели каждого типа. Подмножество быстрого вывода составляет три подмодели ($K=3$). Тестирование проведено на наборе данных UNSW-NB15, содержащем обширный набор примеров атак (фаззеры, бэкдоры, DoS-атаки, эксплойты, общие атаки, разведывательные атаки, шеллкоды и черви). Результаты анализа эффективности разработанного метода представлены в табл. 2. Предложенное решение

Таблица 2 | Анализ применения методов на наборе данных UNSW-NB15

Table 2 | Analysis of the application of methods in the UNSW-NB15 dataset

Метод	Исходная точность защищаемой модели при воздействии атаки, %	Снижение точности защищаемой модели при внедрении метода защиты при воздействии атаки, п. п.	Прирост вычислительной стоимости (процент времени), %	Доля устраняемых атак, %
Разработанный метод	85–95	1–3	15–25	95–98
Гомоморфное шифрование/безопасные вычисления для нескольких сторон	95 (для конфиденциальности данных)	0	≥1000	86–92
Дифференциальная приватность	70–90	3–10	10–25	88–90
Защитное дистиллярование	30–70	2–8	<5	30–35
Обфускация входа/рандомизация	20–60	1–5	1–10	1–10
Сжатие признаков	10–40	2–8	1–5	1–10
Упрощение выходов/ограничение скорости запросов	0–30	0–4	<1	27–35
Дополнительные слои	60–80	0,5–2,0	<2	90–92

обеспечивает баланс между точностью, устойчивостью и эффективностью, кроме того, усложняет задачу компрометации модели при попытке создания и эксплуатации нарушителем состязательных примеров.

4. ЗАКЛЮЧЕНИЕ

Полученные результаты свидетельствуют о том, что предложенный метод обеспечивает высокую точность защищаемой модели под воздействием атаки (85–95 %) при минимальном ее снижении на 1–3 п. п. и позволяет устранить до 98 % атак, что значительно превосходит показатели таких популярных аналогов, как защитное дистиллирование и дифференциальная приватность. При этом вычислительные затраты возрастают лишь на 15–25 %, тогда как, например, гомоморфное шифрование требует более чем 1000 %-ного увеличения времени обработки, что подтверждает достижение методом наилучшего баланса между точностью, устойчивостью и эффективностью среди рассмотренных решений.

Разработанный метод реализует активную стратегию, затрудняющую для нарушителей процесс генерации и подбора состязательных примеров по сравнению с известными аналогами за счет достигнутого эффекта скрытности. В качестве перспективного направления дальнейших исследований рассматривается адаптация разработанного метода для использования в легковесных вычислительных системах.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Okeibunor J. C., Jaca A., Iwu-Jaja C. J. et al.** The use of artificial intelligence for delivery of essential health services across WHO regions: a scoping review // Frontiers in Public Health. 2023. Vol. 11. P. 1102185. DOI: 10.3389/fpubh.2023.1102185.
2. **Buenaventura M., Shenk A., Nergui A. et al.** Artificial Intelligence Adoption and Sectoral Transformation: Implications for Health Care, Financial Services, Climate and Energy, and Transportation. 2025. № RR-A3888-1. DOI: 10.7249/rra3888-1.
3. **Беспалов Д. А., Богатырева М. В.** Роль искусственного интеллекта в финансовом секторе // Вестник Алтайской академии экономики и права. 2023. № 7–1. С. 10.
4. **Abomakhelb A., Jalil K. A., Buja A. G. et al.** A Comprehensive Review of Adversarial Attacks and Defense Strategies in Deep Neural Networks // Technologies. 2025. Vol. 13. № 5. P. 202. DOI: 10.3390/technologies13050202.
5. **Жуковский Е. В., Огнев Р. А.** Анализ возможности реализации состязательных атак на средства проактивной защиты, использующие машинное обучение // Методы и технические средства обеспечения безопасности информации. 2021. № 30. С. 28–29. DOI: 10.31799/2949-0693-2023-161-71.
6. **Беззатеев С. В., Афанасьева А. В., Супрун А. Ф.** Атаки на обучающие выборки в системах машинного обучения и защита от них // Инновационное приборостроение. 2023. Т. 2. № 1. С. 61–71.
7. **Jha P. K.** Adversarial Machine Learning: Attacks, Defenses, and Open Challenges // arXiv preprint arXiv:2502.05637. 2025.
8. **Намиот Д. Е.** Введение в атаки отравлением на модели машинного обучения // International Journal of Open Information Technologies. 2023. Т. 11. № 3. С. 58–68.
9. **El-Husseini F., Noura H. N., Vernier F.** Security and privacy-preserving for machine learning models: attacks, countermeasures, and future directions // Annals of Telecommunications. 2025. P. 1–22. DOI: 10.1109/CSNet64211.2024.10851722.

10. **Kuzlu M., Catak F. O., Cali U. et al.** Adversarial security mitigations of mmWave beamforming prediction models using defensive distillation and adversarial retraining // International Journal of Information Security. 2023. Vol. 22. № 2. P. 319–332.
11. **Xu W., Evans D., Qi Y.** Feature squeezing: Detecting adversarial examples in deep neural networks // arXiv preprint arXiv:1704.01155. 2017.
12. **Lederer I., Mayer R., Rauber A.** Identifying appropriate intellectual property protection mechanisms for machine learning models: A systematization of watermarking, fingerprinting, model access, and attacks // IEEE Transactions on Neural Networks and Learning Systems. 2023. Vol. 35. № 10. P. 13082–13100.
13. **Chen M., Wu M.** Protect your deep neural networks from piracy // 2018 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2018. P. 1–7.
14. **Lecuyer M., Atlidakis V., Geambasu R. et al.** Certified robustness to adversarial examples with differential privacy // 2019 IEEE symposium on security and privacy (SP). IEEE, 2019. P. 656–672.
15. **Isakov M., Bu L., Cheng H., Kinsy M. A.** Preventing neural network model exfiltration in machine learning hardware accelerators // 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). IEEE, 2018. P. 62–67.
16. **Tiwari S. S., Dhasmana G., Al-Jawahry H. M. et al.** Federated Learning Strategies for Privacy-Preserving Machine Learning Models in Cloud Computing Environments // 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE). IEEE, 2024. P. 1457–1462. DOI: 10.1109/IC3SE62002.2024.10593458.
17. **Fan L., Ng K. W., Chan C. S.** Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks // Advances in neural information processing systems. 2019. Vol. 32. DOI: 10.48550/arXiv.1909.07830.
18. **Югай П. Э., Москвин Д. А.** Способы выявления состязательных атак на алгоритмы машинного обучения в системах обнаружения вторжений // Методы и технические средства обеспечения безопасности информации. 2023. № 32. С. 21–22.

REFERENCES

1. **Okeibunor J. C., Jaca A., Iwu-Jaja C. J. et al.** The use of artificial intelligence for delivery of essential health services across WHO regions: a scoping review. *Frontiers in Public Health*. 2023. Vol. 11, pp. 1102185. DOI: 10.3389/fpubh.2023.1102185.
2. **Buenaventura M., Shenk A., Nergui A. et al.** Artificial Intelligence Adoption and Sectoral Transformation: Implications for Health Care, Financial Services, Climate and Energy, and Transportation. 2025. No. RR-A3888-1. DOI: 10.7249/rra3888-1.
3. **Bespalov D. A., Bogatyreva M. V.** The role of artificial intelligence in the financial sector. *Journal of the Altai Academy of Economics and Law*. 2023. No. 7–1, pp. 10. (In Russian)
4. **Abomakhelb A., Jalil K. A., Buja A. G. et al.** A Comprehensive Review of Adversarial Attacks and Defense Strategies in Deep Neural Networks. *Technologies*. 2025. Vol. 13. No. 5, pp. 202. DOI: 10.3390/technologies13050202.
5. **Zhukovsky E. V., Ognev R. A.** Analysis of the possibility of implementing adversarial attacks on proactive defense tools using machine learning. *Metody i tehnicheskie sredstva obespecheniya bezopasnosti informacii*. 2021. No. 30. C. 28–29. DOI: 10.31799/2949-0693-2023-161-71. (In Russian)
6. **Bezzateev S. V., Afanasyeva A. V., Suprun A. F.** Attacks on data sets in machine learning systems and protection against them. *Innovacionnoe priborostroenie = Innovative Instrumentation*. 2023. Vol. 2. No. 1, pp. 61–71. DOI: 10.31799/2949-0693-2023-161-71. (In Russian)
7. **Jha P. K.** Adversarial Machine Learning: Attacks, Defenses, and Open Challenges. *arXiv preprint arXiv:2502.05637*. 2025.
8. **Namiot D. E.** Introduction to poisoning attacks on machine learning models. *International Journal of Open Information Technologies*. 2023. Vol. 11. No. 3, pp. 58–68. (In Russian)
9. **El-Husseini F., Noura H. N., Vernier F.** Security and privacy-preserving for machine learning models: attacks, countermeasures, and future directions. *Annals of Telecommunications*. 2025, pp. 1–22. DOI: 10.1109/CSNet64211.2024.10851722.

10. **Kuzlu M., Catak F. O., Cali U. et al.** Adversarial security mitigations of mmWave beamforming prediction models using defensive distillation and adversarial retraining. *International Journal of Information Security*. 2023. Vol. 22. No. 2, pp. 319–332.
11. **Xu W., Evans D., Qi Y.** Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*. 2017.
12. **Lederer I., Mayer R., Rauber A.** Identifying appropriate intellectual property protection mechanisms for machine learning models: A systematization of watermarking, fingerprinting, model access, and attacks. *IEEE Transactions on Neural Networks and Learning Systems*. 2023. Vol. 35. No. 10, pp. 13082–13100.
13. **Chen M., Wu M.** Protect your deep neural networks from piracy. 2018 IEEE International Workshop on Information Forensics and Security (WIFS). IEEE, 2018, pp. 1–7.
14. **Lecuyer M., Atlidakis V., Geambasu R. et al.** Certified robustness to adversarial examples with differential privacy. 2019 IEEE symposium on security and privacy (SP). IEEE, 2019, pp. 656–672.
15. **Isakov M., Bu L., Cheng H., Kinsy M. A.** Preventing neural network model exfiltration in machine learning hardware accelerators. 2018 Asian Hardware Oriented Security and Trust Symposium (AsianHOST). IEEE, 2018, pp. 62–67.
16. **Tiwari S. S., Dhasmana G., Al-Jawahry H. M. et al.** Federated Learning Strategies for Privacy-Preserving Machine Learning Models in Cloud Computing Environments. 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE). IEEE, 2024, pp. 1457–1462. DOI: 10.1109/IC-3SE62002.2024.10593458.
17. **Fan L., Ng K. W., Chan C. S.** Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks. *Advances in neural information processing systems*. 2019. Vol. 32. DOI: 10.48550/arXiv.1909.07830.
18. **Yugai P. E., Moskvin D. A.** Methods of detecting adversarial attacks on machine learning algorithms in intrusion detection systems. *Metody i tehnicheskie sredstva obespecheniya bezopasnosti informacii*. 2023. No. 32, pp. 21–22. (In Russian)

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ГАВВА Георгий Дмитриевич – магистр, ассистент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: gavva_gd@spbstu.ru

GAVVA Georgij D. – Master's Student, assistant, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

КАЛИНИН Максим Олегович – д-р техн. наук, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: max@ibks.spbstu.ru
ORCID: 0000-0002-9732-0099

KALININ Maxim O. – Doctor of Engineering Sciences, Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья
DOI 10.66424/2071-8217-2026-2-10
УДК 004.04

ЗАЩИТА СИСТЕМ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ AI/ML ОТ АТАК ОТРАВЛЕНИЯ

М. А. Полтавцева*, А. А. Васильева

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

✉ *poltavtseva@ibks.spbstu.ru

ДЛЯ ЦИТИРОВАНИЯ

Полтавцева М. А., Васильева А. А. Защита систем федеративного обучения AI/ML от атак отравления // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 121–137. DOI: 10.66424/2071-8217-2026-2-10

ПОСТУПИЛА 03.03.2026
ПРИНЯТА 27.04.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Системы федеративного обучения искусственного интеллекта подвержены атакам, позволяющим злоумышленнику изменить их поведение, как и обычные AI/ML решения. Наиболее эффективной является атака отравления. При этом защита систем федеративного обучения усложняется возможностью сговора между участниками. В таких условиях обнаруживать и предотвращать атаки становится особенно трудно. Решение этой задачи является целью работы. Исследование предлагает метод обеспечения защиты систем федеративного обучения от атак отравления с использованием сговора, основанный на комбинации известных и доказавших свою эффективность методов защиты. Выбранные методы фильтрации и надежной агрегации модифицированы для учета возможного сговора участников обучения. Корректность и эффективность предложенного метода подтверждается практическими экспериментами, позволяющими не только доказать результативность, но и выявить ограничения разработанного решения.

КЛЮЧЕВЫЕ СЛОВА

Информационная безопасность, искусственный интеллект, машинное обучение, цепочки поставок, атаки отравления

Original article
DOI 10.66424/2071-8217-2026-2-10

PROTECTION OF AI/ML FEDERATED LEARNING SYSTEMS FROM POISONING ATTACKS

M. A. Poltavtseva*, A. A. Vasilyeva

Peter the Great St. Petersburg Polytechnic University, St. Petersburg, Russia

✉ *poltavtseva@ibks.spbstu.ru

FOR CITATION

Poltavtseva M. A., Vasilyeva A. A. Protection of AI/ML federated learning systems from poisoning attacks. *Problems of information security. Computer systems*.

ABSTRACT

Federated artificial intelligence learning systems are susceptible to attacks that allow an attacker to change their behavior, just like conventional AI/ML solutions. The most effective of such attacks today is the poisoning attack. At the same time, the protection of federated learning systems is complicated by the possibility of collusion between the

2026. No. 2, pp. 121–137.
DOI: 10.66424/2071-8217-2026-2-10
(In Russian)

RECEIVED 03.03.2026
ACCEPTED 27.04.2026
PUBLICATION 15.06.2026

participants. In such circumstances, it becomes especially difficult to detect and prevent attacks. The solution of this problem is the purpose of the presented work. The study suggests a method to ensure the protection of federated learning systems from poisoning attacks using collusion, based on a combination of known and proven protection methods. The selected methods of filtering and reliable aggregation have been modified to take into account possible collusion of the training participants. The correctness and effectiveness of the proposed method is confirmed by practical experiments, which make it possible not only to prove its effectiveness, but also to identify the limitations of the developed solution.

KEYWORDS

Information security, artificial intelligence, machine learning, supply chains, poisoning attacks

1. ВВЕДЕНИЕ

Технологии искусственного интеллекта и методы машинного обучения используются повсеместно, на данный момент уже более 45 % предприятий применяют ту или иную форму искусственного интеллекта в своей работе. Эти технологии открывают не только возможности, но и потенциальные риски. В таких системах существует риск нарушения безопасности и конфиденциальности.

Рост популярности моделей распределенного машинного обучения, в том числе с сохранением конфиденциальности, обусловлен удобством и эффективностью такого подхода в B2B решениях. Однако системы федеративного обучения искусственного интеллекта, призванные защитить приватность пользователей и конфиденциальность данных участников, сами подвержены наборам специфических атак [1]. Отдельной проблемой является проведение традиционных атак, доказавших свою эффективность, в условиях сговора двух и более участников процесса обучения общей модели.

Атаки на системы федеративного обучения AI/ML возможны всеми известными способами: через данные, модель, используемые программные пакеты, технические средства, использующиеся при создании и работе AI/ML-систем. Тем не менее одной из самых разнообразных и опасных остается атака отправления данных или модели [2]. Атаки, направленные на системы, использующие либо предварительно обученные модели, либо распределенные системы обучения, такие как федеративное обучение, являются наиболее новыми и опасными, так как в них не представляется возможным проверить сами данные, на которых происходило обучение.

Данная работа направлена на поиск и совершенствование методов обеспечения безопасности систем федеративного обучения от атаки отравления в условиях возможного сговора участников.

2. БЕЗОПАСНОСТЬ СИСТЕМ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Можно выделить множество различных атак на системы федеративного обучения искусственного интеллекта [3, 4], но одной из наиболее эффективных остается атака отравления [5–7]. Систематизация характеристик атаки приведена в табл. 1.

Для федеративного обучения существует множество видов защиты, но основные способы защиты от атак отравления – это защита на основе доверия (фильтры) [8] и при помощи надежного агрегирования [9].

Методы защиты на основе доверия основаны на том, что результаты обучения, передаваемые серверу у вредоносных

клиентов, отличаются от доброкачественных. И в большинстве случаев при нахождении вредоносного клиента его влияние на модель в дальнейшем обнуляется [10]. Главный недостаток данного метода в том, что при увеличении количества вредоносных клиентов данные методы защиты становятся неэффективными.

Защита на основе надежной агрегации основана на проверке характеристик обновлений, присланных клиентами, и усреднении всех параметров на основе медианы, математическом ожидании и т.п. Недостатком данных методов является то, что зачастую влияние атакующего не удаляется полностью и в большинстве случаев точность глобальной модели понижается. Обзор основных методов защиты представлен в табл. 2.

В результате анализа принято решение разрабатывать систему защиты на основе двух методов различных категорий: фильтрации и надежной агрегации, для противодействия максимальному числу

Таблица 1 | Атака отравления в федеративном обучении AI/ML

Table 1 | Poisoning attack in AI/ML federated learning

Атака		Место внедрения	Особенности атаки	Последствия
Отравление	Данные	Через данные, поступающие на вход модели при обучении	При использовании данных от третьих лиц или при атаке на источник данных	Понижение точности, полное забывание модели, установка триггеров
	Модель	Через предварительно обученную модель	При использовании трансферного, аутсорсингового и федеративного обучения	

Таблица 2 | Методы защиты от атаки отравления в федеративном обучении AI/ML

Table 2 | Methods of protection against poisoning attacks in AI/ML federated learning

Защита	Тип защиты	Снижение точности модели, %
Защита, воздействующая на параметры моделей [11]	Модификация параметров модели	Менее 1
FL-WBC [8]	Надежное обучение	3–10
PELF [12]	Фильтр	Менее 1
FedDefender [13,14]	Фильтр	Менее 1
SignGuard [15, 16]	Надежная агрегация с фильтрацией	1–4
MultiKrum [17, 18]	Надежная агрегация	1–5
Trimmed Mean [19, 20]	Надежная агрегация	1–3
Centered clip [21]	Надежная агрегация	Менее 1
Защита на основе кластеризации (DnC) [22]	Надежная агрегация	1–9

атак отравления с использованием сговора. За основу дальнейшей работы взяты методы FedDefender и надежного агрегирования Centered Clipping, так как они являются наиболее эффективными и проработанными из рассмотренных.

3. МЕТОД ЗАЩИТЫ СИСТЕМ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ ОТ АТАК ОТРАВЛЕНИЯ

В данном случае метод защиты разрабатывается для систем федеративного обучения, при следующей модели угроз: рассматривается атака отравления весами, где количество вредоносных клиентов меньше 50 %. При числе вредоносных клиентов больше этого порога обеспечение защиты уже не представляется возможным.

Идея разрабатываемой защиты заключается в объединении метода, основанного на доверии, со способом надежного агрегирования для снижения влияния атаки на точность модели. Методы FedDefender и Centered Clipping будут применяться при передаче на сервер именно весов обученных моделей клиентов, а не градиентов, что получались в каждую эпоху. Данный способ является наиболее применяемым на практике и при этом достаточно безопасным, так как не передается информация, при помощи которой можно восстановить данные клиента. Схема предложенного способа защиты приведена на рис. 1.

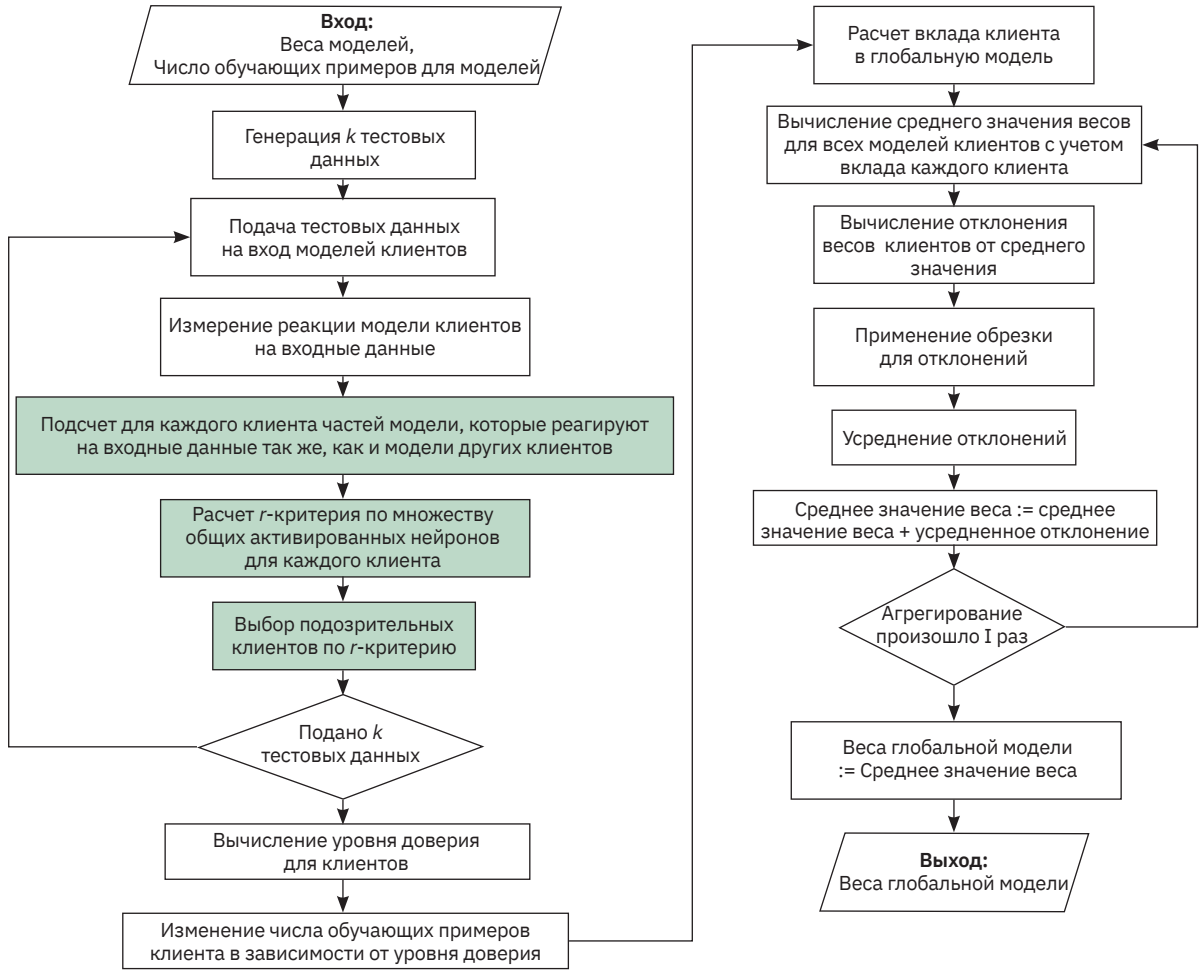


Рис. 1 | Схема предложенного способа защиты

Fig. 1 | Scheme of the proposed protection method

В предложенном способе защиты сначала происходит фильтрация клиентов на основе значения активированных нейронов. Для данного действия выполняется ряд шагов:

1. Получение тестовых данных, т.е. генерация *k* данных, подходящих для тестирования обучающейся модели.

2. Определение на каждом экземпляре тестовых данных для всех моделей клиентов активированных нейронов на каждом слое модели.

3. Определение общих активированных нейронов для каждого клиента с другими $n-1$ клиентами.

4. Определение вредоносности клиента на основе полученных данных для каждого тестового экземпляра данных.

5. Отсеивание клиента, если доверие к нему ниже 60 %, т.е. более чем по 40 % тестовых данных клиент признан вредоносным.

В данном способе защиты в четвертом пункте определяется вредоносность клиента. В работе данное действие предложено выполнить способом, основанным на исключении выбросов при неизвестном σ. Это делается при помощи *r*-критерия. Берется $x_1, \ldots, x_n$ характеризующие количество нейронов, активированных у клиента совместно с другими клиентами на одном и том же тестовом входе. Для этого проводится пересечение бинарных масок активации между всеми клиентами, затем для каждого клиента считается число нейронов, совпадающих с этим общим шаблоном. По ним вычисляется среднее значение:

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}, \quad (1)$$

где n – количество клиентов; x_i – количество нейронов, активированных у клиента совместно с другими клиентами на одном и том же тестовом входе; $\overline{x}$ – среднее количество общих активированных нейронов на одного клиента.

Так же высчитывается стандартное отклонение:

$$S = \sqrt{\frac{\sum_{i=1}^{n}\left(x_i - \overline{x}\right)^2}{n-1}}. \quad (2)$$

Далее высчитывается *r*-критерий для значения каждого клиента. Расчет ведется по формуле

$$r_i = \frac{\left|x - \overline{x}\right|}{S\sqrt{\frac{n-1}{n}}}, \quad (3)$$

где S – стандартное отклонение.

По полученным значениям r_i определяется вредоносный клиент или нет, выяснение ведется только для клиентов, чьи значения общих активированных нейронов ниже, чем среднее значение $\overline{x}$. Клиент считается вредоносным на основе оценки следующих условий:

$$r_i > r\left(0{,}01; f = n-2\right), \quad (4)$$

$$r_i > r\left(0{,}05; f = n-2\right), \quad (5)$$

$$\left|r_i - r\left(0{,}05; f = n-2\right)\right| > \left|r_i - r\left(0{,}01; f = n-2\right)\right|, \quad (6)$$

где $r\left(0{,}05; f = n-2\right)$ и $r\left(0{,}01;\ f = n-2\right)$ являются табличными значениями и выбираются по уровню значимости и числу степеней свободы *f*, который рассчитывается с учетом числа клиентов; r_i – рассчитанный *r*-критерий для клиента *i*.

Вредоносным считается клиент, соответствующий условию (4) или условиям (5) и (6) совместно. В данном случае используется обновленный способ фильтрации, по сравнению с FedDefender, в логике определения вредоносного клиента. Вместо разбиения множества на подмножества из $n-1$ клиентов и нахождения из них доброкачественного выполняется определение клиентов с наибольшим количеством общих активированных нейронов.

После фильтрации применяется метод надежного агрегирования. Чтобы в способе агрегации были учтены результаты фильтра перед ним, делается подсчет коэффициентов для клиентов и нормализации их весов.

По результатам фильтра, изменяется число локальных примеров (num_examples), использованных клиентом при обучении, поэтому, если уровень доверия клиента меньше 0,6, то его num_examples = 0, в ином случае его num_examples = уровень_доверия * num_examples. Далее вычисляется общее число обучающих примеров по всем клиентам.

Для нормализации весов клиентов для каждого из них высчитывается параметр:

$$\alpha_j = \frac{\text{число локальных примеров клиента}}{\text{общее число обучающих примеров}}. \quad (7)$$

Вместо метода агрегации FedAvg и его модификаций [23, 24] в данной разработке использован Centered Clipping [25, 26], чтобы все атаки, которые проходят через фильтр, были либо нейтрализованы, либо смягчены. Centered Clipping выполнен с учетом входных данных в виде параметров обученных клиентских моделей, т.е. с учетом весов, а не с учетом градиентов моделей клиентов. В итоге модифицированный метод надежной агрегации включает несколько шагов.

Во-первых, получение центра полученных весов, т.е. вычисляется среднее от весов клиентов:

$$\overline{W_k} = \frac{1}{n}\sum_{j=1}^{n}\alpha_j W_j, \quad (8)$$

где W_j – параметры модели клиента; α_j – вес доверия клиента после фильтра.

Во-вторых, вычисление отклонения весов клиента от центра:

$$d_j = W_j - \overline{W_k}, \quad (9)$$

где $\overline{W_k}$ – центр усредненная модель на итерации k.

Далее применяется обрезка вида:

$$\overline{d_j} = \begin{cases} d_j & \text{если } \tau \geq \|d_j\| \\ \tau\dfrac{d_j}{\|d_j\|} & \text{иначе} \end{cases}, \quad (10)$$

где d_j – отклонение веса клиента j от центра; τ – коэффициент нормализации.

И усреднение отклонений, полученных после обрезки:

$$\bar{d} = \frac{1}{n}\sum_{j=1}^{n}\overline{d_j}, \quad (11)$$

где $\overline{d_j}$ – обрезанный вектор отклонения для клиента j.

В-третьих, проводится восстановление откорректированного веса:

$$\overline{W_{k+1}} = \overline{W_k} + \bar{d}, \quad (12)$$

где $\bar{d}$ – среднее значение обрезанных отклонений от центра.

Данный алгоритм повторяется I раз, т.е. $k = 1, \ldots, I$. Параметры K и τ назначаются разработчиком на основе предварительного анализа и подбора.

В итоге предложенный способ реализует синергию методов на основе доверия, в частности метода фильтрации, и методов агрегации, который в свою очередь основан на усреднении параметров, присылаемых клиентами серверу. Предложенный способ защиты предполагает, что количество атакующих может быть больше 1 и при этом меньше 50 % от общего количества клиентов и предполагает, что серверу от клиентов приходит только стандартная информация, а именно веса моделей и количество данных, на которых проведено обучение.

4. РАЗРАБОТКА ПРОГРАММНОГО СТЕНДА И ПОСТАНОВКА ЭКСПЕРИМЕНТА

Тестовый стенд реализован на языке Python, основными использованными библиотеками были tensorflow, Flower, torchvision и numpy. Сама система запускалась на ОС Linux с использованием CUDA. Для использования технологии распараллеливания CUDA установлен драйвер и необходимая библиотека для взаимодействия с ним. Архитектура развернутой системы федеративного обучения приведена на рис. 2.

Само обучение системы происходит по следующему сценарию:

- сервер отправляет клиенту параметры глобальной модели;
- клиент обновляет локальную модель параметрами, полученными от сервера;
- клиент обучает модель на локальных данных, что изменяет параметры модели локально;
- клиент отправляет обновленные/измененные параметры модели обратно на сервер;
- сервер получает данные с клиентов и объединяет их для получения параметров глобальной модели.

Все повторяется с шага (1) пока не будут пройдены все раунды (num_round) обучения.

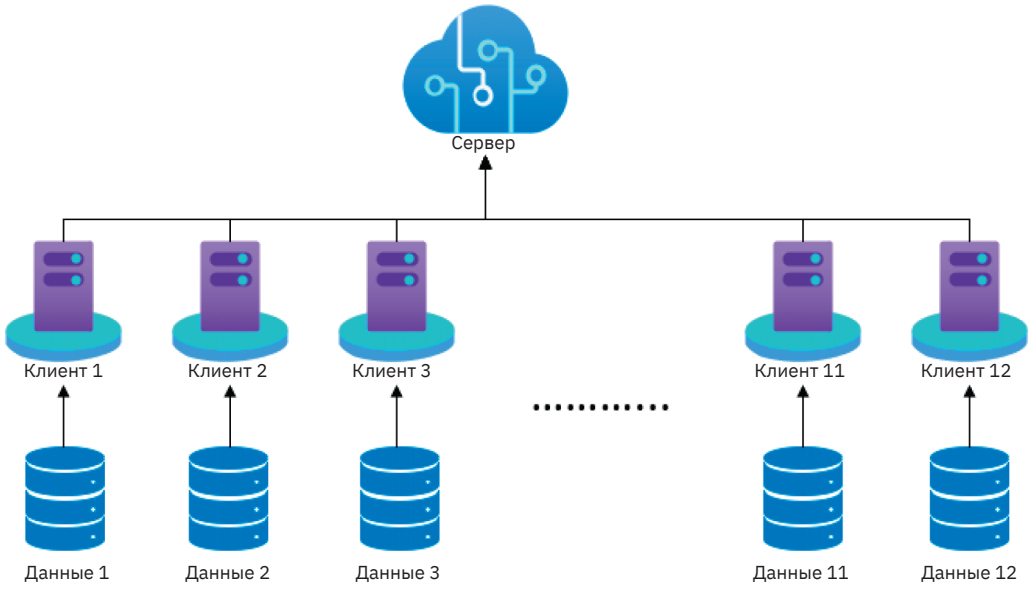


Рис. 2 | Развернутая система федеративного обучения

Fig. 2 | The expanded system of federated education

В самом коде запуска клиента создается образ локальной модели клиента, принимаются тестовые и обучающие данные для каждого клиента, проводятся атаки отравления для вредоносных клиентов и создается образ самого клиента и его запуск.

Вся симуляция федеративного обучения включает функции для инициализации клиентов, количество клиентов, участвующих в обучении, количество раундов, стратегия агрегации и задания места, где происходит обучение клиентов (GPU или CPU).

Для тестирования и обучения реализована CNN модель. Модель содержит шесть слоев. На первом слое создается шесть выходных каналов, используя ядро размером 5×5. На втором слое уменьшается размерность входных данных из первого слоя. Третий слой принимает шесть входных каналов и создает 16 выходных каналов с использованием ядра размером 5×5. На четвертом слое принимает входные данные с размерностью 16×5×5 и выводит 120 нейронов. Пятый слой принимает 120 входов и выводит 84 нейрона. Шестой слой принимает 84 входа и выводит 10 нейронов.

Реализовано несколько атак отравления для тестирования эффективности метода защиты. Первая атака – бэкдор-атака, которая реализуется изменением части существующих данных. В изображение вводится квадрат определенного цвета, например, белого, который незаметен человеческому глазу, но заметен компьютеру. Данный квадрат является триггером, при помощи которого злоумышленник может управлять действиями модели.

Второй атакой представлена одна из стандартных атак отравления – атака случайного переворота меток, не зависящая от вида модели. В ней при заданном обучающем наборе $\{x_i, y_i\} = 1$, где $x_i \in X$ и $y_i \in \{-1,1\}$, злоумышленник может случайным образом выбрать $[np]$ обучающих меток и перевернуть их.

Для предложенного способа защиты тестирование проводилось при 12 клиентах, 8 раундах и 10 эпох в раунде. Использовались датасеты MNIST и CIFAR10. Также проведен ряд тестов для определения наиболее подходящих параметров в фазе фильтрации и в фазе агрегации.

В методе агрегации есть два параметра это τ и I. Для подбора наиболее подходящих значений для них проведен ряд тестов. Сначала происходил подбор параметра τ, затем параметра I. При тестировании использовалась атака переворачивания

меток с двумя атакующими. Для сравнения параметров использовались значения точности, полученной в результате обучения модели. Подбор параметра τ велся на интервале от (0, 3], так как значения отклонения от центра в большинстве случаев не превышали значения 4 для используемых наборов данных, и после достижения значения $\tau = 2,5$ точность модели начинала падать. Все измерения проведены по три раза для получения точных результатов. На рис. 3 полученные результаты представлены в виде графиков.

По результатам тестирования сделан вывод, что данный параметр сильно зависит от данных. Его подбор необходимо

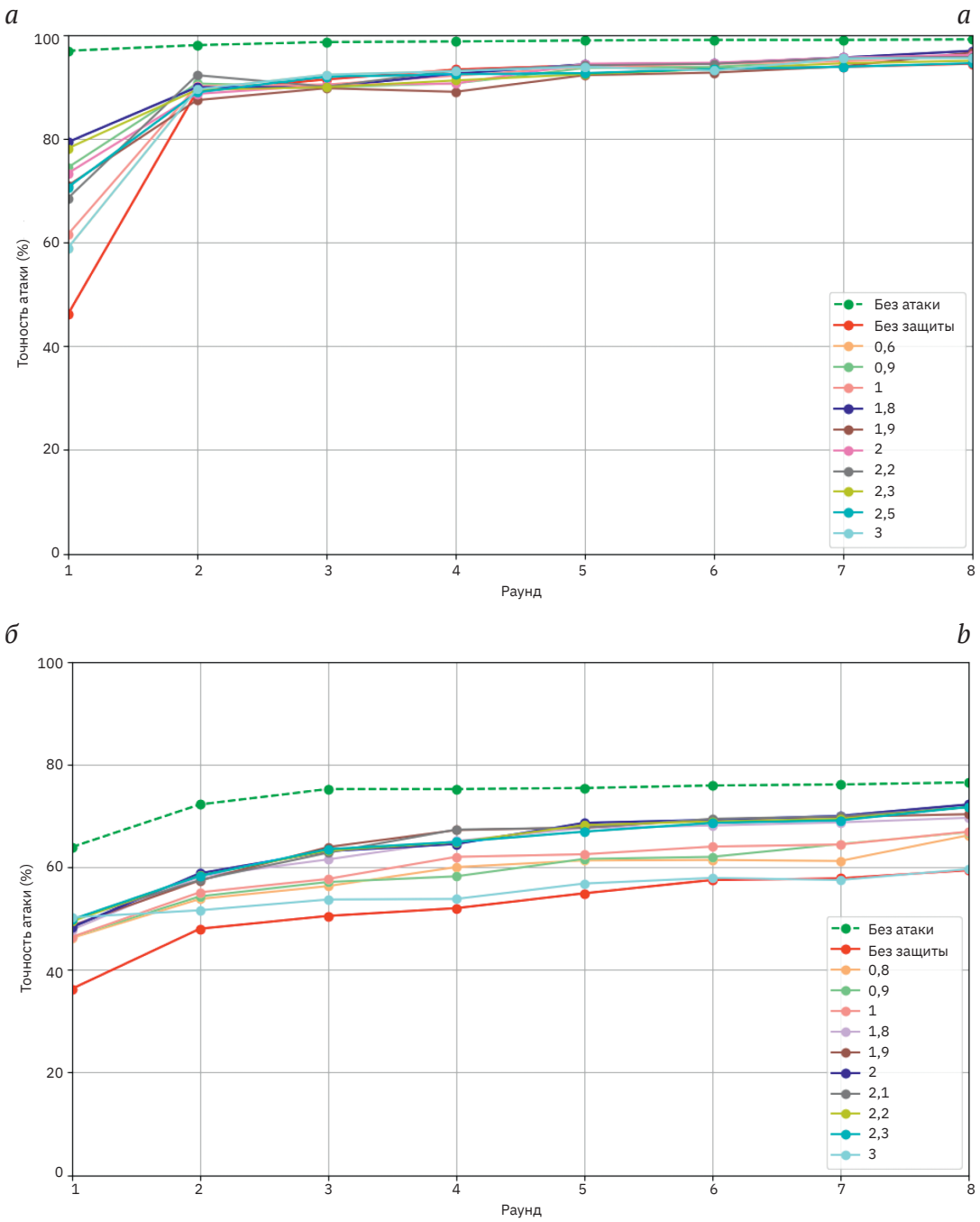


Рис. 3 | Подбор параметра τ для набора данных MNIST (*а*) и CIFAR10 (*б*)

Fig. 3 | Selection of the parameter τ for the MNIST (*a*) and CIFAR10 (*b*) data sets

вести в зависимости от значений отклонения от центра, т.е. от высчитываемого среднего. Замечено, что брать слишком маленькие значения, т.е. из интервала (0, 0,5], не имеет смысла, так как слишком сильно уменьшается влияние на модель не только вредоносных клиентов, но и нормальных. Наилучшими параметрами для датасетов MNIST и CIFAR10 оказались параметры из интервала (1, 2], а именно $\tau = 1{,}8$ и $\tau = 2{,}0$.

Подбор параметра I велся на интервале от [1, 10], так как далее изменения в работе алгоритма были не существенны. При этом параметр τ был равен для каждого датасета значениям, которые выбраны при помощи предыдущих тестов. На рис. 4

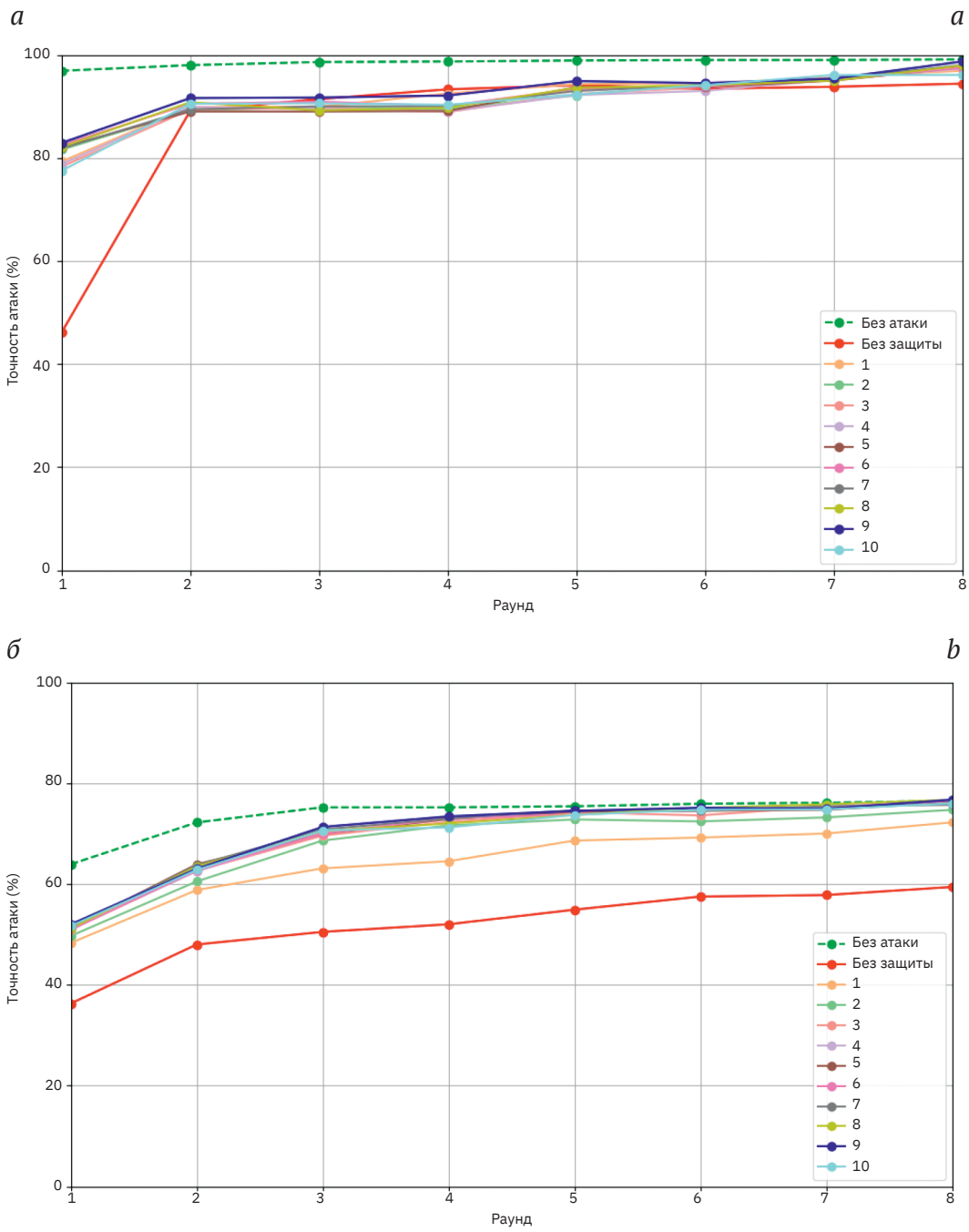


Рис. 4 | Подбор параметра I для набора данных MNIST (*а*) и CIFAR10 (*б*)

Fig. 4 | Selection of parameter I for the MNIST (*a*) and CIFAR10 (*b*) data sets

полученные результаты представлены в виде графиков.

По результатам тестирования сделан вывод, что данный параметр не зависит от данных. При его увеличении влияние атаки уменьшается, особенно это видно на результатах первого раунда. Но сильное увеличение данного параметра ведет к увеличению времени работы, затрачиваемого на агрегирование моделей клиентов. Для датасетов MNIST и CIFAR10 наилучшим параметром оказался $i = 9$.

5. РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Проведено тестирование способа защиты. Сравнение велось со способами защиты, на которых основан разрабатываемый способ защиты и со случаем, когда защита не применялась вообще, также не отмечено эталонное значение точности, которое получается без проведения атаки.

Для датасета MNIST проведены тесты с 1–4 атакующими с двумя различными атаками – перевертывание меток и бэкдор-атака. Результаты представлены в виде графиках на рис. 5, на них приведено сравнение точности модели в каждом раунде при атаке перевертывания меток для MNIST для ситуаций, когда защиты нет, нет атаки, применяется предложенный способ защиты и Centered Clipping.

По полученным результатам видно, что предложенный способ защиты эффективен и снижает влияние атаки почти во всех случаях как минимум на 40 %.

Далее на графиках приведено сравнение точности модели в каждом раунде при бэкдор-атаке для MNIST, для ситуаций, когда защиты нет, нет атаки, применяется предложенный способ защиты и Centered Clipping. Результаты приведены на рис. 6.

Проведено сравнение точности бэкдор-атаки для MNIST для ситуаций, когда защиты нет, нет атаки, применяется предложенный способ защиты и Centered Clipping. На рис. 7 показаны графики точности атаки.

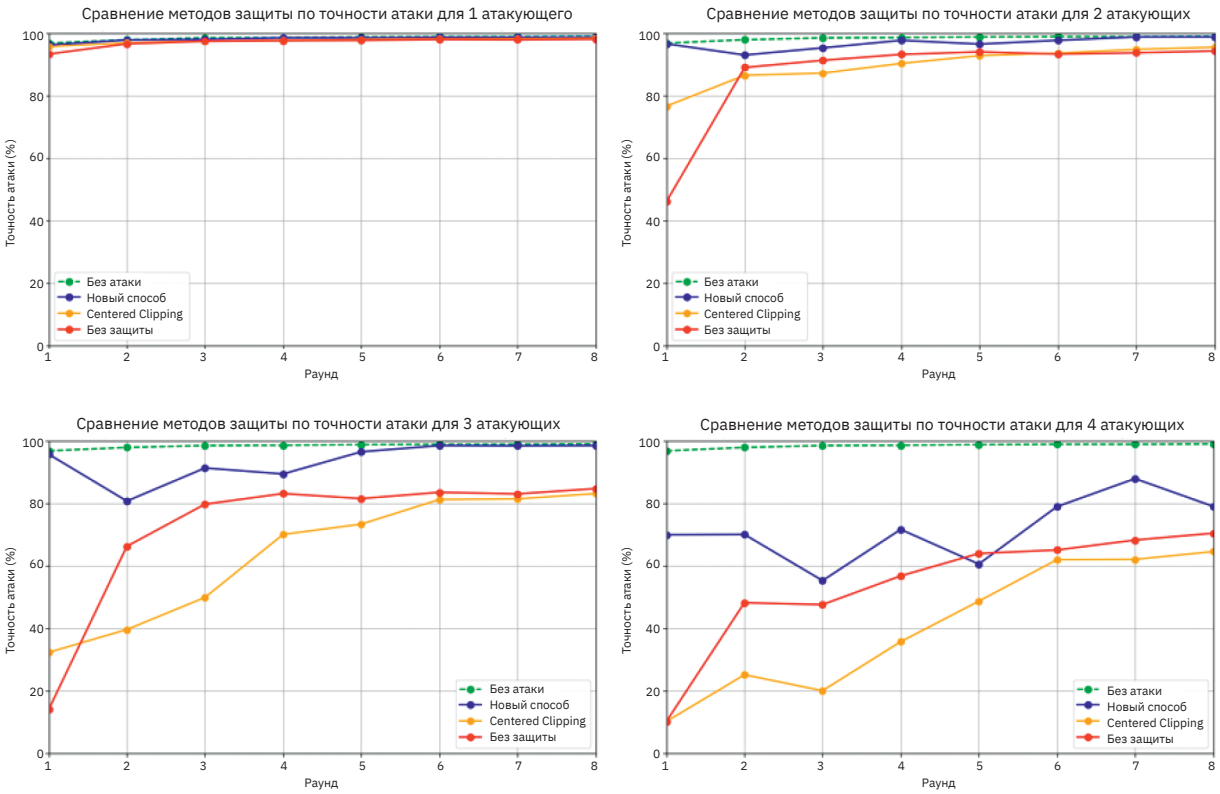


Рис. 5 | Графики точности модели для MNIST при атаке переворота метки

Fig. 5 | Graphs of model accuracy for MNIST under a label flip attack

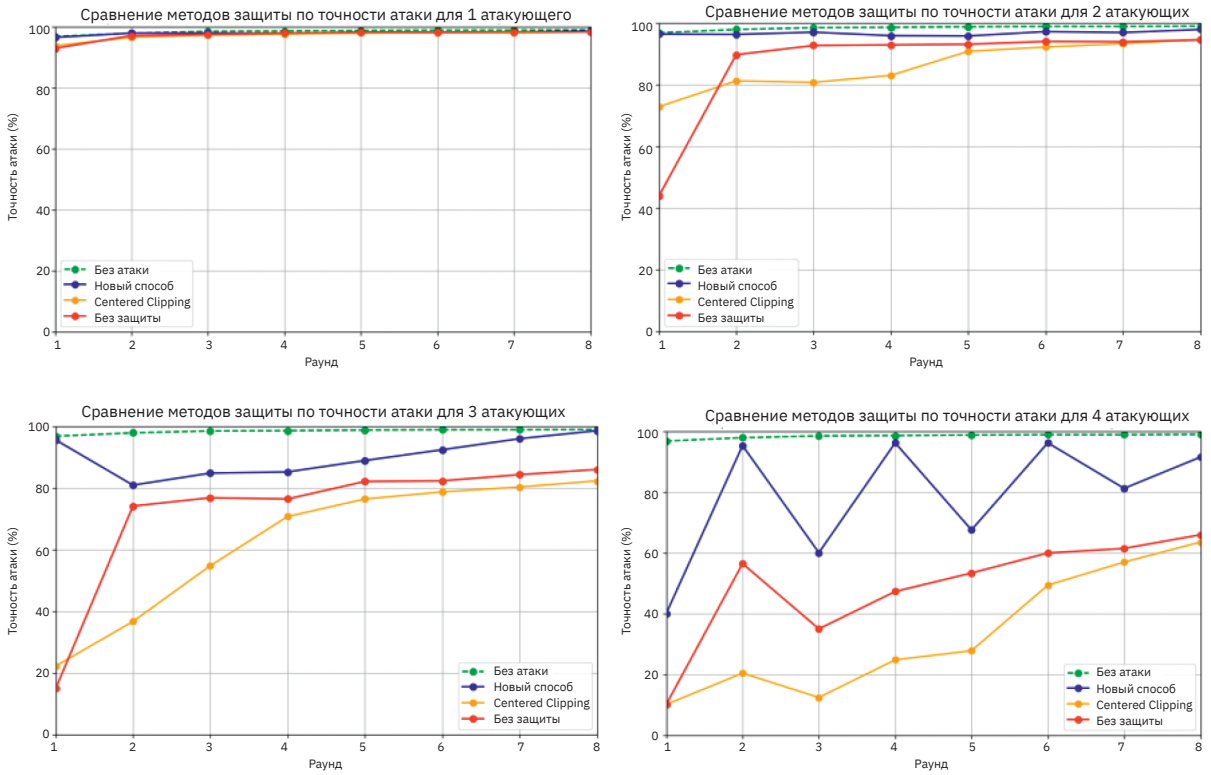


Рис. 6 | Графики точности модели для MNIST при бэкдор-атаке

Fig. 6 | MNIST model accuracy graphs during backdoor attack

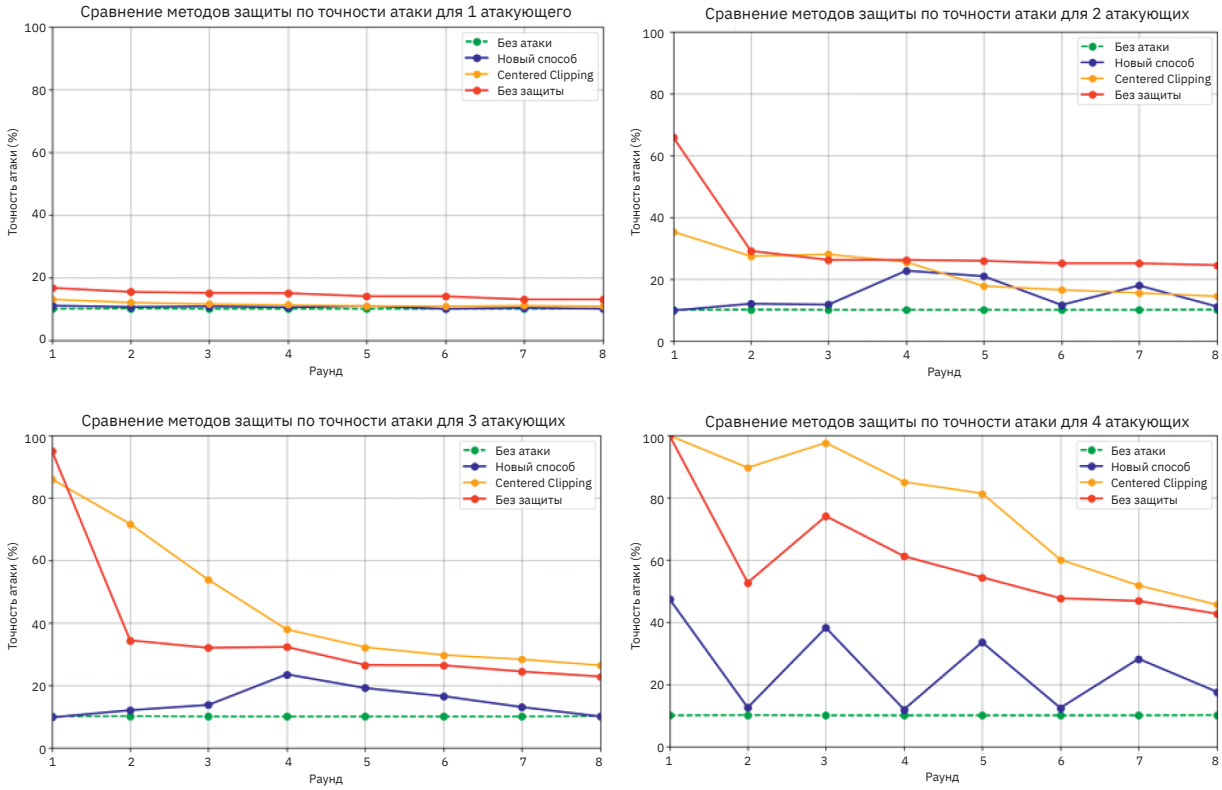


Рис. 7 | Графики точности атаки для MNIST при бэкдор-атаке

Fig. 7 | MNIST attack accuracy graphs for backdoor attacks

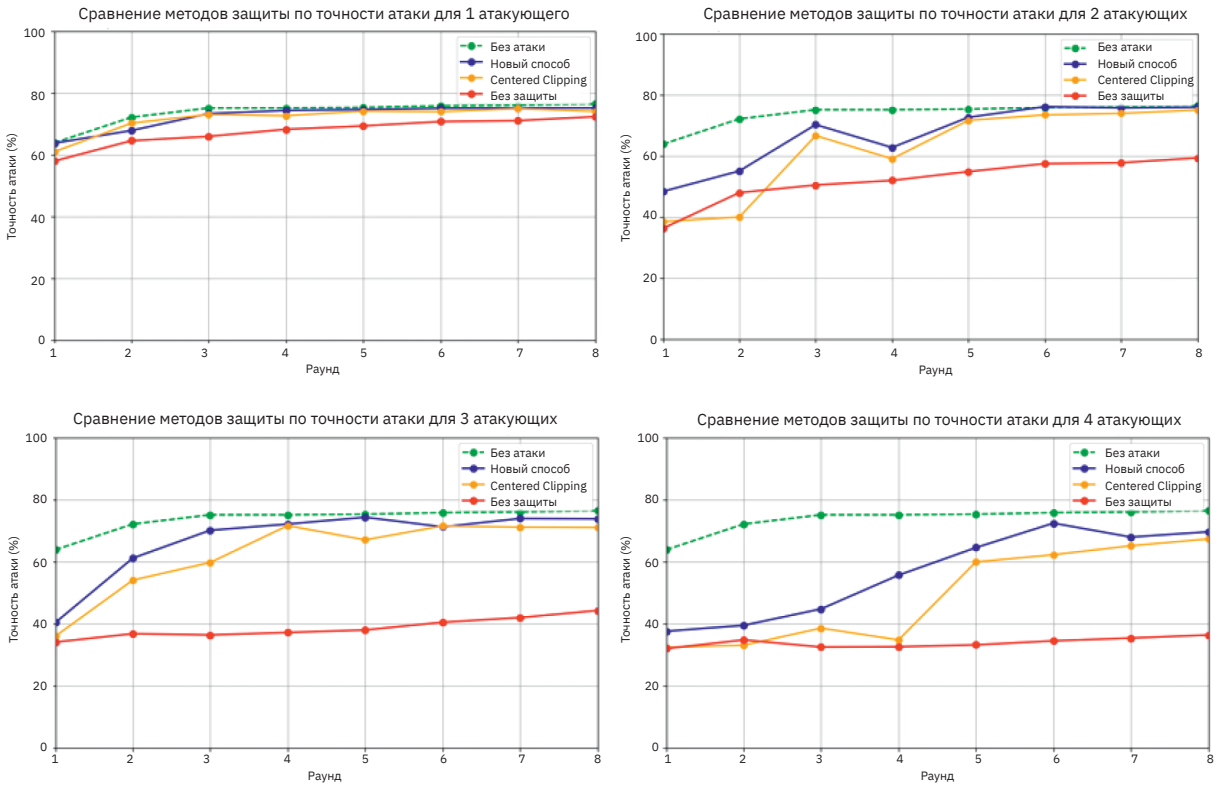


Рис. 8 | Графики точности модели для CIFAR10 при атаке переворота метки

Fig. 8 | Graphs of model accuracy for CIFAR10 during a label flip attack

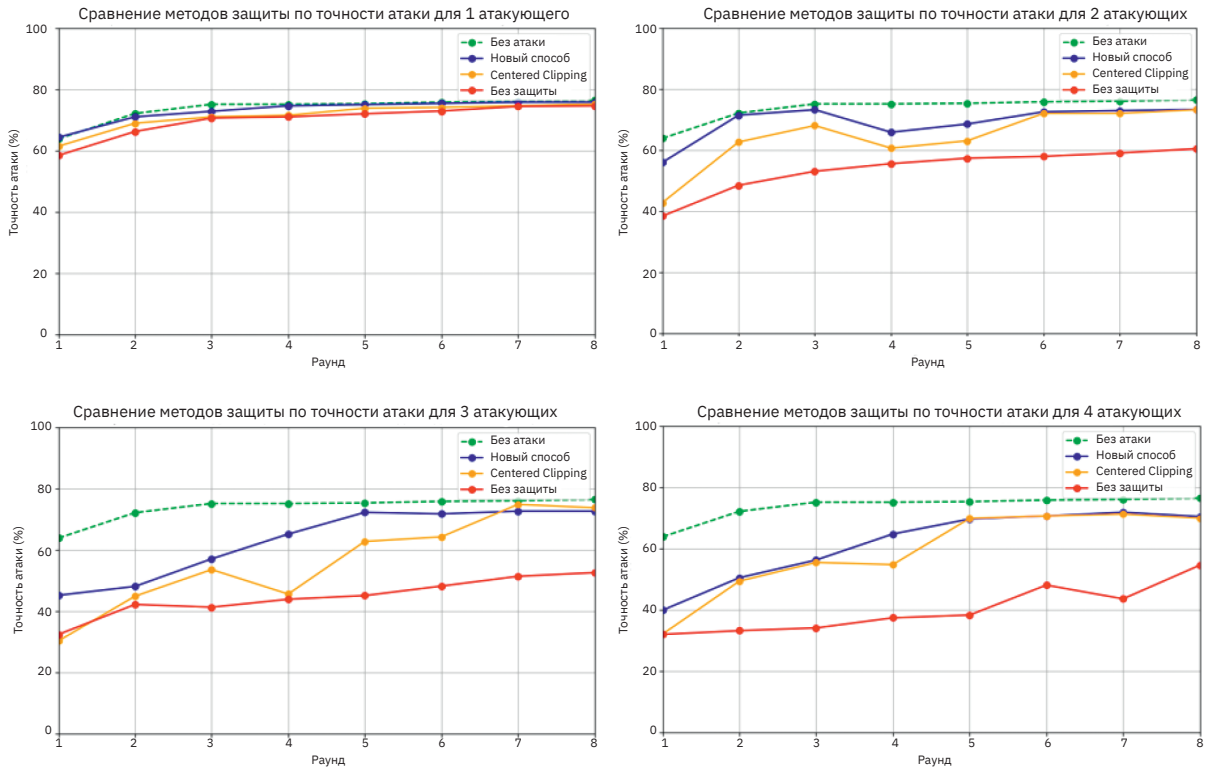


Рис. 9 | Графики точности модели для CIFAR10 при бэкдор-атаке

Fig. 9 | Graphs of model accuracy for CIFAR10 during backdoor attack

По полученным результатам можно сделать вывод, что предложенный метод почти во всех раундах снижает влияние атаки на точность модели как минимум на 30 %, при этом точность атаки не превышает значения 24 % для ситуаций, когда атакующих меньше четырех, и точность атаки не превышает значения 40 % при четырех атакующих, но при этом есть раунды, когда влияние атаки на точность модели не снижается.

Для набора данных CIFAR10 проведены тесты с 1–4 атакующими с двумя различными атаками перевертывание меток и бэкдор-атака. На графиках (рис. 8) приведено сравнение точности модели в каждом раунде при атаке перевертывания меток для CIFAR10 для ситуаций, когда защиты нет, нет атаки, применяется предложенный способ защиты и Centered Clipping.

По полученным результатам можно сделать вывод, что предложенный метод хорошо снижает влияние атаки на точность модели при атаке перевертывания модели. При количестве атакующих менее трех влияние атаки на точность модели снижается как минимум на 30 %, при четырех атакующих же влияние атаки на точность модели снижается более чем на 10 % во всех раундах. При этом при увеличении количества атакующих предложенный способ работает лучше, чем метод надежной агрегации Centered Clipping.

Далее на графиках (рис. 9) показано сравнение точности модели в каждом раунде при бэкдор-атаке для CIFAR10 для ситуаций, когда защиты нет, нет атаки, применяется предложенный способ защиты и Centered Clipping.

Для бэкдор-атаки получены графики для точности атаки. В них сравнивается точность бэкдор-атаки для CIFAR10 для ситуаций, когда защиты нет, нет атаки, применяется предложенный способ защиты и Centered Clipping (рис. 10).

По полученным результатам можно сделать вывод, что влияние атаки на точность модели снижается более чем на 27 %

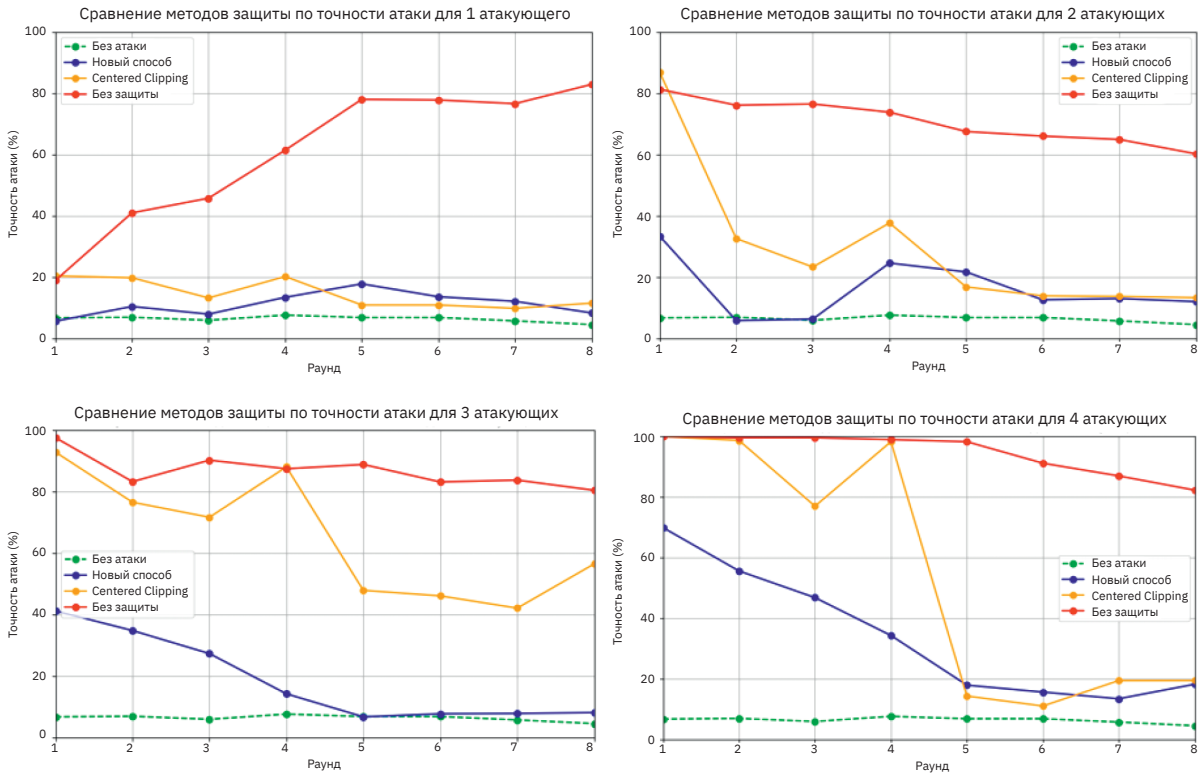


Рис. 10 | Графики точности атаки для CIFAR10 при бэкдор-атаке

Fig. 10 | Attack accuracy graphs for CIFAR10 in a backdoor attack

для ситуации, когда атакующих меньше трех, при четырех атакующих же влияние атаки снижается более чем на 22 % на всех раундах. Точность модели для предложенного метода и для Centered Clipping почти одинаковы, но при этом точность атаки для предложенного метода падает для любого количества атакующих достаточно сильно, как минимум на 20 %, когда при Centered Clipping точность атаки для ситуаций, когда количество атакующих больше одного, доходит до максимальных значений.

6. ЗАКЛЮЧЕНИЕ

При рассмотрении методов защиты от атак отравления изучены методы защиты, применяющиеся при возможности доступа к подозрительным данным и без доступа к ним. Также рассмотрены способы защиты от атаки отравления модели, которая возможна только при использовании предварительно обученных моделях или распределенных типах обучения. По результатам анализа атак и методов защит предложен метод защиты от атаки отравления для систем федеративного обучения, основанный на объединении модификаций известных методов. В предложенном решении использован метод фильтрации FedDefender, измененный для определения более чем одного вредоносного клиента, и метод надежной агрегации Centered Clipping, примененный не к градиентам, а к самим весам моделей.

Для тестирования данного способа разработан тестовый стенд, включающий в себя: систему федеративного обучения с сервером, которому от клиентов приходят только параметры обученных моделей и количество данных, на которых велось обучение; 12 клиентов; две реализованные атаки отравления – бэкдор-атака и атака переворота метки. Предложенное решение протестировано при различном числе атакующих на двух разных атаках. Выполнено сравнение с ситуациями, когда атака на систему не осуществляется, атака проводится, но защита от нее отсутствует или в защите применяется только Centered Clipping. На основе тестов сделан вывод, что предложенный метод эффективен против отравляющих атак. На представленных тестах он выполняет свою задачу не хуже отдельных методов, на которых он основан не смотря на модификации, а в ситуациях, когда атакующих много его эффективность взрастает по сравнению с исходными методами. Недостатком данного решения является рост числа ложных срабатываний при увеличении числа вредоносных клиентов.

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Авторы заявляют об отсутствии конфликта интересов / The authors declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Александрова Е. Б., Гадисова В. А.** Проблемы безопасности федеративных систем, использующих криптографическую защиту // Проблемы информационной безопасности. Компьютерные системы. 2025. № 4. С. 76–88. DOI: 10.48612/jisp/rp53-1tp9-n87g
2. **Безбородов П. Д., Лаврова Д. С.** Защита нейросетевых моделей от угроз нарушения конфиденциальности в федеративном обучении с использованием методов оптимизации // Проблемы информационной безопасности. Компьютерные системы. 2025. № 1. С. 21–29. DOI: 10.48612/jisp/fpvk-xpna-9hx5
3. **Kai Hu, Sheng Gong, Qi Zhang et al.** An overview of implementing security and privacy in federated learning // Artificial intelligence review. 2024. Vol. 57. № 8. P. 204.

4. **Hallaji E., Razavi-Far R., Saif M. et al.** Decentralized federated learning: A survey on security and privacy // IEEE Transactions on Big Data. 2024. Vol. 10. № 2. P. 194–213.
5. **Geming Xia, Jian Chen, Chaodong Yu, Jun Ma.** Poisoning attacks in federated learning: A survey // IEEE Access. 2023. Vol. 11. P. 10708–10722.
6. **Yazdinejad A., Dehghantanha A., Karimipour H. et al.** A robust privacy-preserving federated learning model against model poisoning attacks // IEEE Transactions on Information Forensics and Security. 2024. Vol. 19. P. 6693–6708.
7. **Kasyap H., Tripathy S.** Beyond data poisoning in federated learning // Expert Systems with Applications. 2024. Vol. 235. P. 121192.
8. **Jingwei Sun, Ang Li, DiValentin L. et al.** Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective // Advances in Neural Information Processing Systems. 2021. Vol. 34. P. 12613–12624.
9. **Wei W., Liu L.** Trustworthy distributed AI systems: Robustness, privacy, and governance // ACM Computing Surveys. 2025. Vol. 57. № 6. P. 1–42.
10. **Крундышев В. М., Ческидов В. К., Калинин М. О.** Метод защиты глобальных моделей в системах федеративного обучения на основе модели доверия // Проблемы информационной безопасности. Компьютерные системы. 2024. № 4. С. 94–108. DOI: 10.48612/jisp/mf2n-fb13-p7p6
11. **Shammar E., Cui X., Al-qaness M. A. A.** Swarm learning: a survey of concepts, applications, and trends // arXiv preprint arXiv: 2405.00556. 2024.
12. **Ramirez M. A., Sangyoung Yoon, Damian E. et al.** New data poison attacks on machine learning classifiers for mobile exfiltration // arXiv preprint arXiv:2210.11592. 2022.
13. **Sungwon Park, Sungwon Han, Fangzhao Wu et al.** Feddefender: Client-side attack-tolerant federated learning // Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining. 2023. P. 1850–1861.
14. **Gill W., Anwar A., Gulzar M. A.** Feddefender: Backdoor attack defense in federated learning // Proceedings of the 1st International Workshop on Dependability and Trustworthiness of Safety-Critical Systems with Machine Learned Components. 2023. P. 6–9.
15. **Jian Xu, Shao-Lun Huang, Linqi Song et al.** Byzantine-robust federated learning through collaborative malicious gradient filtering // 2022 IEEE 42nd international conference on distributed computing systems (ICDCS). IEEE, 2022. P. 1223–1235.
16. **Chenghao Yang, Zhengchun Zhou, Yihuai Liang, Chunming Tang.** Sign-Based Privacy-Preserving and Communication-Efficient Federated Learning in Large-Scale Edge Computing // IEEE Transactions on Vehicular Technology. 2026.
17. **Zhang J., Li Q.** Federated Learning Against Dynamic Mixed Poisoning Attack and Defense // International Symposium on Cyberspace Safety and Security. Singapore: Springer Nature Singapore, 2025. P. 316–331.
18. **Xie Y., Fang M., Gong N. Z.** Model poisoning attacks to federated learning via multi-round consistency // Proceedings of the Computer Vision and Pattern Recognition Conference. 2025. P. 15454–15463.
19. **Wang T., Zheng Z., Lin F.** Federated learning framework based on trimmed mean aggregation rules // Expert Systems with Applications. 2025. Vol. 270. P. 126354.
20. **Omer A. R., Khan M. S., Yousafzai A.** Robust federated learning: Defence against model poisoning using mean filtering // 2024 Horizons of Information Technology and Engineering (HITE). IEEE, 2024. P. 1–5.
21. **Wang X., Xia H., Zhang Y.** Defending against model poisoning attacks in federated learning via client-guided trust // 2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2024. P. 1749–1755.
22. **Shejwalkar V., Houmansadr A.** Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning // Network and Distributed System Security Symposium. 2021. DOI: 10.14722/ndss.2021.24498.
23. **Xing S., Ning Z., Zhou J. et al.** N-FedAvg: Novel federated average algorithm based on FedAvg // 2022 14th International Conference on Communication Software and Networks (ICCSN). IEEE, 2022. P. 187–196.
24. **Mehta S., Aneja A.** Securing data privacy in machine learning: The fedavg of federated learning approach // 2024 4th Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2024. P. 1–5.

25. **Yang C., Ghaderi J.** Byzantine-robust decentralized learning via remove-then-clip aggregation // Proceedings of the AAAI Conference on Artificial Intelligence. 2024. Vol. 38. № 19. P. 21735–21743.

26. **Partohaghighi M., Marcia R., West B. J., Chen Y. Q.** When Gradient Clipping Becomes a Control Mechanism for Differential Privacy in Deep Learning // arXiv preprint arXiv: 2602.10584. 2026.

REFERENCES

1. **Aleksandrova E. B., Gadisova V. A.** Security issues in federated learning systems. *Problems of information security. Computer systems*. 2025. No. 4, pp. 76–88. DOI: 10.48612/jisp/rp53-1tp9-n87g.
2. **Bezborodov P. D., Lavrova D. S.** Protecting neural network models from privacy violation threats in federated learning using optimization methods. *Problems of information security. Computer systems*. 2025. No. 1, pp. 21–29. DOI: 10.48612/jisp/fpvk-xpna-9hx5
3. **Kai Hu, Sheng Gong, Qi Zhang et al.** An overview of implementing security and privacy in federated learning. *Artificial intelligence review*. 2024. Vol. 57. No. 8, pp. 204.
4. **Hallaji E., Razavi-Far R., Saif M. et al.** Decentralized federated learning: A survey on security and privacy. *IEEE Transactions on Big Data*. 2024. Vol. 10. No. 2, pp. 194–213.
5. **Geming Xia, Jian Chen, Chaodong Yu, Jun Ma.** Poisoning attacks in federated learning: A survey. *IEEE Access*. 2023. Vol. 11, pp. 10708–10722.
6. **Yazdinejad A., Dehghantanha A., Karimipour H. et al.** A robust privacy-preserving federated learning model against model poisoning attacks. *IEEE Transactions on Information Forensics and Security*. 2024. Vol. 19, pp. 6693–6708.
7. **Kasyap H., Tripathy S.** Beyond data poisoning in federated learning. *Expert Systems with Applications*. 2024. Vol. 235, pp. 121192.
8. **Jingwei Sun, Ang Li, DiValentin L. et al.** Fl-wbc: Enhancing robustness against model poisoning attacks in federated learning from a client perspective. *Advances in Neural Information Processing Systems*. 2021. Vol. 34, pp. 12613–12624.
9. **Wei W., Liu L.** Trustworthy distributed AI systems: Robustness, privacy, and governance. *ACM Computing Surveys*. 2025. Vol. 57. No. 6, pp. 1–42.
10. **Krundyshev V. M., Cheskidov V. K., Kalinin M. O.** A protection method for the global model of the federated learning systems based on a trust model. *Problems of information security. Computer systems*. 2024. No. 4, pp. 94–108. DOI: 10.48612/jisp/mf2n-fb13-p7p6.
11. **Shammar E., Cui X., Al-qaness M. A. A.** Swarm learning: a survey of concepts, applications, and trends. *arXiv preprint arXiv:2405.00556*. 2024.
12. **Ramirez M. A., Sangyoung Yoon, Damian E. et al.** New data poison attacks on machine learning classifiers for mobile exfiltration. *arXiv preprint arXiv:2210.11592*. 2022.
13. **Sungwon Park, Sungwon Han, Fangzhao Wu et al.** Feddefender: Client-side attack-tolerant federated learning. Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining. 2023, pp. 1850–1861.
14. **Gill W., Anwar A., Gulzar M. A.** Feddefender: Backdoor attack defense in federated learning. Proceedings of the 1st International Workshop on Dependability and Trustworthiness of Safety-Critical Systems with Machine Learned Components. 2023, pp. 6–9.
15. **Jian Xu, Shao-Lun Huang, Linqi Song et al.** Byzantine-robust federated learning through collaborative malicious gradient filtering. 2022 IEEE 42nd international conference on distributed computing systems (ICDCS). IEEE, 2022, pp. 1223–1235.
16. **Jian Xu, Shao-Lun Huang, Linqi Song et al.** Sign-Based Privacy-Preserving and Communication-Efficient Federated Learning in Large-Scale Edge Computing. *IEEE Transactions on Vehicular Technology*. 2026.
17. **Zhang J., Li Q.** Federated Learning Against Dynamic Mixed Poisoning Attack and Defense. International Symposium on Cyberspace Safety and Security. Singapore: Springer Nature Singapore, 2025, pp. 316–331.
18. **Xie Y., Fang M., Gong N. Z.** Model poisoning attacks to federated learning via multi-round consistency. Proceedings of the Computer Vision and Pattern Recognition Conference. 2025, pp. 15454–15463.

19. **Wang T., Zheng Z., Lin F.** Federated learning framework based on trimmed mean aggregation rules. *Expert Systems with Applications*. 2025. Vol. 270, pp. 126354.
20. **Omer A. R., Khan M. S., Yousafzai A.** Robust federated learning: Defence against model poisoning using mean filtering. 2024 Horizons of Information Technology and Engineering (HITE). IEEE, 2024, pp. 1–5.
21. **Wang X., Xia H., Zhang Y.** Defending against model poisoning attacks in federated learning via client-guided trust. 2024 IEEE 23rd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE, 2024, pp. 1749–1755.
22. **Shejwalkar V., Houmansadr A.** Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. Network and Distributed System Security Symposium. 2021. DOI: 10.14722/ndss.2021.24498.
23. **Xing S., Ning Z., Zhou J. et al.** N-FedAvg: Novel federated average algorithm based on FedAvg. 2022 14th International Conference on Communication Software and Networks (ICCSN). IEEE, 2022, pp. 187–196.
24. **Mehta S., Aneja A.** Securing data privacy in machine learning: The fedavg of federated learning approach. 2024 4th Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2024, pp. 1–5.
25. **Yang C., Ghaderi J.** Byzantine-robust decentralized learning via remove-then-clip aggregation. Proceedings of the AAAI Conference on Artificial Intelligence. 2024. Vol. 38. No. 19, pp. 21735–21743.
26. **Partohaghighi M., Marcia R., West B. J., Chen Y. Q.** When Gradient Clipping Becomes a Control Mechanism for Differential Privacy in Deep Learning. *arXiv preprint arXiv:2602.10584*. 2026.

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

ПОЛТАВЦЕВА Мария Анатольевна – д-р техн. наук, доцент, профессор, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: poltavtseva@ibks.spbstu.ru
ORCID: 0000-0001-9659-1244

POLTAVTSEVA Maria A. – Doctor of Engineering Sciences, Associate Professor, Professor, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

ВАСИЛЬЕВА Анастасия Александровна – студент, Санкт-Петербургский политехнический университет Петра Великого, Россия, 195251, Санкт-Петербург, ул. Политехническая, д. 29
E-mail: vamp.be.live@gmail.com
ORCID: 0009-0007-3203-6007

VASILYEVA Anastasia A. – Student, Peter the Great St. Petersburg Polytechnic University, Russia, 195251, St. Petersburg, Polytechnicheskaya str., 29

Научная статья
DOI 10.66424/2071-8217-2026-2-11
УДК 004.056.55

О ВЛИЯНИИ ДИСКРЕТИЗАЦИИ НА ПРАКТИЧЕСКУЮ СЕКРЕТНОСТЬ КЛЮЧЕЙ, ФОРМИРУЕМЫХ ПО СХЕМЕ ИНТЕРВАЛОВ

Д. С. Богданов*

Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

✉ *bogdanovds@ramlber.ru

ДЛЯ ЦИТИРОВАНИЯ

Богданов Д. С. О влиянии дискретизации на практическую секретность ключей, формируемых по схеме интервалов // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 138–148.
DOI: 10.66424/2071-8217-2026-2-11

ПОСТУПИЛА 10.02.2026
ПРИНЯТА 07.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Зачастую биты ключа, формируемые физическими генераторами случайных чисел, не являются реализациями независимых в совокупности равномерно распределенных случайных величин, в связи с чем возникает понятие «практическая секретность ключа». Для некоторых физических генераторов случайных чисел указанное отличие обусловлено дискретностью времени, измеряемого электронными компонентами. Для модели физического генератора случайных чисел, построенного по схеме интервалов, получены оценки практической секретности ключей с учетом влияния дискретизации времени измерений.

КЛЮЧЕВЫЕ СЛОВА

Физические генераторы случайных чисел, случайные процессы, теоретико-вероятностная модель, практическая секретность ключей, схема интервалов

Original
DOI 10.66424/2071-8217-2026-2-11

ON THE IMPACT OF DISCRETIZATION ON THE PRACTICAL SECRECY OF KEYS, FORMED BY THE INTERVAL SCHEME

D. S. Bogdanov*

National Research University Higher School of Economics, Moscow, Russia

✉ *bogdanovds@ramlber.ru

FOR CITATION

Bogdanov D. S. On the impact of discretization on the practical secrecy of keys, formed by the interval scheme. *Problems of information security.*

ABSTRACT

Often the key bits generated by physical random number generators are not realizations of jointly independent uniformly distributed random variables, which gives rise to the concept of “practical key secrecy”. For some physical random number generators this deviation is caused by

Computer systems.
2026. No. 2, pp. 138–148.
DOI: 10.66424/2071-8217-2026-2-11
(In Russian)

RECEIVED 10.02.2026
ACCEPTED 07.05.2026
PUBLICATION 15.06.2026

the discreteness of time measured by electronic components. In this paper, for a physical random number generator model based on the interval scheme, we obtain bounds on the practical secrecy of keys taking into account the impact of measurement time discretization.

KEYWORDS

Physical random number generators, random processes, probabilistic model, practical key secrecy, interval scheme

1. ВВЕДЕНИЕ

В практике построения физических генераторов случайных чисел (ФГСЧ) широко применяется подход, основанный на измерении временных интервалов между случайными событиями или от фиксированного начала до момента их наступления. Данная методология, известная как «схема интервалов», рассматривается в обзорных работах [1]. В рамках этой схемы генератор фиксирует моменты наступления некоторых случайных событий (например, приход фотонов на детектор), измеряет интервалы времени между ними с помощью высокочастотного счетчика и формирует выходные биты путем взятия младших разрядов накопленных значений счетчика. Конкретные реализации ФГСЧ, использующие эту схему, можно найти в исследованиях [2–5].

Актуальность разработки надежных ФГСЧ и их строгого анализа подтверждается их критической ролью в современных криптографических системах и постоянным развитием соответствующих стандартов и методов тестирования [6, 7]. Современные исследования в области ФГСЧ направлены как на поиск новых физических принципов [8], так и на углубленный анализ статистических свойств и стойкости существующих конструкций [9, 10].

Как и в случае других физических генераторов, последовательности битов, формируемые по схеме интервалов, часто не обладают свойствами независимости и равновероятности [11]. Это обстоятельство делает необходимым разработку адекватных вероятностных моделей для корректной оценки криптографической стойкости таких ключей. Так, в работе [12] И. М. Арбековым введено понятие практической секретности ключа, характеризующее среднее количество попыток, требуемых для его угадывания. Дальнейшее развитие данной концепции представлено в работах [13, 14]. В частности, в [14] предложена модель, в которой практическая секретность ключа полностью характеризуется параметром ε, определяющим степень отклонения распределения ключей от идеальной равновероятной схемы.

Особенностью ФГСЧ, построенных по схеме интервалов, является то, что отклонение от равновероятности может быть вызвано дискретностью измерения времени электронными компонентами. В работе [4] показано, что конечная частота регистрации сигнала приводит к возникновению зависимостей между соседними значениями выходной последовательности. Схожие эффекты есть и в реализации данной схемы из работы [3].

В настоящей работе с использованием подходов, основанных на исследовании условных распределений и анализе цепей Маркова, аналогичных применяемым в [15], исследуется практическая секретность ключей, формируемых по схеме интервалов. Цель работы – получение оценок сверху отклонения ε для таких ключей с учетом влияния дискретизации времени измерений.

2. МЕТОДЫ ИССЛЕДОВАНИЙ

Приведем основные определения, используемые в работе. В источнике [12] И. М. Арбековым введено понятие практической секретности ключа. Пусть ключ шифра принимает значения в конечном

множестве K с распределением вероятностей $P_K(k)$, $k \in K$. Практической секретностью ключа называется среднее число опробованных ключей до определения истинного ключа шифрования при использовании оптимального усеченного алгоритма перебора [12, 13]. Пусть злоумышленник, обладая некоторой априорной информацией, формирует упорядоченный список ключей в порядке убывания их условных вероятностей и перебирает их последовательно, пока не найдет верный. Если $p_{(1)} \geq p_{(2)} \geq \ldots \geq p_{(|K|)}$ – упорядоченные по убыванию вероятности ключей (при заданной дополнительной информации), то практическая секретность выражается как

$$S = \sum_{i=1}^{|K|} i p_{(i)}.$$

Данный подход формализует классическую идею К. Шеннона о «среднем объеме работы, необходимой для определения ключа», переводя ее на язык строгих вероятностных оценок. В отличие от энтропийных критериев, практическая секретность непосредственно измеряет стойкость ключа к атакам перебором с использованием всей доступной противнику информации о распределении ключей.

В контексте физических генераторов случайных чисел ключ формируется как последовательность знаков (символов), генерируемых устройством. Поэтому распределение ключа P_K полностью определяется совместным распределением выходных знаков ФГСЧ. В работе [14] предложена модель, в которой практическая секретность ключа полностью определяется параметром ε, задающим степень отклонения распределения отдельных знаков и их наборов от идеальной равновероятной схемы. При этом ε-представление позволяет получить достижимые и легко вычислимые оценки практической секретности без непосредственного анализа переборных алгоритмов.

Определение 1. Пусть $\gamma_1, \gamma_2, \ldots$ – случайные величины со значениями в $\{0,1,\ldots,m-1\}$.

Тогда через ε обозначим число, при котором для любого $k \in N$, любых попарно различных $t_1, t_2, \ldots, t_k \in N$ и любых $x_1, x_2, \ldots, x_k \in \{0,1,\ldots,m-1\}$ выполняется

$$\left(\frac{1}{m} - \varepsilon\right)^k \leq P(\gamma_{t_1} = x_1, \ldots, \gamma_{t_k} = x_k) \leq \left(\frac{1}{m} + \varepsilon\right)^k. \quad (1)$$

При $k = 1$ условие (1) означает, что каждое отдельное значение последовательности отклоняется от равномерного распределения не более чем на ε. При $k > 1$ оно ограничивает степень зависимости между элементами последовательности, не позволяя совместному распределению слишком сильно отличаться от произведения маргинальных.

3. ОЦЕНКА ВЛИЯНИЯ ДИСКРЕТИЗАЦИИ НА ПРАКТИЧЕСКУЮ СЕКРЕТНОСТЬ КЛЮЧЕЙ

Определение 2. Пусть $_i$, $i \in N$ – независимые неотрицательные одинаково распределенные случайные величины с плотностью $f(x)$. Через S_n, $n = 0,1,\ldots$, обозначим случайные величины

$$S_0 = 0,\ S_1 = \xi_1, \ldots, S_n = \sum_{i=1}^{n} \xi_i,$$

а через $\tau > 0$ – частоту регистрации сигнала.

Пусть $m \in N$, $m \geq 2$. Через γ_i обозначим случайные величины

$$\gamma_i = \left[\frac{S_i}{\tau}\right] \mod m,$$

где $i \in N$.

Значения γ_i называются «выходной последовательностью», полученной по схеме интервалов. Приведем пример ФГСЧ, построенного по схеме интервалов.

Пример 1: квантовый ФГСЧ на основе подсчета длительности интервалов времени до прилета фотона [3]. Рассмотрим ФГСЧ, работающий по следующему принципу. Источник испускает фотоны в моменты времени, являющиеся случайными величинами. Пусть ξ_n – время между испусканием $(n-1)$-го и n-го фотона. В соответствии с модельным предположением, случайные величины $\{\xi_n\}_{n=1}^{\infty}$

независимы, неотрицательны и одинаково распределены.

В момент регистрации n-го фотона считывается значение счетчика, который инкрементируется с фиксированной частотой $\tau > 0$. Формируемый знак равен младшему биту считанного значения, т.е.

$$\gamma_n = \left[\frac{S_n}{\tau}\right] \bmod 2,$$

где $S_n = \xi_1 + \ldots + \xi_n$ – момент прихода n-го фотона; $[x]$ – целая часть числа x. Таким образом, последовательность $\{\gamma_n\}$ формируется в точном соответствии со схемой интервалов (определение 2) при $m = 2$.

Для оценки сверху ε воспользуемся следующим замечанием.

Утверждение 1. Пусть $\gamma_1, \gamma_2, \ldots$ – случайные величины со значениями в $\{0,1,\ldots,m-1\}$. Условие (1) очевидным образом выполняется, если для любого $k \in N$, для любых попарно различных $t_1, t_2, \ldots, t_k \in N$, для любых $x_1 \in \{0,1,\ldots,m-1\}$,

$$\left| P(\gamma_{t_1} = x_1 \mid \gamma_{t_2} = x_2, \ldots, \gamma_{t_k} = x_k) - \frac{1}{m} \right| \le \varepsilon. \quad (2)$$

При $k = 1$ понимается безусловная вероятность, т.е. $P(\gamma_{t_1} = x_1)$.

Вместо рассмотрения условных вероятностей перейдем к условным математическим ожиданиям и изучим их существенные верхние грани.

Определение 3. Пусть ξ – случайная величина, заданная на вероятностном пространстве (Ω, F, P). Через $\operatorname{ess\,sup}\xi$ обозначим существенную верхнюю грань случайной величины ξ, т.е. такое число, что $P(\omega : \xi > \operatorname{ess\,sup}\xi) = 0$ и $P(\omega : \xi > \operatorname{ess\,sup}\xi - \varepsilon) > 0$ для любого $\varepsilon > 0$. Дальнейшая цель – получить оценки вида:

$$\operatorname{ess\,sup} \left| P(\gamma_{t_1} = x_1 \mid \gamma_{t_2}, \ldots, \gamma_{t_k}) - \frac{1}{m} \right| \le \varepsilon.$$

Тогда для любого набора значений $(x_2, \ldots, x_k)$, для которого множество $\{\omega : \gamma_{t_2}(\omega) = x_2, \ldots, \gamma_{t_k}(\omega) = x_k\}$ имеет положительную вероятность, почти наверное на этом множестве выполняется

$$\left| P(\gamma_{t_1} = x_1 \mid \gamma_{t_2} = x_2, \ldots, \gamma_{t_k} = x_k) - \frac{1}{m} \right| \le \varepsilon,$$

и, следовательно, будут получены оценки практической секретности ключей.

Справедлива следующая лемма.

Лемма 1 [15]. Пусть (Ω, F, P) вероятностное пространство и ξ – случайная величина на нем. Пусть $\mathcal{B}_1$, $\mathcal{B}_2$ – две σ-алгебры, такие что $\mathcal{B}_1 \subseteq \mathcal{B}_2 \subseteq \mathcal{F}$:

$$\operatorname{ess\,sup} E(\xi \mid \mathcal{B}_1) \le \operatorname{ess\,sup} E(\xi \mid \mathcal{B}_2). \quad (3)$$

Из нее можно получить важное следствие: пусть γ_n, $n \in N$ – последовательность, полученная по схеме интервалов. Зафиксируем некоторые $n \in N$ и $r \in \{0,1,\ldots,m-1\}$.

Пусть $\mathcal{B}_1$, $\mathcal{B}_2$ – две σ-алгебры, такие что $\mathcal{B}_1 \subseteq \mathcal{B}_2 \subseteq \mathcal{F}$:

$$\operatorname{ess\,sup} \left| P(\gamma_n = r \mid \mathcal{B}_1) - \frac{1}{m} \right| \le$$

$$\le \operatorname{ess\,sup} \left| P(\gamma_n = r \mid \mathcal{B}_2) - \frac{1}{m} \right|.$$

Доказательство. Обозначим через $\xi = I_{\{\gamma_n = r\}} - \frac{1}{m}$, где $I_{\{\gamma_n = r\}}$ – индикатор события $\{\gamma_n = r\}$. Тогда по определению условного математического ожидания:

$$P(\gamma_n = r \mid \mathcal{B}_1) - \frac{1}{m} = E(\xi \mid \mathcal{B}_1),$$

$$P(\gamma_n = r \mid \mathcal{B}_2) - \frac{1}{m} = E(\xi \mid \mathcal{B}_2).$$

Поскольку $\mathcal{B}_1 \subseteq \mathcal{B}_2$, по телескопическому свойству условных математических ожиданий имеем:

$$E(\xi \mid \mathcal{B}_1) = E(E(\xi \mid \mathcal{B}_2) \mid \mathcal{B}_1).$$

Для любой случайной величины η и σ-алгебры $\mathcal{B}$ выполняется неравенство $|E(\eta \mid \mathcal{B})| \le E(|\eta| \mid \mathcal{B})$ почти наверное (частный случай неравенства Йенсена для условного математического ожидания). Применяя его к $\eta = E(\xi \mid \mathcal{B}_2)$, получаем:

$$|E(\xi \mid \mathcal{B}_1)| = |E(E(\xi \mid \mathcal{B}_2) \mid \mathcal{B}_1)| \le$$

$$\le E(|E(\xi \mid \mathcal{B}_2)| \mid \mathcal{B}_1).$$

Пусть $C = \operatorname{ess\,sup} |E(\xi \mid \mathcal{B}_2)|$. По определению существенной верхней грани $|E(\xi \mid \mathcal{B}_2)| \le C$ почти наверное. Тогда, используя монотонность условного математического ожидания, имеем:

$$E(|E(\xi \mid \mathcal{B}_2)| \mid \mathcal{B}_1) \le E(C \mid \mathcal{B}_1) = C$$

почти наверное. Следовательно, $|E(\xi \mid \mathcal{B}_1)| \le C$ почти наверное.

Таким образом,

$$\operatorname{ess\,sup}\left|E\left(\xi \mid \mathcal{B}_1\right)\right| \le C = \\ = \operatorname{ess\,sup}\left|E\left(\xi \mid \mathcal{B}_2\right)\right|.$$

Возвращаясь к условным вероятностям, получаем требуемое неравенство.

Введем последовательность случайных величин $\theta_1,\theta_2,\ldots$ по следующему правилу:

$$\theta_i = \left\{\frac{S_i}{m\tau}\right\}, \qquad (4)$$

где $\{x\}$ – дробная часть числа x.

Заметим, что

$$\gamma_i = \left[m\theta_i\right], \qquad (5)$$

$i \in N$.

Действительно

$$\frac{S_n}{\tau} = \left[\frac{S_n}{m\tau}\right]m + \left\{\frac{S_n}{m\tau}\right\}m.$$

Выражение $\left[\frac{S_n}{m\tau}\right]m$ является целым числом кратным m, тогда

$$\gamma_i = \left[\frac{S_i}{\tau}\right] \mod m = \left[\left\{\frac{S_n}{m\tau}\right\}m\right] \\ \mod m = \left[m\theta_i\right] \mod m.$$

Но $m\theta_i \in \left[0,m\right)$, значит

$$\left[m\theta_i\right] \mod m = \left[m\theta_i\right].$$

Поскольку значения θ_i однозначно определяют значения γ_i, верно следующее включение:

$$\mathcal{B}_\gamma^{(n)} \subseteq \mathcal{B}_\theta^{(n)},$$

где $\mathcal{B}_\gamma^{(n)}$ – σ-алгебра, порожденная случайными величинами $\gamma_1,\ldots,\gamma_{n-2},\gamma_{n-1},\gamma_{n+1},\gamma_{n+2},\ldots$; $\mathcal{B}_\theta^{(n)}$ – σ-алгебра, порожденная случайными величинами $\theta_1,\theta_2,\ldots,\theta_{n-2},\theta_{n-1},\theta_{n+1},\theta_{n+2},\ldots$. Тогда по следствию 1 получаем, что

$$\sup_n \max_{r=0}^{m-1} \operatorname{ess\,sup}\left|P\left(\gamma_n = r \mid \gamma_1,\ldots,\gamma_{n-2},\gamma_{n-1},\gamma_{n+1},\gamma_{n+2},\ldots\right) - \frac{1}{m}\right| \le \\ \le \sup_n \max_{r=0}^{m-1} \operatorname{ess\,sup}\left|P\left(\left[m\theta_n\right] = r \mid \theta_1,\theta_2,\ldots,\theta_{n-2},\theta_{n-1},\theta_{n+1},\theta_{n+2},\ldots\right) - \frac{1}{m}\right|.$$

Докажем некоторые свойства последовательности θ_n, которые позволят оценить указанные условные вероятности.

Лемма 2. Случайные величины θ_n образуют цепь Маркова со значениями в $\left[0,1\right]$.

Доказательство. Возьмем произвольное измеримое множество $A \in \mathcal{B}\left(\left[0,1\right]\right)$, произвольные $n \in N$, $x_{n-1},x_{n-2},\ldots \in \left[0,1\right]$. Заметим, что

$$\theta_n - \theta_{n-1} = \left\{\frac{S_n}{m\tau}\right\} - \left\{\frac{S_{n-1}}{m\tau}\right\} = \left\{\frac{S_n - S_{n-1}}{m\tau}\right\} = \left\{\frac{\xi_n}{m\tau}\right\},$$

где под $\theta_n - \theta_{n-1}$ понимается вычитание по модулю 1. Тогда все разности $\theta_n - \theta_{n-1}$ независимы между собой, независимы с θ_i, $i < n$, и одинаково распределены. Следовательно

$$P\left(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n-2} = x_{n-2},\ldots\right) = \\ = P\left(\theta_n - \theta_{n-1} \in A - x_{n-1} \mid \theta_{n-1} = x_{n-1}, \theta_{n-2} = x_{n-2},\ldots\right) = \\ = P\left(\theta_n - \theta_{n-1} \in A - x_{n-1} \mid \theta_{n-1} = x_{n-1}\right) = \\ = P\left(\theta_n \in A \mid \theta_{n-1} = x_{n-1}\right).$$

Значит, последовательность θ_n, $n \in N$, действительно образует цепь Маркова и, следовательно, существует функция $p(s,x)$, такая что

$$P\left(\theta_n \in A \mid \theta_{n-1} = s\right) = \int_A p(s,x)dx,$$

для любого $n \in N$, $A \in \mathcal{B}\left(\left[0,1\right]\right)$, $s \in \left[0,1\right]$.

Неформально говоря, случайная величина θ_n характеризует «остаток» $\left\{\frac{S_n}{\tau}\right\}$, который влияет на подсчет длительности следующего интервала. При этом сами остатки образуют цепь Маркова – вероятность следующего интервала (и, следовательно, его остатка) зависит от значения предыдущего остатка и не зависит от более «старых» остатков.

Лемма 2 и формула (5) демонстрируют заявленную зависимость знаков выходной последовательности. Действительно, распределение знака γ_i определяется значением θ_i, которые зависимы между собой.

В силу марковского свойства для любого $n \in N$, любого $r \in \{0,1,\ldots,m-1\}$, любых $x_1, x_2, \ldots, x_{n-1} \in [0,1]$ выполнено

$$P([m\theta_n] = r \mid \theta_1 = x_1, \theta_2 = x_2, \ldots, \theta_{n-1} = x_{n-1}) = \\ = P([m\theta_n] = r \mid \theta_{n-1} = x_{n-1}).$$

Однако нужно искать вероятности при условии фиксации не только конечного «прошлого», но и бесконечного «будущего».

Лемма 3. Пусть $\{\theta_n\}_{n \in N}$ – цепь Маркова со значениями в $[0,1]$ с переходной плотностью $p(s,x)$. Тогда для любого $n \in N$, любого измеримого множества $A \subseteq [0,1]$, любого со события $B \in \mathcal{F}_{N \setminus \{n\}}$ и любых $x_{n-1}, x_{n+1} \in [0,1]$, таких что $P(B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) > 0$, справедливо равенство

$$P(\theta_n \in A \mid B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = P(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}).$$

Рассмотрим сначала случай, когда B – цилиндрическое множество, т.е. представимое в виде

$$B = \{\theta_1 \in B_1\} \cap \{\theta_2 \in B_2\} \cap \ldots \cap \{\theta_{n-1} \in B_{n-1}\} \cap \\ \cap \{\theta_{n+1} \in B_{n+1}\} \cap \ldots \cap \{\theta_{n+t} \in B_{n+t}\}, \quad (6)$$

где $t \in N$ и $B_1, \ldots, B_{n+t} \in \mathcal{B}([0,1])$.

Для такого B имеем:

$$P(\theta_n \in A, B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = \int_{B_1} \cdots \int_{B_{n-2}} \int_{A} \int_{B_{n+1}} \cdots \int_{B_{n+t}} f_{\theta_1}(z_1) \times \\ \times \prod_{i=2}^{n} p(z_{i-1}, z_i) \prod_{i=n}^{n+t} p(z_i, z_{i+1}) dz_{n+t} \cdots dz_{n+1} dz_n dz_{n-2} \cdots dz_1,$$

где $z_{n-1} = x_{n-1}$, $z_{n+1} = x_{n+1}$; f_{θ_1} – плотность распределения θ_1.

Заметим, что подынтегральное выражение факторизуется в произведение трех частей: часть, зависящая только от $z_1, \ldots, z_{n-2}$ и x_{n-1}; зависящая только от x_{n-1}, z_n, x_{n+1}: $p(x_{n-1}, z_n) p(z_n, x_{n+1})$; зависящая только от $x_{n+1}, z_{n+2}, \ldots, z_{n+t}$.

При интегрировании по $z_1, \ldots, z_{n-2}$ и $z_{n+2}, \ldots, z_{n+t}$ получаем множители, не зависящие от z_n. Поэтому

$$P(\theta_n \in A, B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = C_1(x_{n-1}) \int_A p(x_{n-1}, z_n) p(z_n, x_{n+1}) dz_n C_2(x_{n+1}),$$

где $C_1(x_{n-1})$ и $C_2(x_{n+1})$ – некоторые константы (зависящие от B и x_{n-1}, x_{n+1}, но не от A). Аналогично,

$$P(B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = C_1(x_{n-1}) \int_0^1 p(x_{n-1}, z_n) p(z_n, x_{n+1}) dz_n C_2(x_{n+1}).$$

Следовательно,

$$P(\theta_n \in A \mid B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = \frac{\int_A p(x_{n-1}, z_n) p(z_n, x_{n+1}) dz_n}{\int_0^1 p(x_{n-1}, z_n) p(z_n, x_{n+1}) dz_n},$$

что совпадает с $P(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1})$.

Таким образом, утверждение леммы доказано для всех цилиндрических множеств B (образующих π-систему в P).

Теперь докажем, что класс L всех множеств $B \in \mathcal{F}_{N \setminus \{n\}}$, для которых или выполняется утверждение леммы, или B имеет нулевую вероятность $P(B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = 0$, является λ-системой.

Проверим свойства λ-системы:

1. $\Omega \in \mathcal{L}$, так как $P(\Omega, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) > 0$ и утверждение леммы для $B = \Omega$ тривиально выполняется.

2. Пусть $B_1, B_2 \in \mathcal{L}$ и $B_1 \subseteq B_2$. Рассмотрим три случая. Если $P(B_2 \setminus B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = 0$, то $B_2 \setminus B_1 \in \mathcal{L}$ по определению. Если $P(B_2) > 0$, а $P(B_1) = 0$, то очевидно, что $B_2 \setminus B_1 \in \mathcal{L}$. Пусть теперь $P(B_2, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) > 0$ и $P(B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) > 0$. Тогда

$$P(\theta_n \in A, B_2 \setminus B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = P(\theta_n \in A, B_2, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) - \\ - P(\theta_n \in A, B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = P(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) \times \\ \times \left(P(B_2, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) - \right. \\ \left. - P(B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) \right).$$

Аналогично

$$P(B_2 \setminus B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) = \\ = P(B_2, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}) - \\ - P(B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}).$$

Следовательно,

$$P\left(\theta_n \in A \mid B_2 \setminus B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) =$$
$$= \frac{P\left(\theta_n \in A, B_2 \setminus B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right)}{P\left(B_2 \setminus B_1, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right)} =$$
$$= P\left(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right).$$

Значит, $B_2 \setminus B_1 \in \mathcal{L}$.

3. Пусть $B_k \in \mathcal{L}$, $k \geq 1$, и $B_k \uparrow B$. Рассмотрим два случая. Если $P\left(B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) = 0$, то $B \in \mathcal{L}$ по определению.

Если $P\left(B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) > 0$, то найдется k_0 такое, что $P\left(B_{k_0}, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) > 0$. По непрерывности вероятности:

$$P\left(\theta_n \in A \mid B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) =$$
$$= \lim_{k\to\infty} P\left(\theta_n \in A \mid B_k, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) =$$
$$= \lim_{k\to\infty} P\left(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) =$$
$$= P\left(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right).$$

Значит, $B \in \mathcal{L}$.

Таким образом, L – λ-система, содержащая π-систему P цилиндрических множеств. По теореме о $\pi - \lambda$ системах [16, С. 205, теорема 2], $\sigma(\mathcal{P}) \subseteq \mathcal{L}$. Но $\sigma(\mathcal{P}) = \mathcal{F}_{N\setminus\{n\}}$.

Следовательно, для любого $B \in \mathcal{F}_{N\setminus\{n\}}$, если $P\left(B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) > 0$, то

$$P\left(\theta_n \in A \mid B, \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right) =$$
$$= P\left(\theta_n \in A \mid \theta_{n-1} = x_{n-1}, \theta_{n+1} = x_{n+1}\right).$$

Таким образом, оценка практической секретности сводится к оценке выражения

$$\sup_n \max_{r=0}^{m-1} \sup_{x,y\in[0,1]} \left| P\left([m\theta_n] = r \mid \theta_{n-1} = x, \theta_{n+1} = y\right) - \frac{1}{m} \right|.$$

Справедлива следующая лемма.

Лемма 4. Пусть функция $f(x)$ из определения 2 является ограниченной, дифференцируемой на $(0,\infty)$ и $f(x)$ имеет на $(0,\infty)$ ограниченную вариацию. Тогда

$$\sup_{x,y\in[0,1]} \left| p(x,y) - 1 \right| \leq m\tau Var_f,$$

где

$$Var_f = \sup_{0=t_0<t_1<\ldots} \sum_{k\geq 1} \left| f(t_k) - f(t_{k-1}) \right|.$$

Доказательство. Заметим, что из определения случайных величин θ_n следует равенство

$$p(x,y) = p(0, \{y - x\}),$$

где под $\{x\}$ понимается дробная часть числа x. Значит для доказательства леммы достаточно рассмотреть случай $x = 0$.

Заметим, что плотность случайной величины θ_1 равна $p(0,y)$. Выведем формулу

$$p(0,y) = \frac{d}{dy} P\left(\theta_1 \leq y\right) = \frac{d}{dy} P\left(\left\{\frac{\xi_1}{m\tau}\right\} \leq y\right) =$$
$$= \frac{d}{dy} \sum_{k=0}^{\infty} P\left(k < \frac{\xi_1}{m\tau} \leq k + y\right) =$$
$$= \frac{d}{dy}\left[\sum_{k=0}^{\infty} P\left(\frac{\xi_1}{m\tau} \leq k + y\right) - \sum_{k=0}^{\infty} P\left(\frac{\xi_1}{m\tau} < k\right)\right] =$$
$$= m\tau \sum_{k=0}^{\infty} f(m\tau(y+k)).$$

Тогда для любого $z \in [0,1]$

$$\left|p(0,y) - p(0,z)\right| \leq$$
$$\leq m\tau \left| \sum_{k=0}^{\infty} f(m\tau(y+k)) - \sum_{k=0}^{\infty} f(m\tau(z+k)) \right| \leq$$
$$\leq m\tau \sum_{k=0}^{\infty} \left| f(m\tau(y+k)) - f(m\tau(z+k)) \right| \leq m\tau Var_f.$$

Кроме того

$$1 = \int_0^1 p(0,z)dz = \int_0^1 \left(p(0,z) - p(0,y) + p(0,y)\right)dz \leq$$
$$\leq \int_0^1 p(0,y)dz + m\tau Var_f = p(0,y) + m\tau Var_f.$$

Значит, $p(0,y) \geq 1 - m\tau Var_f$. Аналогично доказывается, что $p(0,y) \leq 1 + m\tau Var_f$.

Теорема 1. Пусть в терминах определения 2 функция $f(x)$ является ограниченной, дифференцируемой на $(0,\infty)$, и $f(x)$ имеет на $(0,\infty)$ ограниченную вариацию Var_f, причем $m\tau Var_f < 1$. Тогда

$$\sup_n \max_{r=0}^{m-1} \operatorname{ess\,sup} \left| P\left(\gamma_n = r \mid \gamma_1, \ldots, \gamma_{n-1}, \gamma_{n+1}, \ldots\right) - \right.$$
$$\left. - \frac{1}{m} \right| \leq \frac{4\tau Var_f}{(1 - m\tau Var_f)^2}.$$

Доказательство. Из следствия 1 и леммы 3 следует, что

$$\sup_n \max_{r=0}^{m-1} \operatorname{ess\,sup} \left| P(\gamma_n = r | \gamma_1, \ldots, \gamma_{n-1}, \gamma_{n+1}, \ldots) - \frac{1}{m} \right| \le$$

$$\le \sup_n \max_{r=0}^{m-1} \sup_{x,y \in [0,1]} \left| P([m\theta_n] = r | \theta_{n-1} = x, \theta_{n+1} = y) - \frac{1}{m} \right|.$$

Из доказательства леммы 3 имеем:

$$P\left([m\theta_n] = r \mid \theta_{n-1} = x, \theta_{n+1} = y\right) =$$

$$= \frac{\int_{r/m}^{(r+1)/m} p(x,s)p(s,y)ds}{\int_0^1 p(x,z)p(z,y)dz}.$$

Из леммы 4 следует, что для любых $s, x, y \in [0,1]$ выполняется

$$1 - m\tau Var_f \le p(x,s), p(s,y) \le 1 + m\tau Var_f.$$

Используя эти оценки, получаем верхнюю границу для числителя:

$$\int_{r/m}^{(r+1)/m} p(x,s)p(s,y)ds \le \frac{1}{m}(1 + m\tau Var_f)^2,$$

и нижнюю границу для знаменателя:

$$\int_0^1 p(x,z)p(z,y)dz \ge (1 - m\tau Var_f)^2.$$

Следовательно,

$$P\left([m\theta_n] = r | \theta_{n-1} = x, \theta_{n+1} = y\right) \le \frac{(1 + m\tau Var_f)^2}{m(1 - m\tau Var_f)^2}.$$

Аналогично получаем нижнюю оценку:

$$P\left([m\theta_n] = r | \theta_{n-1} = x, \theta_{n+1} = y\right) \ge \frac{(1 - m\tau Var_f)^2}{m(1 + m\tau Var_f)^2}.$$

Таким образом,

$$\left| P\left([m\theta_n] = r | \theta_{n-1} = x, \theta_{n+1} = y\right) - \frac{1}{m} \right| \le$$

$$\le \max\left\{ \frac{(1 + m\tau Var_f)^2}{m(1 - m\tau Var_f)^2} - \frac{1}{m}, \frac{1}{m} - \frac{(1 - m\tau Var_f)^2}{m(1 + m\tau Var_f)^2} \right\}.$$

Обозначим $x = m\tau Var_f$, где $0 < x < 1$. Рассмотрим два отклонения:

$$A = \frac{(1+x)^2}{m(1-x)^2} - \frac{1}{m}, \quad B = \frac{1}{m} - \frac{(1-x)^2}{m(1+x)^2}.$$

Умножим оба выражения на m:

$$mA = \frac{(1+x)^2}{(1-x)^2} - 1, \quad mB = 1 - \frac{(1-x)^2}{(1+x)^2}.$$

Вычислим

$$mA = \frac{(1+x)^2 - (1-x)^2}{(1-x)^2} = \frac{4x}{(1-x)^2}$$

и

$$mB = \frac{(1+x)^2 - (1-x)^2}{(1+x)^2} = \frac{4x}{(1+x)^2}.$$

Поскольку $(1-x)^2 < (1+x)^2$ при $0 < x < 1$, то $mA > mB$, а значит $A > B$. Следовательно, максимум достигается на первом выражении.

Возвращаясь к A, получаем:

$$A = \frac{1}{m} \cdot \frac{4x}{(1-x)^2} = \frac{4m\tau Var_f}{m(1 - m\tau Var_f)^2} = \frac{4\tau Var_f}{(1 - m\tau Var_f)^2}.$$

Таким образом,

$$\sup_n \max_{r=0}^{m-1} \operatorname{ess\,sup} \left| P(\gamma_n = r | \gamma_1, \ldots, \gamma_{n-1}, \gamma_{n+1}, \ldots) - \right.$$

$$\left. - \frac{1}{m} \right| \le \frac{4\tau Var_f}{(1 - m\tau Var_f)^2}.$$

Пример 2. В соответствии с модельными предположениями работы [17] время до прилета n-го фотона есть случайная величина ξ_n, причем случайные величины ξ_i, $i = 1, 2, \ldots$, являются независимыми, неотрицательными и одинаково распределенными с экспоненциальным распределением $Exp(\lambda)$, где параметр λ определяется из характеристик лазера, испускающего фотоны.

Пусть параметр m из определения 2 равен двум. Для экспоненциального распределения $Exp(\lambda)$ плотность $f(x) = \lambda e^{-\lambda x}$ является убывающей функцией на $(0, \infty)$, поэтому ее полная вариация равна $Var_f = \lambda$. Для обеспечения выполнения условия теоремы 1 $m\tau Var_f = 2\tau\lambda < 1$ необходимо соответствующим образом подбирать параметры τ и λ (см. таблицу). Приведем оценки параметра ε для различных значений λ и τ, рассчитанные по формуле из теоремы 1:

$$\varepsilon \le \frac{4\tau\lambda}{(1 - 2\tau\lambda)^2}.$$

Все представленные комбинации параметров удовлетворяют условию $2\tau\lambda < 1$,

Оценки на ε при различных значениях параметра λ и τ для $m=2$, вычисленные по формуле $\varepsilon \leqslant 4\tau\lambda/(1-2\tau\lambda)^2$

Estimates for ε at different values of the parameter λ and τ for $m = 2$, , calculated by the formula $\varepsilon \leq 4\tau\lambda/(1-2\tau\lambda)^2$

λ\τ	0,05	0,01	0,0065	10^{-4}
0,25	$5{,}27\cdot10^{-2}$	$1{,}01\cdot10^{-2}$	$6{,}54\cdot10^{-3}$	$1{,}00\cdot10^{-4}$
0,5	$1{,}11\cdot10^{-1}$	$2{,}04\cdot10^{-2}$	$1{,}32\cdot10^{-2}$	$2{,}00\cdot10^{-4}$
1	$2{,}47\cdot10^{-1}$	$4{,}17\cdot10^{-2}$	$2{,}69\cdot10^{-2}$	$4{,}00\cdot10^{-4}$

а полученные оценки ε не превышают 0,247. Наибольшее значение $\varepsilon \approx 0{,}247$ соответствует $\lambda = 1$, $\tau = 0{,}05$ $(2\tau\lambda = 0{,}1)$. Как видно из таблицы, уменьшение τ приводит к существенному снижению оценки ε. Например, при $\tau = 10^{-4}$ даже для $\lambda = 1$ получаем $\varepsilon \approx 4 \cdot 10^{-4}$, что свидетельствует о высокой практической секретности ключа.

4. ЗАКЛЮЧЕНИЕ

Предложен подход к оценке практической секретности ключей, формируемых по схеме интервалов с учетом дискретизации времени. Основной результат заключается в сведении исходной криптографической задачи к анализу цепи Маркова, образованной остатками от деления накопленных интервалов времени.

Доказано, что для случайных величин с ограниченной вариацией плотности распределения величина ε может быть сделана сколь угодно малой за счет увеличения частоты регистрации сигнала τ. Представленный пример с экспоненциальным распределением демонстрирует работоспособность подхода и позволяет получить конкретные численные оценки.

Полученные оценки позволяют количественно обосновать требования к параметрам ФГСЧ (частота дискретизации τ и параметры распределения ξ_i) для достижения заданного уровня практической секретности, что является важным шагом на пути к стандартизуемому и верифицируемому проектированию подобных устройств [6, 10].

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Богданов Д. С., Логачев А. С., Миронкин В. О.** Теоретико-вероятностные модели физических генераторов случайных чисел // Проблемы информационной безопасности. Компьютерные системы. 2024. Т. 61. № 3. С. 9–19.
2. **Killmann W., Schindler W.** A Design for a Physical RNG with Robust Entropy Estimators // Cryptographic Hardware and Embedded Systems–CHES 2008: 10th International Workshop, 10–13 August 2008, Washington, DC, USA. 2008. P. 146–163.
3. **Dynes J. F., Yuan Z. L., Sharpe A. W., Shields A. J.** A High Speed, Post-Processing Free, Quantum Random Number Generator // Applied Physics Letters. 2008. Vol. 93. № 3. P. 031109.
4. **Stipcevic M., Rogina B. M.** Quantum random number generator based on photonic emission in semiconductors // Rev. Sci. Instrum. 2007. Vol. 78. P. 1–7.
5. **Nie Y.-Q., Zhang H.-F., Zhang Z. et al.** Practical and fast quantum random number

generation based on photon arrival time relative to external reference // Applied Physics Letters. 2014. Vol. 104. № 5. P. 051110.
6. **Turan M. S., Barker E., Kelsey J. et al.** Recommendation for the Entropy Sources Used for Random Bit Generation // NIST Special Publication (SP) 800-90B. 2018. 84 p.
7. **Killmann W., Schindler W.** A proposal for: Functionality classes and evaluation methodology for true (physical) random number generators // Bundesamt für Sicherheit in der Informationstechnik (BSI). 2002. 38 p.
8. **Marangon D. G., Vallone G., Villoresi P.** Source-Device-Independent Ultrafast Quantum Random Number Generation // Physical Review Letters. 2017. Vol. 118. P. 060503.
9. **Saini A., Tsokanos A., Kirner R.** Quantum randomness in cryptography – a survey of cryptosystems, RNG-based ciphers, and QRNGs // Information. 2022. Vol. 13. № 8. P. 358.
10. **Sunar B., Martin W. J., Stinson D. R.** A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks // IEEE Transactions on Computers. 2019. Vol. 56. № 1. P. 109–119.
11. **Арбеков И. М.** Элементарная квантовая криптография: для криптографов, не знакомых с квантовой механикой. М. : URSS, 2022. 168 с.
12. **Арбеков И. М.** Критерии секретности ключа // Математические вопросы криптографии. 2016. Т. 7. № 1. С. 39–56.
13. **Arbekov I. M.** Lower bounds for the practical secrecy of a key // Mathematical Questions of Cryptography. 2017. Vol. 8. № 2. P. 29–38.
14. **Логачев А. С., Миронкин В. О**. О влиянии вероятностных характеристик дискретных источников, формирующих криптографические ключи, на практическую секретность ключа // Прикладная дискретная математика. 2024. Т. 65. С. 66–83.
15. **Богданов Д. С.** О практической секретности ключей, формируемых из мгновенных значений стационарного гауссовского процесса // Предварительные материалы конференции «CTCrypt 2025». 2025. С. 170–183.
16. **Ширяев А. Н.** Вероятность. В 2-х кн. 7-е изд., стер. М. : МЦНМО, 2021. 416 с.
17. **Furst M., Weier H., Nauerth S. et al.** High speed optical quantum random number generation // Optics Express. 2010. Vol. 18. № 12. P. 13029–13037.

REFERENCES

1. **Bogdanov D. S., Logachev A. S., Mironkin V. O.** The probabilitic-theoretic models of physical random number generators. *Problems of information security. Computer systems*. 2024. No. 3, pp. 9–19. DOI: 10.48612/jisp/m3bn-24ap-6tn8. (In Russian)
2. **Killmann W., Schindler W.** A Design for a Physical RNG with Robust Entropy Estimators. Cryptographic Hardware and Embedded Systems–CHES 2008: 10th International Workshop, 10–13 August 2008, Washington, DC, USA. 2008, pp. 146–163.
3. **Dynes J. F., Yuan Z. L., Sharpe A. W., Shields A. J.** A High Speed, Post-Processing Free, Quantum Random Number Generator. *Applied Physics Letters*. 2008. Vol. 93. No. 3, pp. 031109.
4. **Stipcevic M., Rogina B. M.** Quantum random number generator based on photonic emission in semiconductors. *Rev. Sci. Instrum*. 2007. Vol. 78, pp. 1–7.
5. **Nie Y.-Q., Zhang H.-F., Zhang Z. et al.** Practical and fast quantum random number generation based on photon arrival time relative to external reference. *Applied Physics Letters*. 2014. Vol. 104. No. 5, pp. 051110.
6. **Turan M. S., Barker E., Kelsey J. et al.** Recommendation for the Entropy Sources Used for Random Bit Generation. NIST Special Publication (SP) 800-90B. 2018, 84 p.
7. **Killmann W., Schindler W.** A proposal for: Functionality classes and evaluation methodology for true (physical) random number generators. Bundesamt für Sicherheit in der Informationstechnik (BSI). 2002, 38 p.
8. **Marangon D. G., Vallone G., Villoresi P.** Source-Device-Independent Ultrafast Quantum Random Number Generation. *Physical Review Letters*. 2017. Vol. 118, pp. 060503.
9. **Saini A., Tsokanos A., Kirner R.** Quantum randomness in cryptography – a survey of cryptosystems, RNG-based ciphers, and

QRNGs. *Information*. 2022. Vol. 13. No. 8, pp. 358.
10. **Sunar B., Martin W. J., Stinson D. R.** A Provably Secure True Random Number Generator with Built-In Tolerance to Active Attacks. *IEEE Transactions on Computers*. 2019. Vol. 56. No. 1, pp. 109–119.
11. **Arbekov I. M.** Elementary Quantum Cryptography: for Cryptographers Unfamiliar with Quantum Mechanics. Moscow : URSS, 2022, 168 p. (In Russian)
12. **Arbekov I. M.** Key secrecy criteria. *Mathematical Questions of Cryptography*. 2016. Vol. 7. No. 1, pp. 39–56. (In Russian)
13. **Arbekov I. M.** Lower bounds for the practical secrecy of a key. Mathematical Questions of Cryptography. 2017. Vol. 8. No. 2, pp. 29–38.
14. **Logachev A. S., Mironkin V. O.** On the influence of probabilistic characteristics of discrete sources forming cryptographic keys on the practical secrecy of a key. *Prikladnaya Diskretnaya Matematika (Applied Discrete Mathematics)*. 2024. Vol. 65, pp. 66–83. (In Russian)
15. **Bogdanov D. S.** On the practical secrecy of keys generated from instantaneous values of a stationary Gaussian process. Preliminary materials of the conference “CTCrypt 2025”. 2025, pp. 170–183. (In Russian)
16. **Shiryaev A. N.** Probability. In 2 books. 7th ed., stereotype. Moscow: MCNMO, 2021, 416 p. (In Russian)
17. **Furst M., Weier H., Nauerth S. et al.** High speed optical quantum random number generation. *Optics Express*. 2010. Vol. 18. No. 12, pp. 13029–13037.

СВЕДЕНИЯ ОБ АВТОРЕ / INFORMATION ABOUT AUTHOR

БОГДАНОВ Дмитрий Сергеевич – преподаватель, Национальный исследовательский университет «Высшая школа экономики», Россия, 109028, Москва, Покровский бульвар, д.11
E-mail: bogdanovds@rambler.ru
ORCID: 0000-0001-6178-6420

BOGDANOV Dmitry S. – Lecturer, National Research University Higher School of Economics, Russia, 109028, Moscow, Pokrovksy Bulvar, 11

Научная статья
DOI 10.66424/2071-8217-2026-2-12
УДК 003.26

ИННОВАЦИОННЫЙ МЕТОД ВИЗУАЛЬНОЙ КРИПТОГРАФИИ

И. А. Сикарев[1*], Т. М. Татарникова[2]

[1]Российский государственный гидрометеорологический университет, Санкт-Петербург, Россия
[2]Санкт-Петербургский государственный университет аэрокосмического приборостроения, Санкт-Петербург, Россия

✉ *sikarev@yandex.ru

ДЛЯ ЦИТИРОВАНИЯ
Сикарев И. А., Татарникова Т. М. Инновационный метод визуальной криптографии // Проблемы информационной безопасности. Компьютерные системы. 2026. № 2. С. 149–157.
DOI: 10.66424/2071-8217-2026-2-12

ПОСТУПИЛА 17.02.2026
ПРИНЯТА 06.05.2026
ОПУБЛИКОВАНА 15.06.2026

Издатель: Санкт-Петербургский политехнический университет Петра Великого

АННОТАЦИЯ

Предложен инновационный метод визуальной криптографии с использованием маршрутной перестановки при небольшом размере графического файла и регулярной смене ключей шифрования. Предлагается принять один пиксель изображения в качестве одного элемента шифрования, отказаться от последовательных перемещений и использовать двумерное пространство размерностью $n \times n$ для перемещения пикселей. Предлагаются сценарии применения предложенного инновационного алгоритма визуальной криптографии, такие как хранение биометрических данных в защищенном виде, совместное использование секретов и предварительная обработка изображения для блочного шифрования. Выполнена оценка предложенного метода визуальной криптографии.

КЛЮЧЕВЫЕ СЛОВА

Визуальная криптография, маршрутные перестановки, изображение, метод, программное приложение

Original article
DOI 10.66424/2071-8217-2026-2-12

VISUAL CRYPTOGRAPHY INNOVATIONS METHOD

I. A. Sikarev[1*], T. M. Tatarnikova[2]

[1]Russian State Hydrometeorological University, St. Petersburg, Russia
[2]State University of Aerospace Instrumentation, St. Petersburg, Russia

✉ *sikarev@yandex.ru

FOR CITATION
Sikarev I. A., Tatarnikova T. M. Visual cryptography innovations method. *Problems of information security. Computer systems*. 2026. No. 2, pp. 149–157.
DOI: 10.66424/2071-8217-2026-2-12
(In Russian)

ABSTRACT

Proposed innovative method of visual cryptography using route permutation for small image file size and regular change of encryption keys. It is proposed to take one pixel of the image as one encryption element, to abandon sequential movements and to use a two-dimensional space of dimension $n \times n$ for moving pixels. Application scenarios of the proposed innovative visual cryptography algorithm are proposed, such as storing biometric data in a secure form, sharing

RECEIVED 17.02.2026
ACCEPTED 06.05.2026
PUBLICATION 15.06.2026

secrets, and preprocessing images for block encryption. The proposed visual cryptography algorithm is evaluated.

KEYWORDS

Visual cryptography, route permutations, image, method, software application

1. ВВЕДЕНИЕ

Традиционными задачами обеспечения информационной безопасности являются криптографическая защита, разделение прав доступа, целостность, подлинность электронных документов и др. [1–3]. Для решения указанных задач обеспечения безопасности может использоваться визуальная криптография [4].

Визуальная криптография является методом шифрования зрительной информации – картинки или текста таким образом, что дешифрование становится механической операцией, не требующей использования компьютера.

Самый известный метод визуальной криптографии – это графическая схема с разделением секрета, разработанная М. Наором и А. Шамиром в 1994 г., согласно которой изображение разделено на n частей так, что только имеющий все n частей мог расшифровать изображение, в то время как остальные (n–1) части не несут никакой информации об оригинальном изображении. Каждая часть напечатана на отдельном диапозитиве. Расшифровка осуществлялась путем наложения всех частей, в результате чего появлялось исходное изображение. Переложение этого алгоритма в компьютерную систему предполагает наложение частей изображения друг на друга с помощью логических операций конъюнкции, дизъюнкции, исключающего или.

Если изображение представить квадратной матрицей, размер которой соответствует количеству пикселей по горизонтали и вертикали, то использование маршрутных перестановок также может стать основой шифрования изображения. Каждая промежуточная перестановка может считаться отдельным диапозитивом. Отметим, что для текстовых сообщений подобные алгоритмы уже существуют. Алгоритмы визуальной криптографии [5–8] обладают определенными недостатками и могут быть улучшены способом за счет применения идеи шифров маршрутной перестановки [9].

Идея перестановочных шифров в целом адекватна идее визуальной криптографии, поскольку процесс шифрования больше механический, чем вычислительный. Каждая перестановка может считаться одним слоем; за один элемент шифрования может быть принят один пиксель изображения.

Целью исследования является разработка инновационного метода визуальной криптографии за счет изменений схем применения маршрутной перестановки.

2. МЕТОДЫ И МАТЕРИАЛЫ

В ходе исследования рассмотрены математические аспекты переупорядочивания набора целых чисел. Также использовались методические основы следующих алгоритмов визуальной криптографии: простой алгоритм визуальной криптографии для бинарных (черно-белых) изображений [5]; визуальная схема (k, N) [6]; частный случай (2, N) визуальной схемы шифрования (k, N) [7]; алгоритм шифрования цветного изображения [8].

3. РЕЗУЛЬТАТЫ ИССЛЕДОВАНИЙ

В ходе исследований установлено, что основной недостаток маршрутных перестановок, например, решетки Кардано состоит в том, что шифр формируется

определенным образом – слева направо, сверху вниз, что делает его легко раскрываемым. Решением данной проблемы является усиление алгоритма шифрования следующими требованиями: отказаться от последовательного вписывания символов в «окна» решетки; отказаться от последовательного поворачивания решетки по или против часовой стрелки; использовать алгоритм в двумерном пространстве размерностью $n \times n$ для перемещения пикселей; искусственно усложнить алгоритм в рамках языка программирования.

Формирование ключа в предлагаемом алгоритме состоит из следующих шагов (на примере формирования ключа 4×4):

• выбираем классическую матрицу (решетку)

*	*	*	X
*	*	*	*
X	*	X	*
*	X	*	*

• выбираем P-Box (перестановку):

*	*	*	3
*	*	*	*
1	*	4	*
*	2	*	*

• выбираем комбинацию поворотов, например, [2031].

Получаем конечный массив перестановок:

$$[2, 5, 16, 10, 15, 12, 1, 7, 9, 14, 4, 11, 8, 3, 13, 6]_{4\times4=16.}$$

На плоскости маршрут выглядит так, как на рис. 1. В линейном виде маршрут перестановок приведен на рис. 2.

Общее количество ключей рассчитывается как произведение числа матриц на число поворотов:

$$\text{Количество ключей } = 4^{N^2/4} \cdot 24 .$$

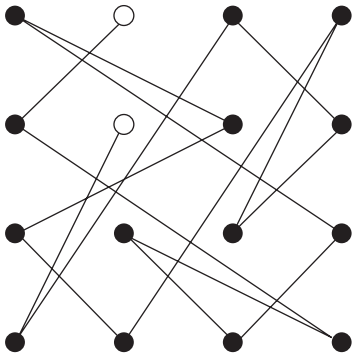


Рис. 1 | Маршрут перестановок на плоскости

Fig. 1 | The route of permutations on the plane

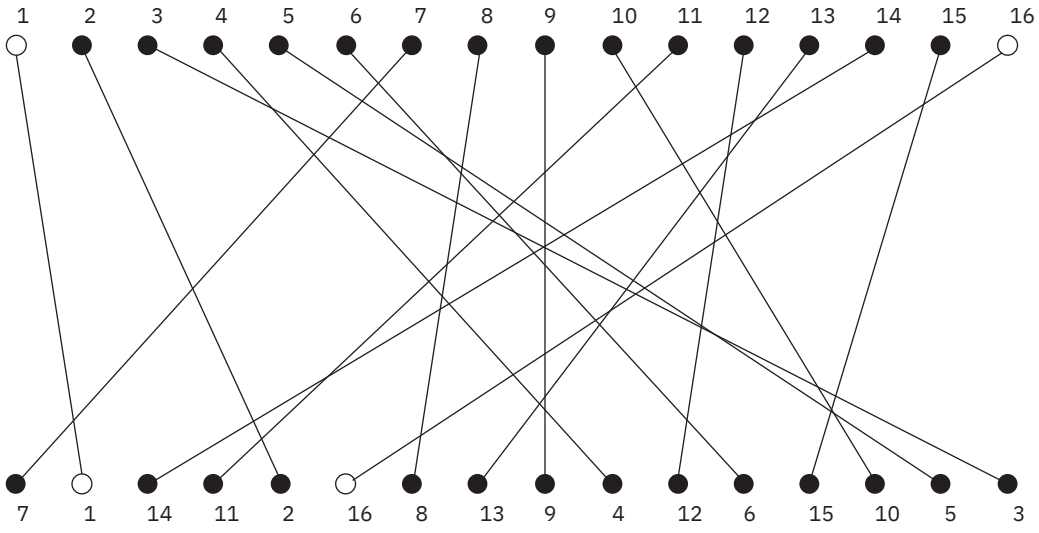


Рис. 2 | Линейный маршрут перестановок

Fig. 2 | Linear route of permutations

Для предлагаемого метода общее число матриц маршрутных перестановок равно произведению числа матриц на число поворотов и число возможных последовательностей:

$$\text{Количество ключей} = 4^{N^2/4} \cdot 24 \cdot \left[\left(\frac{N^2}{4}!\right)\right]^4.$$

На основании данных формул построим сводную таблицу с данными о количестве ключей для классического и предложенного методов в соответствии с размерностью матрицы маршрутных перестановок $n \times n$ и при $N > 16$ количество ключей в предлагаемом методе становится практически бесконечным.

На рис. 3. приведен график зависимости количества ключей от размерности матрицы 14×14, из которого можно проследить зависимость между возрастанием количества ключей с применением классического и предлагаемого метода.

Количество ключей
Quantity of keys

N	Количество ячеек	Количество ключей	
		Классический метод	Предлагаемый метод
2	4	96	96
4	256	6144	2038431744
6	262144	6291456	1,09095*E*+29
8	4294967296	1,03079*E*+11	1,97537*E*+64
10	1,13*E*+15	2,70216*E*+16	1,5642*E*+117
12	4,72*E*+21	1,13337*E*+23	2,1703*E*+189
14	3,17*E*+29	7,6059*E*+30	1,0413*E*+282
16	3,40*E*+38	8,16678*E*+39	∞

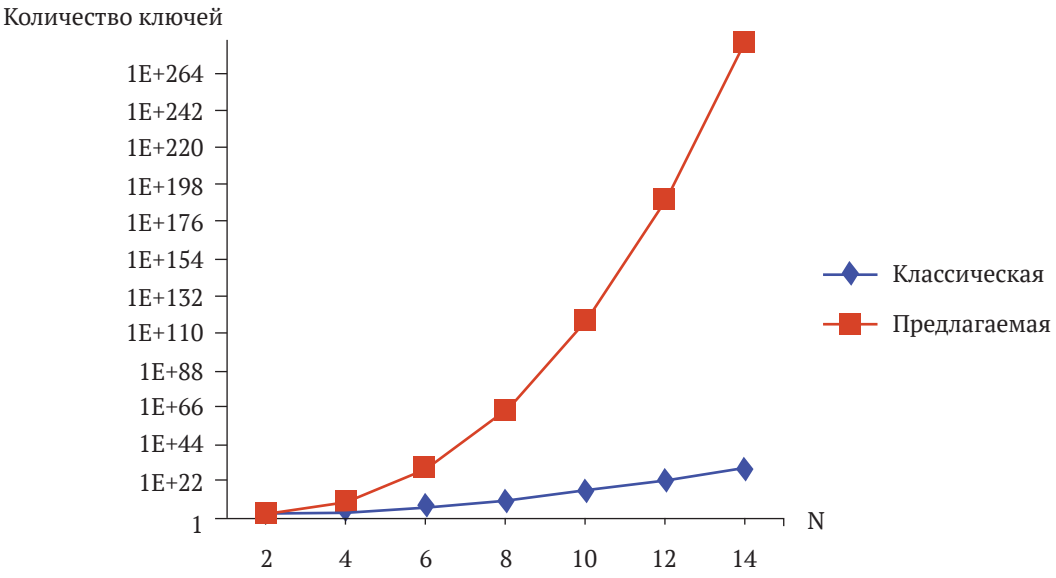


Рис. 3 | График зависимости количества ключей от способа шифрования

Fig. 3 | Graph of the dependence of the number of keys on the encryption method

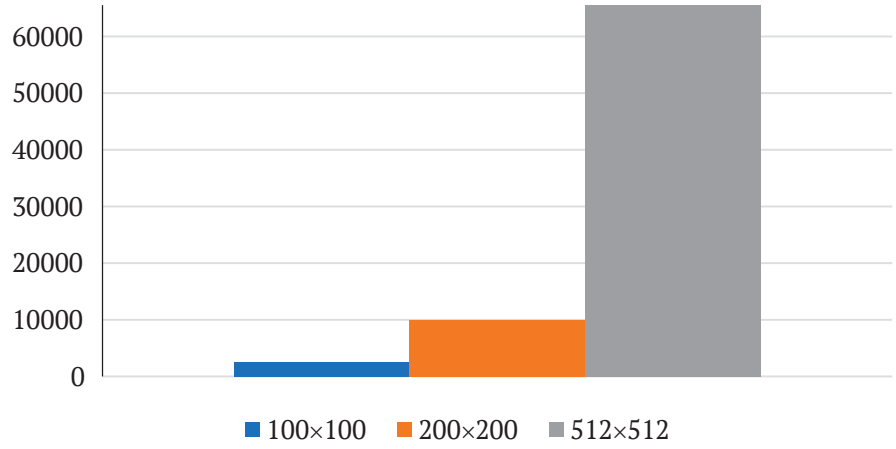


Рис. 4 | Сложность генерации маршрутных перестановок для шифрования изображений разного размера

Fig. 4 | The complexity of generating route permutations for encrypting images of different sizes

В соответствии с предлагаемым методом шифрования для визуальной криптографии оценена сложность генерации маршрутных перестановок для шифрования изображений разного размера. На рис. 4 приведены результаты этих сравнений.

4. ОБСУЖДЕНИЕ

Предлагаемый метод визуальной криптографии может найти несколько вариантов применения.

Первый сценарий – это хранение в защищенном виде биометрических данных. Разработанное приложение визуальной криптографии можно использовать для защищенного хранения визуальных файлов малого размера. Это применимо к биометрическим системам, основанным на работе с отпечатками пальцев, такие системы хранят изображения малого размера и предложенный алгоритм визуальной криптографии способен сокрыть информацию с этих изображений [10]. Также можно передавать эти зашифрованные изображения и расшифровывать их на другой стороне, для чего нужно передать файл с ключом по защищенному каналу. Злоумышленник не сможет определить, какой алгоритм шифрования использован пока не получит доступ к коду программы и не узнает способ генерации и поворота матриц.

Вторая схема применения предлагаемого алгоритма визуальной криптографии – разделение секрета путем создания коалиции участников из некоторой первоначальной группы участников с установлением утвержденного лимита числа участников коалиции. Каждый участник имеет свою сгенерированную матрицу: у одного участника решетка без поворота, у второго с поворотом на 90°, у третьего – 180°, у четвертого – 270°. После наложения всех решеток друг на друга и применения этой результирующей решетки на зашифрованное изображение получится расшифровать это изображение. При отсутствии хотя бы одной решетки расшифровать изображение не удастся, т.е. злоумышленнику недостаточно перехватить одну или несколько из решеток, ему нужно перехватить все решетки, зашифрованное изображение из открытого канала и знать какой алгоритм применялся для сокрытия секрета.

Третий вариант применения – предварительная подготовка изображения для шифрования блочными алгоритмами типа AES, GOST и др. Как правило во всех этих алгоритмах перед непосредственно шифрованием происходит выравнивание

статистических свойств блока, т.е. равновероятное распределение битов нулей и единиц. Для такого случая также можно использовать предлагаемый алгоритм.

Еще один вариант применения алгоритма – это симметричная схема обмена визуальной информацией.

В ходе исследования по предложенному методу визуальной криптографии разработано и реализовано приложение, которое работает с изображениями квадратной формы формата bmp. В приложении, согласно алгоритму, изображение разбивается на блоки и строятся матрицы перестановки. Выходом приложения является зашифрованное изображение и сообщение об их несоответствии с эталонным. В качестве справочной информации выступает файл с расширением .txt, хранящий ключи шифрования. На рис. 5 приведены результаты шифрования (справа) некоторых контрольных изображений (слева).

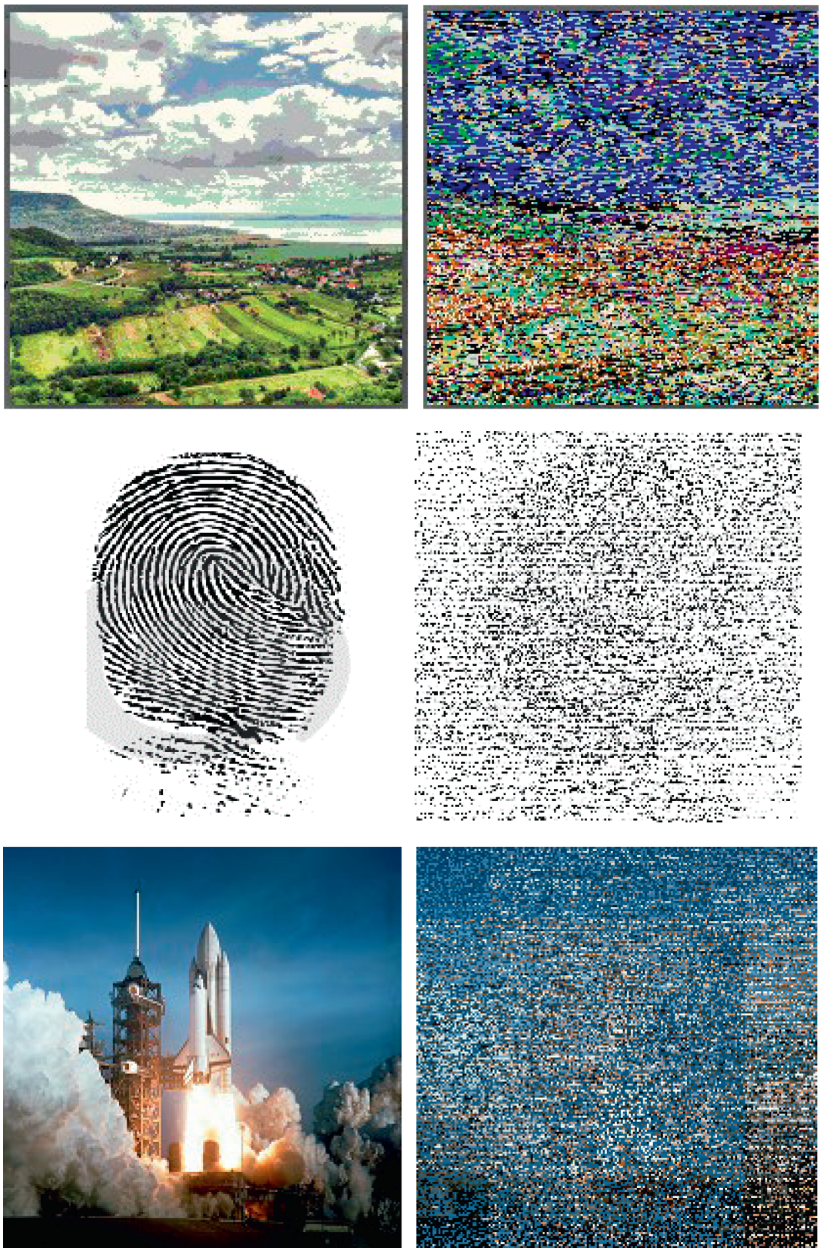


Рис. 5 | Примеры шифрования изображений небольших объемов

Fig. 5 | Examples of small-volume image encryption

5. ЗАКЛЮЧЕНИЕ

Предложено применить маршрутные перестановки для визуальной криптографии. Указано, что идея перестановочных шифров хорошо отображает идеи визуальной криптографии, заключающийся в том, что процесс шифрования сводится не к вычислениям, а перестановкам и за один элемент шифрования может быть принят один пиксель изображения.

Предложено усовершенствование классического шифра маршрутных перестановок применительно к визуальной криптографии за счет отказа от последовательных перестановок и использовании *n*-мерного пространства перестановок.

Предложенный метод может найти применение при шифровании графических файлов небольшого объема и при регулярной смене ключей шифрования/дешифрования. Выполнена оценка предложенного метода визуальной криптографии, которая показала экспоненциальный рост количества ключей в сравнении с классическим методом.

Показано, что предложенный метод может быть применен при хранении в защищенном виде биометрических данных. Следует указать, что он также может быть использован в геоинформационных системах [11–18], автоматизированных транспортных системах [19–21] и в области высшего образования [22].

КОНФЛИКТ ИНТЕРЕСОВ / CONFLICT OF INTERESTS

Автор заявляет об отсутствии конфликта интересов / The author declare no conflict of interests.

СПИСОК ИСТОЧНИКОВ

1. **Tatarnikova T. M., Sikarev I. A., Bogdanov P. Yu., Timochkina T. V.** Botnet Attack Detection Approach in Out Networks // Automatic Control and Computer Sciences. 2022. Vol. 56. № 8. P. 838–846. DOI: 10.3103/s0146411622080259. EDN: VILOAN.
2. **Сикарев И. А., Абрамов В. М., Простакевич К. С. и др.** Инфокоммуникационный инструментарий для управления природными рисками при мореплавании автономных судов в Арктике при изменении климата // Проблемы информационной безопасности. Компьютерные системы. 2024. № 1(58). С. 110–120. DOI: 10.48612/jisp/v28t-z3kr-nrn2. EDN: RUESZV.
3. **Бескид П. П., Татарникова Т. М.** О некоторых подходах к решению проблемы авторского права в сети интернет // Ученые записки Российского государственного гидрометеорологического университета. 2010. № 15. С. 199–210.
4. **Ibrahim D. R., Teh J. S., Abdullah R.** An overview of visual cryptography techniques // Multimed Tools Appl. 2021. № 80. P. 31927–31952. DOI: 10.1007/s11042-021-11229-9.
5. **Prasad S., Pal A. K.** An RGB color image steganography scheme using overlapping block-based pix-el-value differencing // Royal Society Open Science. 2017. Vol. 4. № 161066. DOI: 10.1098/ rsos.161066.
6. **Косолапов Ю. В.** О построении (k,n)-схемы визуальной криптографии с применением класса линейных хэш-функций над бинарным полем // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. 2018. Т. 18. Вып. 2. С. 227–239. DOI: 10.18500/1816-9791-2018-18-2-227-239. EDN: XQFNSP.
7. **Lakshmanan R., Arumugam S.** Construction of a (k, n)-visual cryptography scheme // Des. Codes Cryptogr. 2017. Vol. 82. Iss. 3. P. 629–645. DOI: 10.1007/s10623-016-0181-z.
8. **Савельева М. Г., Урбанович П. П.** Метод стеганографического преобразования web-документов на основе растровой графики и модели RGB // Труды БГТУ. Сер. 3:

Физико-математические науки и информатика. 2022. № 2 (260). С. 99–107.
9. **Татарникова Т. М.** Анализ данных в прикладных задачах обеспечения информационной безопасности. СПб. : ГУАП, 2018. 115 с.
10. **Ефремов Д. А., Борисова С. Н.** Использование отпечатков пальцев в задачах биометрического ограничения доступа // Успехи современного естествознания. 2011. № 7. С. 107–108.
11. **Sikarev I. A., Chistyakov G. B., Garanin A. V., Moskvin D. A.** Algorithms for Enhancing Information Security in the Processing of Navigation Data of Unmanned Vessels of the Technical Fleet of the Inland Waterways of the Russian Federation // Automatic Control and Computer Sciences. 2020. Vol. 54. № 8. P. 964–967. DOI: 10.3103/S0146411620080325. EDN: AKAYKV.
12. **Lukyanov S., Popov N., Sikarev I. et al.** Digital learning technologies within geoinformation management // E3S Web of Conferences, 17–19 February 2021, Chelyabinsk, Russia. 2021. P. 01004. DOI: 10.1051/e3sconf/202125801004. EDN: GWVYAN.

REFERENCES

1. **Tatarnikova T. M., Sikarev I. A., Bogdanov P. Yu., Timochkina T. V.** Botnet Attack Detection Approach in Out Networks. *Automatic Control and Computer Sciences*. 2022. Vol. 56. No. 8, pp. 838–846. DOI: 10.3103/s0146411622080259. EDN: VILOAN.
2. **Sikarev I. A., Abramov V. M., Prostakevich K. S. et al.** Infocommunication instrumentarium for natural risk management while navigation of autonomous vessels in Arctic under climate change. *Problems of information security. Computer systems*. 2024. No. 1, pp. 110–120. DOI: 10.48612/jisp/v28t-z3kr-nrn2. (In Russian)
3. **Beskid P. P., Tatarnikova T. M.** About some approaches to the copyright solution of a problem in the internet. *Proceedings of the Russian State Hydrometeorological University*. 2010. No. 15, pp. 199–210. (In Russian)
4. **Ibrahim D. R., Teh J. S., Abdullah R.** An overview of visual cryptography techniques. *Multimed Tools Appl*. 2021. No. 80, pp. 31927–31952. DOI: 10.1007/s11042-021-11229-9.
5. **Prasad S., Pal A. K.** An RGB color image steganography scheme using overlapping block-based pix-el-value differencing. *Royal Society Open Science*. 2017. Vol. 4. No. 161066. DOI: 10.1098/ rsos.161066.
6. **Kosolapov Yu. V.** On the Construction of (k, n)-Schemes of Visual Cryptography Using a Class of Linear Hash Functions Over a Binary Field. *Izv. Saratov Univ. (N. S.), Ser. Math. Mech. Inform.* 2018. Vol. 18. Iss. 2, pp. 227–239. DOI: 10.18500/1816-9791-2018-18-2-227-239. EDN: XQFNSP. (In Russian)
7. **Lakshmanan R., Arumugam S.** Construction of a (k, n)-visual cryptography scheme. *Des. Codes Cryptogr*. 2017. Vol. 82. Iss. 3, pp. 629–645. DOI: 10.1007/s10623-016-0181-z.
8. **Saveleva M. G., Urbanovich P. P.** Method of steganographic transformation of web-documents based on raster graphics and RGB model. *Proceedings of BSTU, issue 3, Physics and Mathematics. Informatics*. 2022. No. 2 (260), pp. 99–107. (In Russian)
9. **Tatarnikova T. M.** Data analysis in applied information security tasks. St. Petersburg : GUAP, 2018, 115 p.
10. **Efremov D. A., Borisova S. N.** The use of fingerprints in biometric access control tasks. *Advances in current natural sciences*. 2011. No. 7, pp. 107–108.
11. **Sikarev I. A., Chistyakov G. B., Garanin A. V., Moskvin D. A.** Algorithms for Enhancing Information Security in the Processing of Navigation Data of Unmanned Vessels of the Technical Fleet of the Inland Waterways of the Russian Federation. *Automatic Control and Computer Sciences*. 2020. Vol. 54. No. 8, pp. 964–967. DOI: 10.3103/S0146411620080325. EDN: AKAYKV.
12. **Lukyanov S., Popov N., Sikarev I. et al.** Digital learning technologies within geoinformation management. E3S Web of Conferences, 17–19 February 2021, Chelyabinsk, Russia. 2021, pp. 01004. DOI: 10.1051/e3sconf/202125801004. EDN: GWVYAN.

СВЕДЕНИЯ ОБ АВТОРАХ / INFORMATION ABOUT AUTHORS

СИКАРЕВ Игорь Александрович – д-р техн. наук, профессор, Российский государственный гидрометеорологический университет, Россия, 192007, Санкт-Петербург, Воронежская ул., д. 79
Email: sikarev@yandex.ru
ORCID: 0000-0001-6289-3295

SIKAREV Igor A. – Doctor of Engineering Sciences, Professor, Russian State Hydrometeorological University, Russia, 192007, St. Petersburg, Voronezhskaya str., 79

ТАТАРНИКОВА Татьяна Михайловна – профессор, Санкт-Петербургский государственный университет аэрокосмического приборостроения, Россия, 190000, Санкт-Петербург, Большая Морская ул., д. 67, лит. А
Email: Tm-tatarn@yandex.ru
ORCID: 0000-0002-6419-0072

TATARNIKOVA Tatiana M. – Doctor of Engineering Sciences, Professor, State University of Aerospace Instrumentation, Russia, 190000, St. Petersburg, Bolshaya Morskaya str., 67, lit. A

ПРОБЛЕМЫ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ
КОМПЬЮТЕРНЫЕ СИСТЕМЫ

№ 2, 2026

Редактор *В. Е. Филиппова*
Корректор *В. Е. Филиппова*
Компьютерная верстка *А. А. Новиковой*
Дизайн обложки *И. А. Теплякова*

Подписано в печать 15.06.2026. Формат 60×84/8.
Усл. печ. л. 19,75. Тираж 250. Заказ 3286.

Отпечатано в Издательско-полиграфическом центре
Политехнического университета.
195251, Санкт-Петербург, Политехническая ул., 29.
Тел.: (812) 552-77-17; 550-40-14.