Russian Technological Journal.
2025;13(5):25–40

ISSN 2782-3210 (Print)
ISSN 2500-316X (Online)



*УДК 004.056.5*
*https://doi.org/10.32362/2500-316X-2025-13-5-25-40*
*EDN JKQMQM*



НАУЧНАЯ СТАТЬЯ

# Имитационная модель масштабируемого метода выявления многовекторных атак с учетом ограничений вычислительных и информационных ресурсов IoT-устройств

**В.И. Петренко, Ф.Б. Тебуева, М.Г. Огур @, Г.И. Линец, В.П. Мочалов**

*Северо-Кавказский федеральный университет, Ставрополь, 355017 Россия*
*@ Автор для переписки, e-mail: ogur26@gmail.com*

• **Поступила:** 14.10.2024 • **Доработана:** 13.05.2025 • **Принята к опубликованию:** 07.08.2025

**Резюме**

**Цели.** Основная цель работы – разработка масштабируемого метода для выявления многовекторных атак на устройства интернета вещей (Internet of Things, IoT). Учитывая рост угроз безопасности в IoT-сетях, решение должно обеспечивать высокую точность обнаружения атак при минимальных вычислительных затратах и с учетом ограничений ресурсов IoT-устройств.

**Методы.** Для достижения поставленной цели разработана гибридная архитектура нейронных сетей, сочетающая сверточные сети для анализа пространственных зависимостей и сети долгой краткосрочной памяти или Gated Recurrent Units (управляемые рекуррентные блоки) – один из видов рекуррентных нейронных сетей для анализа временных зависимостей в сетевом трафике. Техника обрезки (pruning) сокращает параметры модели и вычислительные затраты. Блокчейн с механизмом консенсуса Proof of Voting[1] обеспечивает безопасное управление данными и децентрализованную верификацию.

**Результаты.** Эксперименты на датасете CIC IoT Dataset 2023[2] показали эффективность модели: точность и F1-мера составили 99.1%, что подтверждает способность выявлять известные и новые атаки в реальном времени с высокой точностью и полнотой. Время обработки сокращено до 12 мс, использование памяти – до 180 МБ, что делает модель пригодной для устройств с ограниченными ресурсами.

**Выводы.** Разработанная модель превосходит аналоги по точности, времени обработки и использованию памяти. Гибридная архитектура, обрезка и децентрализованная верификация обеспечивают эффективность против многовекторных угроз IoT. Работа открывает перспективы для исследований в кибербезопасности, предлагая решения для защиты IoT-сетей от сложных атак.

[1] Proof of Voting (алгоритм консенсуса) – это консенсусный алгоритм в блокчейн-сетях, при котором участники подтверждают транзакции и обеспечивают безопасность сети путем голосования за блоки или транзакции. [Proof of Voting is a consensus algorithm in blockchain networks, in which participants confirm transactions and ensure network security by voting for blocks or transactions.]

[2] CIC IoT Dataset 2023. http://cicresearch.ca/IOTDataset/CIC_IOT_Dataset2023/Dataset/. Дата обращения 30.06.2025. / Accessed June 30, 2025.

**Ключевые слова:** многовекторные атаки, интернет вещей, выявление угроз, нейронные сети, блокчейн, обрезка нейронов, кибербезопасность, компрометация узлов, консенсус, федеративное обучение

**Для цитирования:** Петренко В.И., Тебуева Ф.Б., Огур М.Г., Линец Г.И., Мочалов В.П. Имитационная модель масштабируемого метода выявления многовекторных атак с учетом ограничений вычислительных и информационных ресурсов IoT-устройств. *Russian Technological Journal*. 2025;13(5):25–40. https://doi.org/10.32362/2500-316X-2025-13-5-25-40, https://www.elibrary.ru/JKQMQM

**Прозрачность финансовой деятельности:** Авторы не имеют финансовой заинтересованности в представленных материалах или методах.

Авторы заявляют об отсутствии конфликта интересов.

RESEARCH ARTICLE

# Simulation model of a scalable method for detecting multi-vector attacks taking into account the limitations of computing and information resources of IoT devices

**Vyacheslav I. Petrenko, Fariza B. Tebueva, Maxim G. Ogur [@], Gennady I. Linets, Valery P. Mochalov**

*North Caucasus Federal University, Stavropol, 355017 Russia*
*[@] Corresponding author, e-mail: ogur26@gmail.com*

• **Submitted:** 14.10.2024 • **Revised:** 13.05.2025 • **Accepted:** 07.08.2025

**Abstract**

**Objectives.** The study sets out to develop a scalable method for detecting multi-vector attacks on Internet of Things (IoT) devices. Given the growth of security threats in IoT networks, such a solution must provide high accuracy in detecting attacks with minimal computing costs while taking into account the resource constraints of IoT devices.

**Methods.** The developed hybrid neural network architecture combines convolutional networks for spatial dependence analysis and long short-term memory networks or gated recurrent units representing types of recurrent neural networks for analyzing time dependencies in network traffic. Model parameters and computational costs are reduced by pruning. A blockchain with a proof of voting[3] consensus mechanism provides secure data management and decentralized verification.

**Results.** Experiments on the CIC IoT Dataset 2023[4] showed the effectiveness of the model: the accuracy and F1 measure were 99.1%. This confirms the ability to detect known and new attacks in real time with high accuracy and completeness. Processing time is reduced to 12 ms, while memory usage is reduced to 180 MB, which makes the model suitable for devices with limited resources.

**Conclusions.** The developed model is superior to analogues in terms of accuracy, processing time, and memory usage. Hybrid architecture, pruning, and decentralized verification provide effectiveness against multi-vector IoT threats.

[3] Proof of Voting is a consensus algorithm in blockchain networks, in which participants confirm transactions and ensure network security by voting for blocks or transactions.

[4] CIC IoT Dataset 2023. http://cicresearch.ca/IOTDataset/CIC_IOT_Dataset2023/Dataset/. Accessed June 30, 2025.

**Keywords:** multi-vector attacks, Internet of Things, threat detection, neural networks, blockchain, neuronal pruning, cybersecurity, node compromise, consensus, federated learning

**For citation:** Petrenko V.I., Tebueva F.B., Ogur M.G., Linets G.I., Mochalov V.P. Simulation model of a scalable method for detecting multi-vector attacks taking into account the limitations of computing and information resources of IoT devices. *Russian Technological Journal.* 2025;13(5):25–40. https://doi.org/10.32362/2500-316X-2025-13-5-25-40, https://www.elibrary.ru/JKQMQM

**Financial disclosure:** The authors have no financial or proprietary interest in any material or method mentioned.

The authors declare no conflicts of interest.

## ВВЕДЕНИЕ

С развитием технологий интернета вещей (Internet of Things, IoT) сети IoT-устройств становятся неотъемлемой частью современной информационной инфраструктуры. Эти устройства обеспечивают взаимодействие множества систем и платформ в реальном времени, что повышает эффективность, удобство и гибкость различных отраслей – от умных домов и городов до промышленных и медицинских систем. Однако широкое распространение IoT приводит к увеличению числа потенциальных угроз информационной безопасности, т.к. IoT-устройства часто ограничены в вычислительных и энергетических ресурсах, что делает их уязвимыми для многовекторных кибератак. Среди наиболее опасных атак можно выделить DDoS[5], атаки на маршрутизацию, SQL[6]-инъекции и другие формы многовекторных угроз.

Современные методы выявления атак часто требуют значительных вычислительных ресурсов и недостаточно эффективны в условиях ограниченных возможностей IoT-устройств. Это приводит к необходимости разработки новых подходов, которые бы учитывали ограничения вычислительных и информационных ресурсов IoT и одновременно обеспечивали высокий уровень безопасности.

В данной статье предложена масштабируемая модель выявления многовекторных атак, которая сочетает гибридную архитектуру нейронных сетей CNN + LSTM/GRU[7] (сверточная нейронная сеть Convolutional Neural Network (CNN) и сеть долгой краткосрочной памяти Long Short-Term Memory (LSTM)) для анализа пространственно-временных зависимостей сетевого трафика, децентрализованную верификацию данных с использованием блокчейн-технологий и технику обрезки нейронов (pruning) для снижения вычислительных затрат. Предложенная модель ориентирована на работу в реальном времени и на ограниченных ресурсах, что делает ее применимой для современных IoT-сетей. В статье проводится экспериментальная оценка эффективности разработанной модели на основе датасета CIC IoT Dataset 2023, что позволяет продемонстрировать ее преимущество по сравнению с существующими решениями.

## 1. АНАЛИЗ ЛИТЕРАТУРЫ

В [1] предложен математический аппарат для моделирования кибератак на энергосети с использованием подходов теории игр и построения графов атак. В основе модели лежит динамическое взаимодействие между атакующим и защитником (attacker-defender dynamics), где атакующий пытается нарушить работу энергосети, а защитник стремится предотвратить атаки, используя предиктивные и реактивные меры защиты. Основным достоинством модели является учет в ней динамического взаимодействия между атакующим и защитником, что делает модель более реалистичной для применения в сложных системах. Модель использует графы атак, что позволяет моделировать многослойные и многошаговые атаки, учитывая их сложность и разнообразие. К недостаткам модели можно отнести требование наличия исходных данных о системе и уязвимостях, а также точных оценок вероятностей успешной атаки и затрат на вывод системы из строя. Это может усложнять ее применение на практике для систем с ограниченными данными.

В [2] предложен метод обнаружения многовекторных кибератак в инфраструктуре IoT на основе анализа сетевого трафика и машинного обучения. Авторы выделяют 4 ключевых типа признаков для анализа трафика: основанные на потоках данных, MQTT[8], DNS[9] и HTTP[10], которые помогают

[5] Distributed Denial of Service (распределенный отказ от обслуживания) – форма кибератаки на веб-системы с целью вывести их из строя или затруднить доступ к ним для обычных пользователей. [Distributed Denial of Service is a form of cyberattack on web systems in order to disable them or make it difficult for ordinary users to access them.]

[6] Structured Query Language – язык структурированных запросов.

[7] Gated Recurrent Unit – управляемый рекуррентный блок.

[8] Message queuing telemetry transport – протокол обмена данными для IoT.

[9] Domain name system – система доменных имен.

[10] HyperText transfer protocol – протокол для передачи данных.

в ускоренном выявлении атак. Метод позволяет повысить эффективность обнаружения атак за счет ранней диагностики вредоносного трафика на основе анализа потоков и глубокого анализа пакетов для точного выявления многовекторных атак. Это делает метод подходящим для сетей IoT с высоким объемом данных и сложной структурой атак. Однако сложность метода заключается в необходимости точного определения набора признаков и их обработки, что требует больших вычислительных ресурсов для реального времени в крупных IoT-сетях.

В [3] предложена методика защиты децентрализованных IoT-сетей от многовекторных DDoS-атак на основе применения блокчейн-технологий и методов глубокого обучения. Предлагается двухэтапный подход «Prevent-then-Detect», где на 1-м этапе система предотвращения атак (intrusion prevention system, IPS) работает через блокчейн-консорциум валидаторов, а на 2-м этапе система обнаружения атак (intrusion detection system, IDS) использует модели глубокого обучения для анализа сетевого трафика и выявления угроз. Блокчейн обеспечивает безопасность передачи данных между узлами сети и управляет доступом к ресурсам IoT-сети с использованием интеллектуальных контрактов, которые фиксируют действия по обнаружению атак и предотвращению угроз. Система предотвращения атак в блокчейн-консорциуме использует алгоритм консенсуса для проверки подозрительного трафика. Однако модель требует значительных вычислительных ресурсов для работы блокчейн-системы и глубоких нейронных сетей, что может ограничить ее применение на устройствах с ограниченными вычислительными возможностями в IoT-сетях.

В [4] предложен метод для обнаружения многовекторных атак (multi-vector attacks, MVA) на основе использования многослойной перцептронной сети (multilayer perceptron, MLP). Авторы используют подход, основанный на анализе сетевого трафика с применением машинного обучения для выявления различных векторов атак. В частности, сетевые данные из пакетных захватов (packet capturing, PCAP) анализируются для определения аномальных паттернов в поведении сетевых соединений. Цель метода – повысить точность обнаружения атак за счет использования нейронных сетей для классификации данных и их последующего анализа. Основной компонент предложенной системы – MLP, состоящая из входного слоя, скрытого слоя и выходного слоя. Метод MLP подходит для задач, где важна высокая точность обнаружения, однако для сетей с ограниченными вычислительными ресурсами могут потребоваться более быстрые модели.

В [5] предложен анализ различных методов выявления DDoS-атак в IoT-сетях с акцентом на особенности и вызовы, возникающие при применении этих методов к сетям Интернета вещей. Авторы проводят обзор нескольких категорий методов обнаружения атак, включая сигнатурные, аномалийные и гибридные подходы. Математический аппарат включает алгоритмы, основанные на машинном обучении, такие как метод опорных векторов (Support Vector Machine, SVM), деревья решений (Decision Trees), метод K-ближайших соседей (K-Nearest Neighbors, KNN) и метод случайного леса (Random Forest). Эти алгоритмы используются для классификации сетевого трафика и выделения аномальных паттернов, характерных для DDoS-атак. Предложенные подходы позволяют более эффективно справляться с высокообъемными и разнообразными атаками на IoT-устройства, однако их эффективность ограничена сложностью настройки моделей и необходимостью больших вычислительных ресурсов.

В [6] предложен математический аппарат для оценки методов обнаружения вторжений в условиях многовекторных атак 5-го поколения (Gen V Multi-Vector Attacks). Основу методики составляет комбинация 2 методов принятия решений: нечеткой аналитической иерархии процессов (Fuzzy Analytic Hierarchy Process, Fuzzy AHP) и техники выбора на основе сходства с идеальным решением (Technique for Order Preference by Similarity to Ideal Solution, TOPSIS). Эти методы позволяют оценить различные критерии эффективности систем обнаружения атак: точность обнаружения, адаптивность, масштабируемость, влияние на ресурсы, время отклика и автоматизация. Особое внимание уделяется таким аспектам, как адаптация к новым угрозам, возможность работы в масштабируемых сетях и минимизация нагрузки на ресурсы при обеспечении высокого уровня автоматизации и быстрого реагирования.

В [7] предложен подход для обнаружения многовекторных DDoS-атак в сетях интернета вещей с использованием глубокого ансамблевого обучения и метода обрезки нейронных сетей. Авторы представляют систему Deep Ensemble learning with Pruning (DEEPShield), которая объединяет сети CNN и LSTM для анализа сетевого трафика и обнаружения как высокообъемных, так и малообъемных DDoS-атак. Основой математического аппарата является использование ансамблевого подхода, где CNN отвечает за извлечение пространственных признаков из сетевого трафика, а LSTM используется для анализа временных зависимостей. Система DEEPShield демонстрирует высокую точность в обнаружении атак, превышающую 90%, и сокращает время на предсказание по сравнению с аналогичными моделями.

В [8] предложена гибридная модель для анализа угроз и классификации атак в сетях Интернета вещей с использованием глубинного обучения и адаптивного алгоритма оптимизации Mayfly (LAMOA[11]). Модель направлена на выявление маршрутизирующих атак в IoT-сетях, таких как атаки типа sinkhole, wormhole, black hole и Sybil, которые значительно снижают производительность сетей и их безопасность. Основой модели является использование рекуррентной нейронной сети с долгой краткосрочной памятью для обработки временных рядов сетевого трафика и классификации атак, дополненной адаптивным алгоритмом Mayfly для оптимизации гиперпараметров модели. Модель демонстрирует высокую способность к точной классификации различных типов атак и является эффективным решением для обеспечения безопасности IoT-сетей, однако ее сложность и вычислительные затраты могут ограничивать ее применение в сетях с ограниченными ресурсами, что требует дальнейшей оптимизации.

В [9] предлагается легковесная структура для обнаружения многовекторных DDoS-атак в системах мобильной медицины на базе IoT с использованием глубокого обучения. Подход акцентирует внимание на важности точности и эффективности, что перекликается с целями предлагаемой модели. подчеркивает необходимость адаптации методов обнаружения к специфике мобильных IoT-устройств, что делает их работу актуальной для дальнейших исследований в данной области.

В [10] предложена методология для обнаружения и противодействия многовекторным угрозам в децентрализованных IoT-системах. Авторы акцентируют внимание на необходимости комплексных стратегий безопасности. Это подчеркивает важность интеграции различных методов защиты, включая машинное обучение и блокчейн-технологии.

Статьи [11–15] рассматриваются для сравнения подходов к совместному смягчению атак в облачных и туманных вычислениях, что может улучшить масштабируемость разработанной в статье модели. В [11] предлагаются методы, которые могут улучшить масштабируемость и эффективность защиты IoT-сетей, что является важным аспектом обеспечения безопасности в условиях растущего числа устройств и объемов трафика. В [12] представлена методика защиты децентрализованных IoT-сетей от многовекторных DDoS-атак с использованием блокчейн-технологий и методов глубокого обучения. Предлагается двухэтапный подход, который сочетает предотвращение и обнаружение атак, что позволяет эффективно управлять сетевыми угрозами и повышать уровень безопасности. В [13] предложен метод для обнаружения многовекторных атак на основе использования MLP. Акцентируется внимание на важности анализа сетевого трафика для выявления аномальных паттернов, что является ключевым моментом для повышения точности обнаружения атак. В [14] проведен обзор различных методов выявления DDoS-атак в IoT-сетях. Рассматриваются сигнатурные, аномалийные и гибридные подходы, перечислены их преимущества и недостатки в контексте IoT. В [15] предложен математический аппарат для оценки методов обнаружения вторжений в условиях многовекторных атак 5-го поколения. Используется комбинация методов принятия решений, что позволяет оценить различные критерии эффективности систем обнаружения атак, включая точность и адаптивность.

Для построения имитационной модели масштабируемого метода выявления многовекторных атак с учетом ограничений вычислительных и информационных ресурсов IoT-устройств были выбраны 3 наиболее подходящих аналога:

1. Метод Deep Ensemble Learning with Pruning [7], использующий комбинацию CNN и LSTM для анализа сетевого трафика и техники обрезки нейронов для снижения вычислительных затрат.
2. Модель Threat Analysis [8], использующая LSTM для анализа маршрутизирующих атак в IoT-сетях с адаптивной оптимизацией гиперпараметров с помощью алгоритма Mayfly.
3. Методика Blockchain-based Threat Intelligence Framework [3], комбинирующая технологию блокчейн с глубоким обучением для защиты IoT-сетей от многовекторных DDoS-атак.

## 2. ИМИТАЦИОННАЯ МОДЕЛЬ МАСШТАБИРУЕМОГО МЕТОДА ВЫЯВЛЕНИЯ МНОГОВЕКТОРНЫХ АТАК С УЧЕТОМ ОГРАНИЧЕНИЙ ВЫЧИСЛИТЕЛЬНЫХ И ИНФОРМАЦИОННЫХ РЕСУРСОВ IoT-УСТРОЙСТВ

Разрабатываемая модель, минимизируя вычислительные затраты и учитывая ограничения, присущие ресурсам IoT-устройств, должна выявлять многовекторные атаки с высокой точностью и быть пригодной для масштабирования в больших децентрализованных IoT-сетях.

Для построения имитационной модели использованы следующие основные компоненты:

1. Модуль анализа сетевого трафика (CNN + LSTM/GRU).
2. Алгоритм Mayfly для адаптивной оптимизации гиперпараметров.

[11] LAMOA (Learning-based Adaptive Mayfly Optimization Algorithm) – глубинное обучение с использованием адаптивного алгоритма оптимизации Mayfly.

3. Блокчейн-ориентированный механизм консенсуса Proof of Voting (PoV) для децентрализованной верификации.
4. Техника обрезки нейронов для уменьшения вычислительных затрат.

Рассмотрим более подробно эти компоненты.

### 2.1. Модуль анализа сетевого трафика (CNN + LSTM/GRU)

Модуль анализа сетевого трафика и выявления аномалий в последовательности данных использует гибридную архитектуру CNN и LSTM (или GRU для уменьшения вычислительных затрат), где:

- Convolutional Neural Network обрабатывает пространственные признаки сетевого трафика. Входные данные представляются в виде многомерного тензора, где каждый элемент представляет характеристики сетевых пакетов (например, время, размер, тип протокола). Сверточные слои выделяют пространственные закономерности в трафике;
- Long Short-Term Memory (или GRU) применяется для анализа временных зависимостей. Это помогает выявлять сложные многовекторные атаки, которые проявляются на разных временных интервалах. Long Short-Term Memory сохраняет информацию о предыдущих состояниях трафика и помогает прогнозировать будущие события, что важно для обнаружения долговременных атак, таких как DDoS.

Модуль анализа сетевого трафика в имитационной модели основывается на гибридной архитектуре, сочетающей сверточные нейронные сети для анализа пространственных зависимостей сетевого трафика и рекуррентные нейронные сети (LSTM или GRU) для анализа временных зависимостей. Такая структура позволяет эффективно анализировать многовекторные атаки, которые могут проявляться через сложные аномалии в пространственных и временных характеристиках сетевого трафика.

Сверточная нейронная сеть используется для извлечения пространственных признаков из сетевого трафика, представленного в виде многомерных данных (тензора). Входной трафик, который включает такие параметры, как временные шаги, размер пакета, тип протокола, IP-адреса и другие метрики, преобразуется в тензор размерности $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$, где $h$ – высота тензора (число пакетов или временных шагов); $w$ – ширина тензора (число признаков или характеристик на 1 пакет); $c$ – количество каналов (например, это может быть разбиение по протоколам или типам данных).

Основное уравнение для операции свертки запишется как

$$\mathbf{Y}_{i,j,k} = \sum_{m=1}^{h_k} \sum_{n=1}^{w_k} \mathbf{X}_{i+m,j+n,c} \mathbf{W}_{m,n,k} + b_k, \qquad (1)$$

где $\mathbf{X}_{i,j,c}$ – входной тензор данных для позиции $(i, j)$ на канале $c$; $\mathbf{W}_{m,n,k}$ – фильтр свертки с размерами $h_k \times w_k$ для канала $k$; $b_k$ – смещение (bias) для канала $k$; $\mathbf{Y}_{i,j,k}$ – результат свертки для канала $k$.

После выполнения операции свертки (1) для повышения нелинейности применяется функция активации:

$$\mathbf{Z}_{i,j,k} = \mathrm{ReLU}(\mathbf{Y}_{i,j,k}) = \max(0, \mathbf{Y}_{i,j,k}),$$

где ReLU (Rectified Linear Unit) является одной из самых распространенных функций активации, которая оставляет только положительные значения.

Сверточная сеть выделяет пространственные паттерны в данных сетевого трафика, такие как частота пакетов и корреляция различных параметров трафика. После того, как пространственные признаки выделены с помощью CNN, они передаются в LSTM для анализа временных зависимостей. Long Short-Term Memory учитывает динамику изменения трафика во времени и помогает выявлять многовекторные атаки, которые могут проявляться через последовательные изменения в поведении сети.

Рассмотрим основные компоненты LSTM.

1. Входные ворота, управляющие выбором нового входного состояния для обновления состояния памяти. Активация входных ворот в момент времени $t$ осуществляется в соответствии со следующим выражением:

$$i_t = \sigma(\mathbf{W}_{\mathrm{in}} \cdot [h_{t-1}, \mathbf{x}_t] + \mathbf{b}_{\mathrm{in}}), \qquad (2)$$

где $\mathbf{x}_t$ – входной вектор в момент времени $t$ (пространственные признаки, извлеченные из CNN); $h_{t-1}$ – скрытое состояние на предыдущем шаге времени; $\mathbf{W}_{\mathrm{in}}$ – матрица весов для входных ворот; $\mathbf{b}_{\mathrm{in}}$ – вектор смещений для входных ворот; in – индекс, обозначающий входные ворота; σ – сигмоида, нормализующая значения в интервале [0; 1].

2. Забывающие ворота определяют, какая часть предыдущего состояния должна быть сохранена. Функция $f_t$ – активация забывающих ворот может быть записана как:

$$f_t = \sigma(\mathbf{W}_{\mathrm{f}} \cdot [h_{t-1}, \mathbf{x}_t] + \mathbf{b}_{\mathrm{f}}), \qquad (3)$$

где $\mathbf{W}_{\mathrm{f}}$ – матрица весов для забывающих ворот; $\mathbf{b}_{\mathrm{f}}$ – вектор смещений для забывающих ворот; f – индекс, обозначающий забывающие ворота (англ. forget – забывать).

3. Состояние памяти $C_t$ обновляется на каждом временном шаге с учетом новой информации следующим образом:

$$C_t = f_t\, C_{t-1} + i_t \tanh(\mathbf{W}_c \cdot [h_{t-1}, \mathbf{x}_t] + \mathbf{b}_c), \quad (4)$$

где $C_{t-1}$ – предыдущее состояние памяти; $C_t$ – новое состояние памяти; $f_t$ – забывающие ворота; $i_t$ – входные ворота; tanh – гиперболический тангенс, являющийся функцией активации, используемой для обновления состояния памяти; $\mathbf{W}_c$ – матрица весов для состояния памяти; $[h_{t-1}, \mathbf{x}_t]$ – конкатенация скрытого состояния на предыдущем шаге и текущего входного вектора; $\mathbf{b}_c$ – вектор смещения для обновления состояния памяти; c – индекс, обозначающий сбор данных в память (англ. collect – собирать).

4. Выходные ворота контролируют, какая часть состояния памяти используется для обновления скрытого состояния. Активация выходных ворот осуществляется следующим образом:

$$o_t = \sigma(\mathbf{W}_o \cdot [h_{t-1}, \mathbf{x}_t] + \mathbf{b}_o), \quad (5)$$

где $o_t$ – активация выходных ворот; σ – сигмовидная функция активации; $\mathbf{W}_o$ – матрица весов для выходных ворот; $\mathbf{b}_o$ – вектор смещения выходных ворот; o – индекс, обозначающий выходные ворота (англ. output – выход).

Новое скрытое состояние $h_t$ вычисляется по следующему выражению:

$$h_t = o_t \tanh(C_t), \quad (6)$$

где $h_t$ – скрытое состояние в момент времени $t$, которое используется для окончательной классификации сетевого трафика; $C_t$ – текущее состояние памяти.

Вместо LSTM можно использовать GRU, которая является более легкой по вычислительным затратам модификацией. GRU объединяет забывающие и входные ворота в единое обновляющее ворота, что снижает вычислительные затраты и улучшает работу модели в условиях ограниченных ресурсов.

Рассмотрим основные компоненты GRU.

1. Обновляющие ворота

$$z_t = \sigma(\mathbf{W}_z \cdot [h_{t-1}, \mathbf{x}_t] + \mathbf{b}_z), \quad (7)$$

где $z_t$ – активация обновляющих ворот, которая контролирует, как сильно текущее состояние влияет на предыдущее; z – индекс, отображающий обновляющие ворота (англ. zero – ноль, обнуление).

2. Ворота сброса

$$r_t = \sigma(\mathbf{W}_r \cdot [h_{t-1}, \mathbf{x}_t] + \mathbf{b}_r), \quad (8)$$

где $r_t$ – активация сбросных ворот, которые контролируют, как сильно предыдущее состояние должно быть забыто; r – индекс, отображающий ворота сброса (англ. reset – сброс).

3. Обновление скрытого состояния

$$h_t = (1 - z_t)\, h_{t-1} + z_t \tanh(\mathbf{W}_h[\mathbf{r}_t h_{t-1}, \mathbf{x}_t] + \mathbf{b}_h), \quad (9)$$

где $h_t$ – обновленное скрытое состояние в момент времени $t$; $\mathbf{W}_h$ – матрица весов для обновления скрытого состояния; h – индекс, отображающий ворота скрытого состояния (англ. hidden – скрытый).

Рекуррентные нейронные сети GRU используют меньше параметров, чем LSTM, что делает их более подходящими для задач, требующих меньших вычислительных затрат, таких как работа в условиях ограниченных ресурсов IoT-устройств.

После обработки данных CNN и LSTM/GRU модель использует полносвязный слой для окончательной классификации трафика. Этот слой вычисляет вероятности принадлежности данных к одному из классов, например, нормальному трафику или атаке:

$$P_{\text{attack}} = \text{Softmax}(\mathbf{W}_{\text{out}} h_T + \mathbf{b}_{\text{out}}), \quad (10)$$

где $P_{\text{attack}}$ – вероятность того, что входной трафик является атакой; $\mathbf{W}_{\text{out}}$ – веса выходного слоя; $h_T$ – скрытое состояние на последнем временном шаге; $\mathbf{b}_{\text{out}}$ – смещение выходного слоя; Softmax нормализует выходные значения в вероятности.

Основными переменными модели являются: $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$ – входной тензор сетевого трафика; $\mathbf{W}$ – весовые матрицы слоев (CNN, LSTM/GRU); $\mathbf{b}$ – смещения слоев (CNN, LSTM/GRU); $i_t, f_t, o_t$ – входные, забывающие и выходные ворота в LSTM; $z_t, r_t$ – обновляющие и сбросные ворота в GRU; $h_t$ – скрытое состояние на шаге времени $t$; $C_t$ – состояние памяти в LSTM; $P_{\text{attack}}$ – вероятность того, что трафик является атакующим.

Модуль анализа сетевого трафика, основанный на гибридной архитектуре CNN+LSTM (или GRU), сочетает пространственные и временные зависимости сетевых данных. Такая архитектура позволяет эффективно выявлять многовекторные атаки в IoT-сетях, что особенно важно для систем с ограниченными вычислительными ресурсами.

Модель работает следующим образом:

1. Входной трафик преобразуется в многомерный тензор, который поступает на сверточные слои для выделения признаков.
2. Выделенные признаки поступают в LSTM/GRU для анализа временных зависимостей.
3. Модель классифицирует трафик как нормальный или атакующий.

### 2.2. Адаптивная оптимизация гиперпараметров с использованием адаптированного алгоритма Mayfly

Для улучшения эффективности и настройки модели в зависимости от конкретных условий

сети (например, объема данных, типа атак) применим адаптивный алгоритм Mayfly [8]. Алгоритм Mayfly помогает автоматически находить оптимальные гиперпараметры модели, такие как:

- количество слоев CNN и LSTM;
- число фильтров и нейронов на каждом слое;
- скорость обучения модели.

Адаптированный алгоритм Mayfly позволяет ускорить процесс настройки модели и обеспечить ее оптимальную производительность без необходимости ручной настройки. Алгоритм использует эволюционные методы для поиска оптимальных параметров и адаптируется в процессе обучения модели.

Основными шагами адаптированного алгоритма Mayfly являются:

1. Инициализация популяции.
2. Мужские и женские особи: разделение на 2 группы с разными поисковыми стратегиями.
3. Глобальный и локальный поиск: поиск лучших решений мужскими и женскими особями.
4. Эволюция и обновление скоростей и позиций.
5. Сближение и размножение.

Основными переменными и параметрами алгоритма являются:

$N$ – количество особей в популяции;

$x_i^{\mathrm{m}}$ – положение мужской особи $i$ в пространстве решений (значение гиперпараметров);

$x_i^{\mathrm{f}}$ – положение женской особи $i$;

$v_i^{\mathrm{m}}$ – скорость движения мужской особи;

$v_i^{\mathrm{f}}$ – скорость движения женской особи;

$\alpha$, $\beta$, $\gamma$ – коэффициенты управления движением особей (коэффициенты инерции, ускорения и взаимодействия);

$g_{\mathrm{best}}$ – глобально лучшее решение, найденное всеми особями;

$p_{\mathrm{best}}$ – личное лучшее решение для каждой особи;

$\lambda$ – коэффициент привлечения для сближения мужских и женских особей;

$\in$ – параметр случайного отклонения, влияющий на мутацию в поиске.

### 2.2.1. Инициализация популяции

Алгоритм Mayfly начинается с инициализации начальной популяции особей (гиперпараметров) случайным образом в поисковом пространстве. Каждая особь представляет собой вектор гиперпараметров модели:

$$\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}],$$

где $d$ – размерность пространства гиперпараметров (например, количество слоев, нейронов, скорость обучения и т.д.).

Положение каждой особи $x_i$ в пространстве гиперпараметров инициализируется случайным образом:

$$\mathbf{x}_i^{\mathrm{m}}(0), \mathbf{x}_i^{\mathrm{f}}(0) \sim \mathrm{Uniform}(\mathbf{x}_{\min}, \mathbf{x}_{\max}),$$

где $\mathbf{x}_{\min}$ и $\mathbf{x}_{\max}$ – границы пространства гиперпараметров; $\mathrm{Uniform}(\mathbf{x}_{\min}, \mathbf{x}_{\max})$ – функция выбора случайных значений из интервала $[\mathbf{x}_{\min}; \mathbf{x}_{\max}]$.

### 2.2.2. Обновление скорости и положения мужских особей

Мужские особи ищут решения в глобальном пространстве, обновляя свое положение на основе личного лучшего решения $p_{\mathrm{best}}$ и глобального лучшего решения $g_{\mathrm{best}}$. Скорость обновления положения особи вычисляется по правилу:

$$\mathbf{v}_i^{\mathrm{m}}(t+1) = \alpha \mathbf{v}_i^{\mathrm{m}}(t) + \beta_1 r_1 (p_{\mathrm{best},i} - \mathbf{x}_i^{\mathrm{m}}(t)) + \\ + \beta_2 r_2 (g_{\mathrm{best},i} - \mathbf{x}_i^{\mathrm{m}}(t)),$$

где $\alpha$ – коэффициент инерции (контролирует, насколько сильно скорость предыдущего шага влияет на текущее положение); $\beta_1$, $\beta_2$ – коэффициенты ускорения, которые управляют влиянием личного и глобального лучшего решения на обновление скорости; $r_1$, $r_2 \sim \mathrm{Uniform}(0, 1)$ – случайные значения, которые обеспечивают случайное отклонение в поиске.

Положение каждой мужской особи обновляется с учетом его новой скорости:

$$\mathbf{x}_i^{\mathrm{m}}(t+1) = \mathbf{x}_i^{\mathrm{m}}(t) + \mathbf{v}_i^{\mathrm{m}}(t+1).$$

### 2.2.3. Обновление скорости и положения женских особей

Женские особи выполняют локальный поиск, обновляя свое положение, опираясь на расстояние до мужских особей. Скорость обновления положения женской особи вычисляется с учетом взаимодействия с мужскими особями:

$$\mathbf{v}_i^{\mathrm{f}}(t+1) = \lambda (x_i^{\mathrm{m}}(t) - x_i^{\mathrm{f}}(t)) + \in,$$

где $\lambda$ – коэффициент привлечения между мужскими и женскими особями; $\in$ – случайное отклонение для обеспечения разнообразия решений.

Положение женских особей обновляется следующим образом:

$$\mathbf{x}_i^{\mathrm{f}}(t+1) = \mathbf{x}_i^{\mathrm{f}}(t) + \mathbf{v}_i^{\mathrm{f}}(t+1).$$

### 2.2.4. Оценка решений

Каждая особь оценивается с помощью целевой функции (fitness function), которая может быть

связана с точностью модели, временем обучения, сложностью модели и другими параметрами. Функция оценки $F(\mathbf{x}_i)$ для каждой особи рассчитывается как:

$$F(\mathbf{x}_i) = \text{Evaluation Model}(\mathbf{x}_i),$$

где $\mathbf{x}_i$ – гиперпараметры, которые особь представляет, а Evaluation Model – это функция, оценивающая производительность модели для данных гиперпараметров.

#### 2.2.5. Сближение и размножение

После обновления скоростей и позиций мужские и женские особи сближаются, что моделирует этап размножения в алгоритме. Когда мужские и женские особи достигают определенной близости, происходит кроссинговер и мутация:

- кроссинговер передает часть генетической информации (гиперпараметров) от мужских особей к женским особям

$$\mathbf{x}_{\text{new}} = \lambda \mathbf{x}_i^{\text{m}} + (1-\lambda)\mathbf{x}_i^{\text{f}},$$

где $\mathbf{x}_{\text{new}}$ – новое значение (положение) особи, полученное в результате кроссинговера; $\mathbf{x}_i^{\text{m}}$ – текущее положение мужской особи $i$; $\mathbf{x}_i^{\text{f}}$ – текущее положение женской особи $i$; $\lambda$ – коэффициент, определяющий вес влияния мужской особи в новом значении (обычно $\lambda$ находится в диапазоне от 0 до 1).

- мутация выполняет случайное изменение некоторых параметров с вероятностью $p_{\text{mut}}$.

#### 2.2.6. Критерий завершения

Алгоритм Mayfly выполняется до тех пор, пока не будет выполнено одно из следующих условий: достигнуто максимальное количество итераций $T_{\max}$ или не наблюдается улучшение функции оценки в течение нескольких последовательных итераций.

### 2.3. Механизм децентрализованной верификации на основе блокчейн-технологии

Для обеспечения безопасности и надежности системы в условиях децентрализованных IoT-сетей применяется блокчейн-ориентированный механизм консенсуса PoV [3]. Основными функциями этого компонента являются

- децентрализованная верификация данных об атаках, при которой несколько узлов сети анализируют сетевой трафик и передают информацию о возможных атаках в распределенный реестр;
- валидация блоков происходит через голосование узлов-валидаторов, если больше 50% узлов подтверждают атаку, информация о ней записывается в распределенный реестр, а вредоносные IP-адреса блокируются с помощью смарт-контрактов.

Модуль децентрализованной верификации использует блокчейн-технологии для обеспечения безопасности данных и предотвращения атак в IoT-сетях. Этот модуль работает на основе механизма консенсуса PoV, который позволяет узлам сети (валидаторам) голосовать за блоки данных о трафике, атаках или состоянии сети. Блокчейн обеспечивает защиту от подделки данных, децентрализованное хранение и автоматическое выполнение действий, таких как блокировка вредоносных IP-адресов, через смарт-контракты.

Основными элементами и переменными являются:

$B$ – блок данных, содержащий информацию о сетевом трафике, выявленных атаках или обновлениях состояния сети;

$N$ – количество узлов (валидаторов) в блокчейн-сети;

$V_i$ – голос валидатора $i$ за принятие или отклонение блока;

$P_{\text{valid}}$ – вероятность того, что блок будет признан действительным;

$T_B$ – время валидации блока;

AM – смарт-контракт (Action Module), который содержит данные об атаках и блокировках IP-адресов.

В данном подразделе представлены ключевые шаги процесса обработки сетевого трафика и обеспечения безопасности в блокчейн-системе. Эти шаги описывают, как узлы сети взаимодействуют для выявления аномалий, верификации данных и автоматической блокировки подозрительных IP-адресов. Каждый шаг играет важную роль в создании надежной и эффективной системы защиты от кибератак, обеспечивая целостность данных и быструю реакцию на угрозы.

Шаги процесса обработки сетевого трафика и обеспечения безопасности в блокчейн-системе состоят в следующем:

Шаг 1. Формирование блока данных.

Каждый узел блокчейн-сети обрабатывает поступающий сетевой трафик и, если обнаружена аномалия или подозрительная активность (например, многовекторная атака), узел формирует блок данных $B$. Этот блок включает следующие элементы:

$$B = \{\text{Block ID, Data, Previous Hash, Timestamp, Signature}\},$$

где Block ID – уникальный идентификатор блока; Data – информация о трафике и возможных атаках (например, IP-адреса, тип атаки, временные

метки); Previous Hash – хеш предыдущего блока в блокчейне для поддержания непрерывной цепочки; Timestamp – время создания блока; Signature – цифровая подпись узла, который сформировал блок.

Шаг 2. Механизм консенсуса PoV.

После создания блока он передается другим узлам сети для верификации с использованием механизма консенсуса PoV.

Валидация блока осуществляется путем голосования узлов сети в следующем порядке:

- каждый узел анализирует блок данных $B$, проверяет его целостность и достоверность, а затем отправляет свой голос $V_i$ (голосование может быть бинарным: $V_i = 1$ – за принятие блока, $V_i = 0$ – за отклонение блока);
- вероятность того, что блок будет признан действительным, рассчитывается следующим образом:

$$P_{\text{valid}} = \frac{\sum_{i=1}^{N} V_i}{N}.$$

Если $P_{\text{valid}} \geq 0.5$ (большинство узлов поддерживают блок), то блок считается действительным и добавляется в распределенный реестр, если $P_{\text{valid}} < 0.5$, то блок отклоняется.

Шаг 3. Обновление распределенного реестра.

После достижения консенсуса и подтверждения блока $B$ он добавляется в распределенный реестр. Каждая запись в распределенном реестре связана с предыдущим блоком через хеш Previous Hash, что обеспечивает непрерывную и неизменяемую цепочку данных. Новый блок добавляется к распределенный реестр:

$$B_{\text{new}} = \{\text{Hash}(B_{\text{prev}}), \text{Data}_{\text{new}}, \text{Timestamp}_{\text{new}}, \text{Signature}_{\text{new}}\},$$

где $\text{Hash}(B_{\text{prev}})$ – хеш предыдущего блока, который гарантирует целостность всей цепочки.

Шаг 4. Использование смарт-контрактов для автоматической блокировки IP-адресов.

Блокчейн-система использует смарт-контракты для автоматического выполнения действий при обнаружении атаки. Смарт-контракты позволяют автоматически блокировать IP-адреса, отправлять уведомления и обновлять черные списки в сети. Структура смарт-контракта может выглядеть следующим образом:

$$\text{AM} = \{\text{Source IP (SIP)}, \text{Destination IP (DIP)}, \text{Signature}, \text{Blacklisted IP}, \text{Attack Label}\},$$

где Source IP (SIP) – IP-адрес, от которого поступает трафик; Destination IP (DIP) – IP-адрес целевого устройства; Signature – цифровая подпись данных для подтверждения подлинности информации; Blacklisted IP – список IP-адресов, которые были заблокированы после обнаружения атаки; Attack Label – метка атаки, которая содержит тип атаки (например, DDoS, SQL-инъекция, многовекторная атака).

Шаг 5. Процесс блокировки IP-адресов.

Как только блок с данными об атаке подтвержден и добавлен в блокчейн, смарт-контракт автоматически выполняет блокировку вредоносных IP-адресов в сети. Например, если обнаружен DDoS-трафик, IP-адрес атакующего устройства SIP добавляется в черный список Blacklisted IP через выполнение смарт-контракта

$$\text{AM(SIP)} = \text{Blacklisted IP}.$$

Эти данные обновляются на всех узлах сети через распределенную блокчейн-структуру, что гарантирует согласованность действий всех участников.

Шаг 6. Время валидации блока.

Для каждого блока $B$ вычисляется время валидации $T_B$, которое зависит от времени голосования всех узлов $t_{\text{vote}}$, времени выполнения смарт-контрактов $t_{\text{contract}}$ и времени передачи блока между узлами $t_{\text{transmit}}$:

$$T_B = t_{\text{vote}} + t_{\text{contract}} + t_{\text{transmit}}.$$

Оптимизация времени валидации блока критична для работы IoT-сетей с ограниченными ресурсами и высокой скоростью обмена данными.

### 2.4. Техника обрезки нейронов (pruning)

Для уменьшения вычислительных затрат и оптимизации работы модели на маломощных IoT-устройствах используется техника обрезки нейронов. После обучения модели малозначимые нейроны и их связи удаляются, что сокращает объем модели без значительного ухудшения ее точности.

Применение техники обрезки нейронов включает следующие этапы:

- после обучения нейронной сети анализируются веса ее связей. Если веса находятся ниже заданного порога, связи удаляются;
- модель перезапускается с уменьшенным количеством нейронов и параметров, что уменьшает ее вычислительную сложность и требования к памяти.

Основная идея состоит в том, чтобы удалить ненужные или малозначимые нейроны или изменить веса после обучения модели, незначительно снижая ее производительность.

Основными элементами и переменными являются:

**W** – матрица весов нейронной сети;

**b** – вектор смещений (bias) нейронов;

$f(\mathbf{W})$ – функция активации для весов сети;

θ – пороговое значение для удаления весов;

**M** – матрица маски (Masking matrix) для обрезки весов;

$n_{\text{total}}$ – общее количество параметров (весов) в нейронной сети;

$n_{\text{pruned}}$ – количество удаленных (обрезанных) весов;

$p$ – доля удаленных весов или нейронов.

### 2.4.1. Определение значимости весов и нейронов

После того как нейронная сеть обучена, необходимо определить, какие веса **W** в нейронной сети оказывают наименьшее влияние на выходные значения и могут быть удалены. Это делается путем вычисления значимости каждого веса $W_{i,j}$. В качестве меры значимости можно использовать метрику абсолютного значения веса, в которой чем меньше значение веса, тем менее значим этот вес для активации нейрона

$$\text{Significance}(W_{i,j}) = |W_{i,j}|. \quad (11)$$

Если вес близок к нулю, то его влияние на выход сети минимально, и такой вес может быть удален.

### 2.4.2. Применение порога обрезки

Для того чтобы решить, какие веса удалить, вводится пороговое значение θ. Веса, абсолютное значение которых меньше θ, считаются малозначимыми и удаляются (приравниваются к 0):

$$W_{i,j} = 0, \text{ если } |W_{i,j}| < \theta. \quad (12)$$

Порог обрезки выбирается эмпирически или оптимизируется в ходе экспериментов. Этот порог может быть статическим или динамическим, адаптируемым на основе анализа структуры модели.

### 2.4.3. Матрица маски

Для того чтобы обрезка весов не затронула нейроны, которые оказывают значительное влияние на выходные значения модели и ее производительность, используется матрица маски **M**, обозначающая, какие веса должны быть сохранены, а какие – обнулены:

$$M_{i,j} = \begin{cases} 1, \text{ если } |W_{i,j}| \geq \theta, \\ 0, \text{ если } |W_{i,j}| < \theta. \end{cases} \quad (13)$$

Обрезанная матрица весов:

$$\mathbf{W}_{\text{pruned}} = \mathbf{W} \odot \mathbf{M}, \quad (14)$$

где $\odot$ – поэлементное произведение матрицы весов **W** и маски **M**. Это гарантирует, что только значимые веса будут участвовать в вычислениях, а малозначимые веса будут исключены.

### 2.4.4. Оценка доли обрезанных весов

Доля обрезанных весов или нейронов вычисляется следующим образом:

$$p = \frac{n_{\text{pruned}}}{n_{\text{total}}}, \quad (15)$$

где $n_{\text{pruned}}$ – количество обрезанных весов, т.е. весов, для которых $|W_{i,j}| < \theta$; $n_{\text{total}}$ – общее количество весов в модели.

### 2.4.5. Адаптивная обрезка (iterative pruning)

Простой порог обрезки может быть недостаточно эффективным для всех слоев сети, особенно для глубоких моделей с множеством слоев. Поэтому может применяться итеративная обрезка: веса обрезаются не за один шаг, а поэтапно, с постепенным увеличением порога θ.

На каждой итерации значения весов пересчитываются с учетом маски:

$$\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} \odot \mathbf{M}. \quad (16)$$

Затем сеть переобучается на новых данных, чтобы восстановить ее точность после обрезки. Этот процесс повторяется несколько раз до тех пор, пока доля обрезанных весов не достигнет желаемого уровня $p$.

### 2.4.6. Основные шаги разработанной имитационной модели и метрики для оценки качества модели после обрезки весов

Основными шагами работы модели являются:

1) анализ трафика с помощью CNN + LSTM для выделения пространственно-временных признаков и выявления аномалий;
2) оптимизация модели: алгоритм Mayfly автоматически настраивает гиперпараметры модели в зависимости от условий сети и данных, что обеспечивает ее адаптивность;
3) верификация и блокировка подозрительных IP-адресов и трафика осуществляются через блокчейн-консорциум. В случае подтверждения атаки на IP-адреса блокируются с помощью смарт-контрактов;

4) применение техники обрезки нейронов: после начального обучения и верификации модели применяется обрезка нейронов для уменьшения вычислительной сложности и адаптации модели к ресурсам IoT-устройств.

После обрезки важно оценить, как изменились производительность и ресурсоемкость модели. Основными метриками качества модели после обрезки весов являются:

1) доля правильных предсказаний среди всех общего количества предсказаний (точность правильных предсказаний):

$$E_1 = \text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (17)$$

где TP – количество истинно-положительных предсказаний (правильные предсказания атак), TN – количество истинно-отрицательных предсказаний (правильные предсказания нормального трафика), FP – количество ложноположительных предсказаний (ложные срабатывания), FN – количество ложноотрицательных предсказаний (пропущенные атаки);

2) полнота обучения:

$$E_2 = \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (18)$$

определяющая способность модели обнаруживать все атаки в выборке;

3) доля истинно-положительных предсказаний среди всех положительных предсказаний (точность истинно-положительных предсказаний):

$$E_3 = \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (19)$$

определяющая насколько предсказания модели о положительных классах являются корректными;

4) F1-мера:

$$E_4 = 2\frac{E_2 E_3}{E_2 + E_3}, \quad (20)$$

являющаяся гармоническим средним между полнотой обучения и долей истинно-положительных предсказаний;

5) время обработки данных и выполнения предсказаний моделью:

$$E_5 = T_{\text{pruned}} = T_{\text{original}}(1 - p), \quad (21)$$

где $T_{\text{pruned}}$ – время вычислений после обрезки, $T_{\text{original}}$ – время вычислений исходной модели, $p$ – доля обрезанных нейронов;

6) использование памяти (уменьшение использования памяти после удаления весов):

$$E_6 = M_{\text{pruned}} = M_{\text{original}}(1 - p), \quad (22)$$

где $M_{\text{pruned}}$ – объем памяти, необходимый для хранения обрезанной модели, а $M_{\text{original}}$ – память, необходимая для исходной модели.

## 3. РЕАЛИЗАЦИЯ И ЭКСПЕРИМЕНТ

### 3.1. Методология эксперимента

В данном разделе проводится экспериментальная проверка разработанной имитационной модели выявления многовекторных атак с учетом ограничений вычислительных и информационных ресурсов IoT-устройств.

В эксперименте используется CIC IoT Dataset 2023, который содержит данные сетевого трафика, как нормального, так и атакующего, с разнообразными характеристиками и признаками, такими как:

- IP-адреса (Source/Destination);
- порты (Source/Destination);
- время соединения;
- размер пакета;
- протоколы (TCP[12], UDP[13], HTTP, DNS);
- метки атак (например, DDoS, SQL-инъекция, Brute Force).

Датасет разбит на несколько классов:

- нормальный трафик;
- атакующий трафик (разные типы атак).

Эксперимент сравнивает результаты работы разработанной имитационной модели с несколькими аналогичными методами, применяющимися для обнаружения атак в IoT-сетях.

Данные CIC IoT Dataset 2023 были нормализованы с использованием Min-Max Scaling для приведения всех признаков к диапазону [0; 1]. Выбросы удалялись с применением метода межквартильного размаха (Interquartile Range, IQR), а новые признаки генерировались на основе агрегации временных характеристик трафика (например, среднее количество пакетов за 10 секунд).

Для тестирования использовалась рабочая станция на основе ноутбука Macbook Pro (Apple Inc., США) с процессором M2 Pro (включает в себя 12 процессорных ядер (8 производительных и 4 энергоэффективных), 19 графических ядер и 16-ядерный нейронный сопроцессор), 16 ГБ оперативное запоминающее устройство с пропускной способностью порядка 200 ГБ/с.

[12] Transmission Control Protocol – протокол управления передачей.

[13] User Datagram Protocol – протокол пользовательских датаграмм.

### 3.2. Структура эксперимента

Эксперимент проводился в несколько этапов:

1. Подготовка данных, на котором данные CIC IoT Dataset 2023 разбиваются на обучающую и тестовую выборки в соотношении 70/30. Далее проводится предварительная обработка данных: нормализация признаков, удаление выбросов, генерация новых признаков (если необходимо).
2. Обучение моделей, на котором обучение предложенной модели проводится на основе гибридной архитектуры CNN + LSTM/GRU с применением техники обрезки нейронов для сокращения вычислительных затрат.

Оптимизация гиперпараметров модели осуществляется с использованием алгоритма Mayfly.

Для сравнения обучаются и другие модели, такие как Random Forest, SVM, Deep Learning (MLP).

Время обработки одного пакета данных определяется как сумма времени свертки (1), времени обработки LSTM/GRU (2)–(9) и времени классификации (10). Благодаря применению техники обрезки нейронов (11)–(16) модель значительно уменьшает количество параметров, что приводит к снижению вычислительных затрат и улучшению производительности на устройствах с ограниченными ресурсами. Это позволяет эффективно использовать модель в реальных условиях, таких как системы мониторинга в реальном времени, где важна быстрая реакция и минимальное потребление памяти. Для оценки качества имитационной модели используются метрики (17)–(22).

### 3.3. Анализ полученных результатов

Для оценки эффективности предложенной модели проведены экспериментальные исследования, результаты которых представлены в таблице. Таблица содержит результаты работы оценки эффективности по метрикам качества (17)–(22) различных моделей, включая предложенную модель на основе гибридной архитектуры CNN + LSTM/GRU с применением техники обрезки нейронов.

**Таблица.** Результаты эксперимента

| Модель | Метрики качества | | | | | |
|---|---|---|---|---|---|---|
| | $E_1$, % | $E_2$, % | $E_3$, % | $E_4$, % | $E_5$, мс | $E_6$, МБ |
| Random Forest | 96.5 | 95.7 | 97.1 | 96.4 | 35 | 220 |
| SVM | 94.3 | 92.6 | 94.5 | 93.5 | 50 | 250 |
| Deep Learning (MLP) | 97.8 | 97.2 | 98.0 | 97.6 | 20 | 210 |
| Имитационная модель CNN + LSTM/GRU | 99.1 | 99.3 | 98.9 | 99.1 | 12 | 180 |

Рисунок 1 иллюстрирует сравнение результатов работы предложенной имитационной модели и ее аналогов по метрикам качества (17)–(20). На графике можно увидеть, что модель CNN + LSTM/GRU значительно превосходит другие модели по всем указанным метрикам, что подтверждает ее высокую эффективность в обнаружении многовекторных атак.

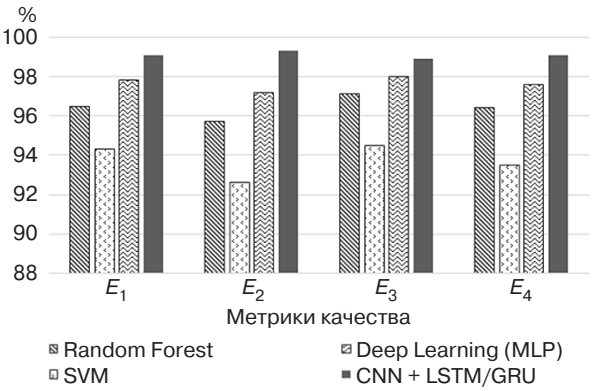


**Рис. 1.** Сравнение результатов работы разработанной имитационной модели CNN + LSTM/GRU и аналогов по метрикам $E_1$ (17) – $E_4$ (20)

На рис. 2 демонстрируются сравнительные результаты работы предлагаемой имитационной модели и ее аналогов по метрике $E_5$ (21). График показывает, что модель CNN + LSTM/GRU имеет наименьшее время обработки данных и наименьшее время выполнение предсказания (12 мс), что делает ее особенно подходящей для применения в реальном времени, в то время как другие модели требуют значительно больше времени.

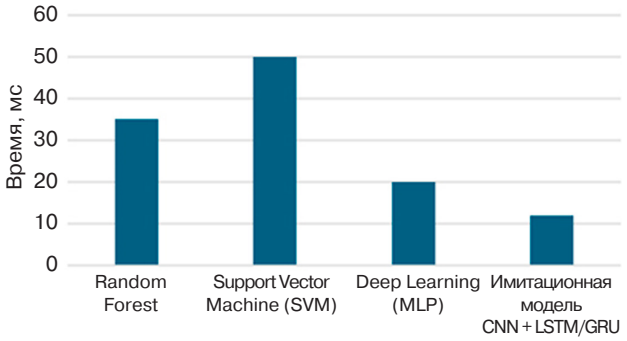


**Рис. 2.** Сравнение результатов работы разработанной имитационной модели CNN + LSTM/GRU и аналогов по метрике $E_5$ (21)

На рис. 3 представлены сравнительные результаты работы предложенной имитационной модели и ее аналогов по метрике $E_6$ (22). На графике видно, что модель CNN + LSTM/GRU требует наименьшего объема памяти (180 МБ) при обработке входных данных объемом 1 млн примеров с 10 признаками, что делает ее более эффективной для использования

на устройствах с ограниченными вычислительными ресурсами по сравнению с другими моделями, такими как Random Forest и SVM.

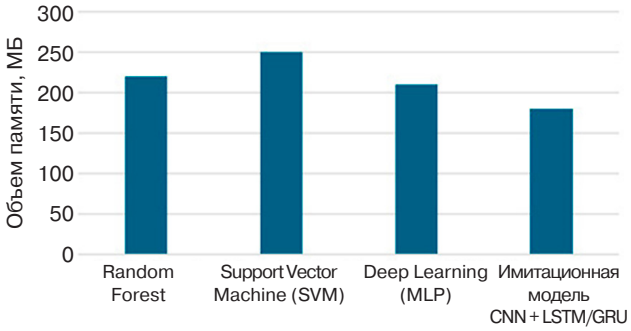


**Рис. 3.** Сравнение результатов работы разработанной имитационной модели CNN + LSTM/GRU и аналогов по метрике $E_6$ (22)

Проведенный эксперимент позволяет сделать следующие выводы:

1. Разработанная имитационная модель достигает высокой точности обнаружения атак на уровне 99.1%, что свидетельствует о ее способности эффективно идентифицировать как известные, так и новые типы атак в реальном времени. Это подтверждает тот факт, что предложенная архитектура, основанная на гибридной модели CNN + LSTM/GRU, является успешной в контексте анализа сетевого трафика.
2. F1-мера разработанной имитационной модели равна 99.1%, что указывает на высокую сбалансированность между точностью истинно-положительных предсказаний (19) и полнотой обучения (18). Это означает, что модель не только точно идентифицирует атаки, но и минимизирует количество ложных срабатываний и пропусков.
3. Время обработки запросов в предложенной имитационной модели при аппаратных ресурсах, указанных в п. 3.1 настоящей статьи, снижено до 12 мс, что делает модель особенно подходящей для систем, требующих быстрой реакции, таких как мониторинг в реальном времени. Это является значительным преимуществом по сравнению с другими моделями, которые требуют больше времени для обработки.
4. Разработанная имитационная модель использует всего 180 МБ памяти, что делает ее подходящей для внедрения на устройствах с ограниченными вычислительными ресурсами. Это особенно важно для IoT-устройств, которые часто имеют ограничения по памяти и вычислительной мощности.
5. Использование техники обрезки нейронов позволило значительно сократить количество параметров модели с 1.5 млн до 300 тысяч, что, в свою очередь, снизило вычислительные затраты на 80% и улучшило производительность. Это подтверждает, что подходы к оптимизации модели имеют решающее значение для ее успешного применения в условиях ограниченных ресурсов.

## ЗАКЛЮЧЕНИЕ

В работе предложена имитационная модель масштабируемого метода выявления многовекторных атак на устройства IoT, которая учитывает ограничения вычислительных и информационных ресурсов. Учитывая растущие угрозы безопасности в IoT-сетях, создание эффективного решения, способного обнаруживать атаки с высокой точностью, стало одной из ключевых задач исследования.

Предложенная модель основана на гибридной архитектуре нейронных сетей, которая сочетает сверточные нейронные сети CNN для анализа пространственных зависимостей и сети долгой краткосрочной памяти LSTM для анализа временных зависимостей сетевого трафика. Важным аспектом является применение техники обрезки нейронов, что позволяет значительно сократить количество параметров модели и снизить вычислительные затраты. Кроме того, использование блокчейн-технологий с механизмом консенсуса PoV обеспечивает безопасность данных и децентрализованную верификацию, что критически важно для защиты IoT-сетей от многовекторных атак.

Экспериментальная оценка, проведенная на датасете CIC IoT Dataset 2023, продемонстрировала высокую эффективность предложенной модели. Модель достигла точности обнаружения атак на уровне 99.1%, что подтверждает ее способность точно идентифицировать как известные, так и новые типы атак в реальном времени. F1-мера, равная 99.1%, указывает на сбалансированность между точностью и полнотой, что является критически важным для систем кибербезопасности, где необходимо минимизировать как ложные срабатывания, так и пропуски атак. В дополнение к высокой точности, время обработки запросов было снижено до 12 мс, что позволяет модели эффективно функционировать в условиях, требующих быстрой реакции, таких как системы мониторинга в реальном времени. Оптимизация использования памяти также была достигнута, и модель продемонстрировала потребление всего 180 МБ, что делает ее подходящей для внедрения на устройствах с ограниченными вычислительными ресурсами.

Таким образом, разработанная имитационная модель превосходит существующие решения по ключевым метрикам, таким как точность, время обработки и использование памяти. Применение гибридной архитектуры, техники обрезки нейронов и децентрализованной верификации обеспечивает высокую эффективность модели в условиях многовекторных угроз IoT.

Данная работа открывает новые горизонты для дальнейших исследований в области кибербезопасности, предлагая эффективные решения для защиты IoT-сетей от сложных киберугроз. В будущем целесообразно продолжить исследование в направлении интеграции дополнительных методов машинного обучения и глубокого обучения для повышения точности и устойчивости модели к новым типам атак. Также стоит рассмотреть возможности оптимизации алгоритмов для уменьшения вычислительных затрат и повышения скорости обработки данных. Важно продолжать изучение вопросов масштабируемости и устойчивости блокчейн-ориентированных решений в условиях увеличения числа устройств и объемов трафика.

## БЛАГОДАРНОСТИ

Данное исследование выполнено при поддержке гранта ИБ МТУСИ по теме «Разработка метода обнаружения вторжений с использованием сценариев многовекторных атак в децентрализованной IoT среде», соглашение № 40469/17-23-К.

## ACKNOWLEDGMENTS

This research was supported by the Information Security grant from the Moscow Technical University of Communications and Informatics (MTUSI) (agreement No. 40469/17-23-K, “Development of an intrusion detection method using multi-vector attack scenarios in a decentralized IoT environment.”

### Вклад авторов

**В.И. Петренко** – идея исследования, планирование исследования и научное редактирование статьи.

**Ф.Б. Тебуева** – идея исследования, планирование исследования, научное редактирование статьи.

**М.Г. Огур** – проведение исследования, написание текста статьи, выполнение экспериментальной части работы, анализ полученных данных и формулирование результатов.

**Г.И. Линец** – консультации по проведению исследования и научному редактированию статьи.

**В.П. Мочалов** – консультации по проведению исследования и научному редактированию статьи.

### Authors’ contributions

**V.I. Petrenko** – research idea, planning the study, and scientific editing the article.

**F.B. Tebueva** – research idea, planning the study, and scientific editing the article.

**M.G. Ogur** – conducting the research, performing the experimental part of the work, analysis of the obtained data, formulating the results, and writing the text of the article.

**G.I. Linets** – consultations on conducting the research and scientific editing the article.

**V.P. Mochalov** – consultations on conducting the research and scientific editing the article.

## СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Sen Ö., Ivanov B., Henze M., Ulbig A. Investigation of Multi-stage Attacks and Defense Modeling for Data Synthesis. In: *Proceedings of the International Conference on Smart Energy Systems and Technologies* (*SEST*). IEEE; 2023. P. 1–12. https://doi.org/10.1109/SEST57387.2023.10257329
2. Lysenko S., Bobrovnikova K., Kharchenko V., Savenko O. IoT Multi-Vector Cyberattack Detection Based on Machine Learning Algorithms: Traffic Features Analysis, Experiments, and Efficiency. *Algorithms*. 2022;15(7):239. https://doi.org/10.3390/a15070239
3. Aguru A., Erukala S. OTI-IoT: A Blockchain-based Operational Threat Intelligence Framework for Multi-vector DDoS Attacks. *ACM Trans. Internet Technol*. 2024;24(3):15.1–15.31. https://doi.org/10.1145/3664287
4. Ipole-Adelaiye N., Tatama F.B., Egena O., Jenom M., Ibrahim L. Detecting Multi-Vector Attack Threats Using Multilayer Perceptron Network. *IRE Journals*. 2024;8(1):119–123.
5. Pakmehr A., Aßmuth A., Taheri N., Ghaffari A. DDoS attack detection techniques in IoT networks: a survey. *Cluster Comput*. 2024;27(4):14637–14668. https://doi.org/10.1007/s10586-024-04662-6
6. Alhakami W. Evaluating modern intrusion detection methods in the face of Gen V multi-vector attacks with fuzzy AHP-TOPSIS. *PLoS One*. 2024;19(5):e0302559. https://doi.org/10.1371/journal.pone.0302559
7. Saiyed M.F., Al-Anbagi I. Deep Ensemble Learning With Pruning for DDoS Attack Detection in IoT Networks. *IEEE Trans. Machine Learning Commun. Networks*. 2024;2:596–616. https://doi.org/10.1109/TMLCN.2024.3395419
8. Liebl S. *Threat Modelling for Internet of Things Devices*. Research Report 2023 of the Technical University OTH Amberg-Weiden. 2023. URL: https://www.researchgate.net/publication/369488078. Дата обращения 25.02.2025. / Accessed February 25, 2025.
9. Aguru A.D., Erukala S.B. A lightweight multi-vector DDoS detection framework for IoT-enabled mobile health informatics systems using deep learning. *Inf. Sci*. 2024;662:120209. https://doi.org/10.1016/j.ins.2024.120209
10. Петренко В.И., Тебуева Ф.Б., Огур М.Г., Линец Г.И., Мочалов В.П. Методика обнаружения и противодействия многовекторным угрозам нарушения информационной безопасности, децентрализованной IoT системы. *Int. J. Open Inf. Technol*. 2025;13(1):14–24.
[Petrenko V.I., Tebueva F.B., Ogur M.G., Linets G.I., Mochalov V.P. Methodology for detecting and countering multi-vector threats to information security of a decentralized IoT system. *Int. J. Open Inf. Technol*. 2025;13(1):13–24 (in Russ.).]

11. Leng S., Guo Y., Zhang L., Hao F., Cao X., Li F., Kou W. Online and Collaboratively Mitigating Multi-Vector DDoS Attacks for Cloud-Edge Computing. In: *ICC 2024 – International Conference on Communications*. 2024. P. 1394–1399. https://doi.org/10.1109/ICC51166.2024.10623052
12. Ali M., Saleem Y., Hina S., Shah G.A. DDoSViT: IoT DDoS attack detection for fortifying firmware Over-The-Air (OTA) updates using vision transformer. *Internet of Things*. 2025;30:101527. https://doi.org/10.1016/j.iot.2025.101527
13. Dalal S., Lilhore U.K., Faujdar N., Simaiya S., et al. Next-generation cyberattack prediction for IoT systems: leveraging multi-class SVM and optimized CHAID decision tree. *J. Cloud Comput*. 2023;12:137. https://doi.org/10.1186/s13677-023-00517-4
14. Zahid F., Funchal G., Melo V., Kuo M.M.Y., et al. DDoS attacks on smart manufacturing systems: A cross-domain taxonomy and attack vectors. In: *2022 20th IEEE International Conference on Industrial Informatics* (*INDIN*). 2022. P. 214–219. https://doi.org/10.1109/INDIN51773.2022.9976172
15. Lungu N., Dash B.B., De U.C., Dash B.B., et al. Multi-vector Monitoring, Detecting and Classifying GPU Side-Channel Attack Vectors on a Secure GPU Execution Framework. In: *2024 8th International Conference on I-SMAC* (*IoT in Social, Mobile, Analytics and Cloud*). 2024. P. 500–505. https://doi.org/10.1109/I-SMAC61858.2024.10714895

## Об авторах

**Петренко Вячеслав Иванович,** к.т.н., доцент, заведующий кафедрой организации и технологии защиты информации, факультет математики и компьютерных наук имени профессора Н.И. Червякова, ФГАОУ ВО «Северо-Кавказский федеральный университет» (355017, Россия, Ставрополь, ул. Пушкина, д. 1). E-mail: vipetrenko@ncfu.ru. Scopus Author ID 57189512011, ResearcherID A-3196-2017, SPIN-код РИНЦ 3923-4295, https://orcid.org/0000-0003-4293-7013

**Тебуева Фариза Биляловна,** д.ф.-м.н., доцент, профессор кафедры вычислительной математики и кибернетики, факультет математики и компьютерных наук имени профессора Н.И. Червякова, ФГАОУ ВО «Северо-Кавказский федеральный университет» (355017, Россия, Ставрополь, ул. Пушкина, д. 1). E-mail: ftebueva@ncfu.ru. Scopus Author ID 57189512319, ResearcherID H-4548-2017, SPIN-код РИНЦ 9343-7504, https://orcid.org/0000-0002-7373-4692

**Огур Максим Геннадьевич,** старший преподаватель, кафедра вычислительной математики и кибернетики, факультет математики и компьютерных наук имени профессора Н.И. Червякова, ФГАОУ ВО «Северо-Кавказский федеральный университет» (355017, Россия, Ставрополь, ул. Пушкина, д. 1). E-mail: ogur26@gmail.com. ResearcherID B-1332-2017, SPIN-код РИНЦ 7180-6971, https://orcid.org/0000-0002-2387-0901

**Линец Геннадий Иванович,** д.т.н., профессор, профессор департамента цифровых, робототехнических систем и электроники, институт перспективной инженерии, ФГАОУ ВО «Северо-Кавказский федеральный университет» (355017, Россия, Ставрополь, ул. Пушкина, д. 1). E-mail: kbytw@mail.ru. Scopus Author ID 6506372022, SPIN-код РИНЦ 1452-6823, https://orcid.org/0000-0002-2279-3887

**Мочалов Валерий Петрович,** д.т.н., профессор, профессор департамента цифровых, робототехнических систем и электроники, институт перспективной инженерии, ФГАОУ ВО «Северо-Кавказский федеральный университет» (355017, Россия, Ставрополь, ул. Пушкина, д. 1). E-mail: mochalov.valery2015@yandex.ru. Scopus Author ID 57202300745, SPIN-код РИНЦ 8695-1648, https://orcid.org/0000-0002-5131-5649

## About the Authors

**Vyacheslav I. Petrenko,** Cand. Sci. (Eng.), Associate Professor, Head of the Department of Organization and Technology of Information Security, Prof. Nikolay Chervyakov Faculty of Mathematics and Computer Sciences, North-Caucasus Federal University (1, Pushkina ul., Stavropol, 355017 Russia). E-mail: vipetrenko@ncfu.ru. Scopus Author ID 57189512011, ResearcherID A-3196-2017, RSCI SPIN-code 3923-4295, https://orcid.org/0000-0003-4293-7013

**Fariza B. Tebueva,** Dr. Sci. (Phys.-Math.), Associate Professor, Professor, Department of Computational Mathematics and Cybernetics, Prof. Nikolay Chervyakov Faculty of Mathematics and Computer Sciences, North-Caucasus Federal University (1, Pushkina ul., Stavropol, 355017 Russia). E-mail: ftebueva@ncfu.ru. Scopus Author ID 57189512319, ResearcherID H-4548-2017, RSCI SPIN-code 9343-7504, https://orcid.org/0000-0002-7373-4692

**Maxim G. Ogur,** Senior Lecturer, Department of Computational Mathematics and Cybernetics, Prof. Nikolay Chervyakov Faculty of Mathematics and Computer Sciences, North-Caucasus Federal University (1, Pushkina ul., Stavropol, 355017 Russia). E-mail: ogur26@gmail.com. ResearcherID B-1332-2017, RSCI SPIN-code 7180-6971, https://orcid.org/0000-0002-2387-0901

**Gennady I. Linets,** Dr. Sci. (Eng.), Professor, Department of Digital, Robotic Systems and Electronics, Institute of Advanced Engineering, North-Caucasus Federal University (1, Pushkina ul., Stavropol, 355017 Russia). E-mail: kbytw@mail.ru. Scopus Author ID 6506372022, RSCI SPIN-code 1452-6823, https://orcid.org/0000-0002-2279-3887

**Valery P. Mochalov,** Dr. Sci. (Eng.), Professor, Department of Digital, Robotic Systems and Electronics, Institute of Advanced Engineering, North-Caucasus Federal University (1, Pushkina ul., Stavropol, 355017 Russia). E-mail: mochalov.valery2015@yandex.ru. Scopus Author ID 57202300745, RSCI SPIN-code 8695-1648, https://orcid.org/0000-0002-5131-5649