http://dx.doi.org/10.35596/1729-7648-2025-23-4-77-84

УДК 004.896:004.852.4

АРХИТЕКТУРА ПРОТОТИПА СИСТЕМЫ ОБНАРУЖЕНИЯ АНОМАЛИЙ СЕТЕВОГО ТРАФИКА НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ И ВИЗУАЛЬНОЙ АНАЛИТИКИ

С. ВАН[1], И. М. САЙМАНОВ[2], А. В. КАБУЛОВ[2], А. М. ПРУДНИК[1]

[1]Белорусский государственный университет информатики и радиоэлектроники (Минск, Республика Беларусь)
[2]Национальный университет Узбекистана имени Мирзо Улугбека (Ташкент, Республика Узбекистан)

Аннотация. Представлена архитектура прототипа системы для обнаружения аномалий сетевого трафика. Система основана на трехуровневой архитектуре с использованием веб-фреймворка Flask для создания RESTful API. Для обнаружения аномалий применяется алгоритм машинного обучения без учителя Isolation Forest (100 эстиматоров, фактор контаминации 0,05) из библиотеки scikit-learn, осуществляющий обработку данных, предварительно нормализованных с помощью StandardScaler, в одночасовых окнах. Результаты анализа, включающие многоуровневую классификацию серьезности аномалий (с нормализованными оценками в диапазоне 0–1, где значения более 0,8 соответствуют критическому уровню) и обеспечивающие совместимость с SIEM-системами, интерактивно визуализируются посредством Chart.js. Рассмотрены ключевые теоретические и практические вызовы, такие как качество данных, выбор признаков, масштабируемость (алгоритмическая сложность $O(n \log n)$), оптимизация параметров и интерпретируемость результатов.

Ключевые слова: обработка сетевого трафика, обнаружение аномалий, машинное обучение, алгоритм Isolation Forest, визуальная аналитика, архитектура системы, масштабируемость, интерпретируемость, качество данных.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Для цитирования. Архитектура прототипа системы обнаружения аномалий сетевого трафика на основе машинного обучения и визуальной аналитики / С. Ван [и др.] // Доклады БГУИР. 2025. Т. 23, № 4. С. 77–84. http://dx.doi.org/10.35596/1729-7648-2025-23-4-77-84.

ARCHITECTURE OF A PROTOTYPE SYSTEM FOR NETWORK TRAFFIC ANOMALY DETECTION BASED ON MACHINE LEARNING AND VISUAL ANALYTICS

XINGYUE WANG[1], ISLAMBEK M. SAYMANOV[2], ANVAR V. KABULOV[2], ALEKSANDER M. PRUDNIK[1]

[1]Belarusian State University of Informatics and Radioelectronics (Minsk, Republic of Belarus)
[2]National University of Uzbekistan named after Mirzo Ulugbek (Tashkent, Republic of Uzbekistan)

Abstract. The paper presents the architecture of a prototype system for detecting network traffic anomalies. The system is based on a three-tier architecture using the Flask web framework to create a RESTful API. Anomaly detection is implemented using the Isolation Forest unsupervised machine learning algorithm (100 estimators, contamination factor 0.05) from the scikit-learn library, which processes data pre-normalized using StandardScaler in one-hour windows. The analysis results, including a multi-level classification of anomaly severity (with normalized scores in the range of 0–1, where values greater than 0.8 correspond to the critical level) and ensuring compatibility with SIEM systems, are interactively visualized using Chart.js. Key theoretical and practical challenges, such as data quality, feature selection, scalability (algorithmic complexity $O(n \log n)$), parameter optimization, and interpretability of results, are discussed.

Keywords: network traffic processing, anomaly detection, machine learning, Isolation Forest algorithm, visual analytics, system architecture, scalability, interpretability, data quality.

Conflict of interests. The authors declare no conflict of interests.

For citation. Wang X., Saymanov I. M., Kabulov A. V., Prudnik A. M. (2025) Architecture of a Prototype System for Network Traffic Anomaly Detection Based on Machine Learning and Visual Analytics. *Doklady BGUIR*. 23 (4), 77–84. http://dx.doi.org/10.35596/1729-7648-2025-23-4-77-84 (in Russian).

Введение

Рост объемов и сложности сетевого трафика, обусловленный увеличением числа подключенных устройств и цифровых сервисов, создает значительные вызовы для существующих систем обеспечения сетевой безопасности [1]. Современные сети обрабатывают многомерные, изменяющиеся потоки данных, что существенно усложняет задачу эффективного обнаружения аномалий [2]. Традиционные методы, основанные на сигнатурах, демонстрируют высокую эффективность при детектировании известных угроз, однако оказываются недостаточно гибкими для противодействия новым и ранее неизвестным атакам [2, 3]. Это обстоятельство подчеркивает необходимость разработки более адаптивных подходов к анализу сетевого трафика.

В данном контексте перспективным направлением является применение методов машинного обучения, способных выявлять неявные отклонения в сетевом поведении без необходимости использования наборов предварительно размеченных данных. Однако внедрение машинного обучения сопряжено с рядом технических и эксплуатационных трудностей [4]. В частности, методы неконтролируемого обучения, актуальные в условиях отсутствия размеченных выборок, могут генерировать значительное количество ложных срабатываний, а чисто алгоритмические системы часто лишены достаточной интерпретируемости для специалистов по безопасности [4, 5].

Цель исследования – разработка концептуальных и архитектурных основ системы обнаружения аномалий сетевого трафика, которая объединяет машинное обучение без учителя с интерактивными средствами визуальной аналитики. Такой гибридный подход призван решить проблемы масштабируемости и интерпретируемости, позволяя администраторам сетей не только обнаруживать, но и эффективно контекстуализировать аномалии. Предлагаемый подход фокусируется на теоретических и архитектурных аспектах, закладывая фундамент для создания надежных систем обнаружения аномалий. Ключевыми вызовами, рассматриваемыми в статье, являются выбор и оптимизация алгоритмов, предварительная обработка данных в больших масштабах, обеспечение масштабируемости для анализа в реальном времени и повышение интерпретируемости результатов.

Методика проведения эксперимента

В основе предлагаемого решения лежит модульная архитектура, нацеленная на решение задач, обозначенных во введении, и обеспечивающая масштабируемость и адаптивность системы. Система включает три основных компонента: модуль ввода и предварительной обработки данных, уровень обработки и анализа, а также модуль генерации выходных данных и управления оповещениями. Прототип системы реализован с использованием трехуровневой архитектуры, включающей уровни сбора данных, анализа и представления, с применением веб-фреймворка Flask. Компоненты взаимодействуют через RESTful API, что обеспечивает модульность и разделение задач.

Уровень сбора данных отвечает за получение сетевого трафика. Для первоначальной валидации и тестирования системы используется модуль генерации синтетических данных, включающий такие характеристики, как временные паттерны и распределение протоколов. При операционном развертывании предполагается работа с реальным трафиком, что сопряжено с проблемами качества и избыточного объема данных.

Уровень анализа включает ядро системы, где происходит обнаружение аномалий. Ключевым элементом является конвейер обработки, который выполняет извлечение признаков и применение алгоритмов машинного обучения. В прототипе этот уровень использует алгоритм Isolation Forest для детектирования аномалий, а сохраняемость данных обеспечивается SQLite и SQLAlchemy ORM.

Уровень представления отвечает за визуализацию результатов и управление оповещениями. В прототипе используется веб-интерфейс на основе Flask и Bootstrap с отрисовкой визуализаций с помощью Chart.js.

Такая архитектура (рис. 1) спроектирована для поддержки как пакетной, так и потоковой обработки данных, удовлетворяя требованиям крупномасштабного мониторинга.

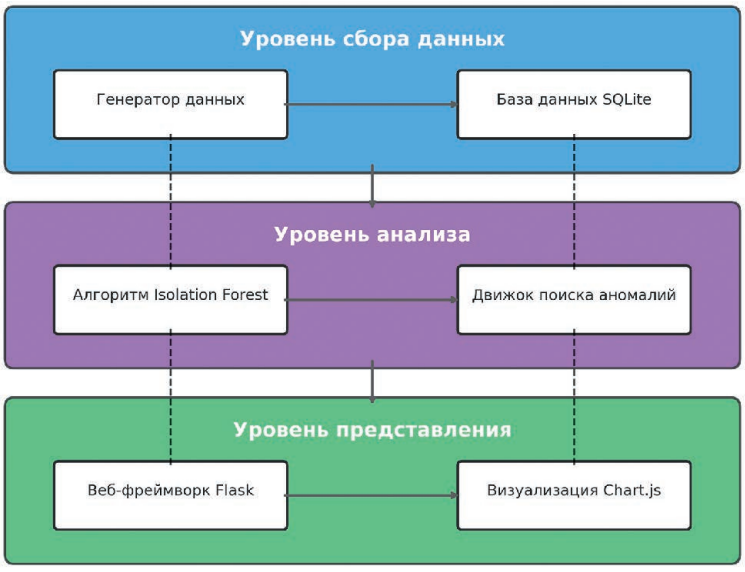

Рис. 1. Трехуровневая архитектура прототипа обнаружения сетевых аномалий
Fig. 1. Three-tier architecture of the network anomaly detection prototype

Качество входных данных является критическим фактором для эффективности системы обнаружения аномалий. В сетях данные часто гетерогенны, с неконсистентной выборкой и пропущенными значениями, что может снизить точность детектирования.

На этапе прототипирования используется модуль генерации синтетических данных, создающий записи с атрибутами, такими как временная метка, IP-адреса источника и назначения, протокол, количество переданных байт и пакетов, длительность соединения и номера портов. Этот подход соответствует практикам, описанным в [6], и позволяет моделировать различные сетевые сценарии, включая временные паттерны и распределение протоколов [7].

Для решения проблем с качеством данных и их разнородностью предлагается адаптивный конвейер предварительной обработки. Он включает нормализацию форматов данных, приведение их к единому виду с сохранением ключевых признаков. Извлечение признаков использует методы снижения размерности и z-нормализацию (z-score normalization) для унификации разнообразных метрик. В прототипе для нормализации признаков перед подачей в алгоритм Isolation Forest применяется StandardScaler из библиотеки scikit-learn, что позволяет устранить проблемы, связанные с различными масштабами признаков. Обработка данных осуществляется в одночасовых окнах, что обеспечивает баланс между эффективностью и сохранением контекста. Система спроектирована для поддержки как пакетной, так и потоковой обработки данных через гибкие интерфейсы, что важно для задач мониторинга в реальном времени [8].

Методы машинного обучения превосходят традиционные сигнатурные подходы в задачах обнаружения аномалий сетевого трафика, особенно в идентификации угроз «нулевого дня» и эволюционирующих атак. Обучение без учителя является предпочтительным ввиду отсутствия больших объемов размеченных данных в операционных сетях, где векторы атак быстро меняются. Такие методы обнаруживают аномалии как отклонения от установленной «нормы» поведения, обеспечивая масштабируемость для новых угроз.

В качестве основного алгоритма обнаружения аномалий был выбран Isolation Forest, который демонстрирует высокую эффективность при работе с многомерными данными сетевого трафика [9]. Он изолирует разреженные аномалии со средней временной сложностью $O(n \log n)$, что позволяет осуществлять обработку в реальном времени [10]. В прототипе используется реа-

лизация Isolation Forest из библиотеки scikit-learn со 100 деревьями (эстиматорами) и фактором контаминации 0,05, параметрами, оптимизированными для корпоративных сетевых сред [11]. Алгоритм вычисляет оценки аномальности на основе длин путей в случайно построенных бинарных деревьях; эти оценки затем трансформируются в диапазон [0, 1] для классификации серьезности.

Для анализа используются такие признаки, как количество переданных байт, количество пакетов, длительность соединения и номера портов. В целом в сетевом трафике может быть идентифицировано более 40 признаков [6], и оптимальный их набор может варьироваться в зависимости от типа атаки, что требует адаптивных механизмов выбора признаков.

Несмотря на преимущества, использование Isolation Forest сопряжено с вызовами. Динамический характер трафика усложняет адаптацию модели и настройку пороговых значений. Оптимизация параметров в алгоритмах изоляции может влиять на точность до 15 % [12], что усугубляется отсутствием размеченных данных. Кроме того, «непрозрачность» выходных данных (эффект «черного ящика») затрудняет интерпретацию результатов и может увеличивать время расследования инцидентов до 73 % [13]. Для решения этих проблем предлагаемый фреймворк использует гибридный подход, дополняя машинное обучение средствами визуализации, что способствует принятию решений.

Эффективность системы обнаружения аномалий во многом зависит от того, насколько понятно и оперативно результаты анализа доводятся до сведения специалистов по безопасности.

Модуль вывода использует многоуровневую систему классификации серьезности сетевых аномалий. Генерируются нормализованные оценки в диапазоне от 0 до 1, основанные на степени отклонения от базового профиля трафика. В прототипе аномалии классифицируются на критические (>0,8), высокие (>0,7), средние (>0,6) и низкие (≤0,6) уровни серьезности. Такой подход соответствует практикам визуализации и выводам о том, что согласованные метрики помогают в принятии решений [5]. Аномалии с оценкой более 0,8 считаются критическими и требуют приоритетного рассмотрения, что согласуется с исследованиями, показывающими, что градуированные оповещения снижают когнитивную нагрузку на аналитиков [14].

Система использует бессостоятельную (stateless) архитектуру, сохраняя результаты в формате, совместимом с системами управления информацией о безопасности и событиями (SIEM), что обеспечивает интероперабельность. При генерации оповещений применяются методы корреляции и агрегации для снижения избыточности при сохранении временных связей между событиями. Структуры данных, не зависящие от протокола, обеспечивают интеграцию с различными аналитическими платформами, сохраняя контекст угрозы.

Для улучшения интерпретируемости результатов и облегчения анализа человеком фреймворк визуальной аналитики преобразует сложные данные сетевого трафика в интерактивные визуализации с использованием Chart.js. Это позволяет администраторам контекстуализировать аномалии в операционной среде [15].

Анализ временных рядов трафика представлен линейными графиками, отображающими объем трафика за настраиваемые периоды времени (от одного часа до семи дней), что позволяет выявлять аномалии, такие как всплески, связанные с DDoS-атакам (рис. 2).

Распределение протоколов визуализируется с помощью кольцевых диаграмм, показывающих частотное распределение протоколов, что способствует быстрому обнаружению необычных паттернов, например, туннелирования или скрытых каналов. Серьезность аномалий изображается с помощью круговых диаграмм, классифицирующих обнаруженные аномалии по уровням серьезности с использованием цветовой схемы (от красного для критического до синего для низкого уровня), что позволяет быстро проводить первоначальную оценку угроз.

Как показано на рис. 2, визуализация нормальных и аномальных паттернов в пространстве признаков помогает в принятии решений. Границы принятия решений алгоритма Isolation Forest выделяют области, где аномалии отклоняются от нормального распределения трафика. Например, атаки большого объема могут кластеризоваться в верхней правой части, тогда как низкоинтенсивные атаки – в левой нижней, демонстрируя различные характеристики трафика. Оттенки фона могут представлять различные оценки аномальности, где более темные области указывают на зоны, более вероятно классифицируемые как нормальные.

Рис. 2. Интерактивные визуализации трафика, протоколов и серьезности аномалий
Fig. 2. Interactive visualizations of traffic, protocol, and anomaly severity

Результаты исследований и их обсуждение

Представленный концептуальный фреймворк и архитектура прототипа демонстрируют жизнеспособный подход к обнаружению аномалий сетевого трафика путем интеграции машинного обучения без учителя с визуальной аналитикой. Прототип, реализованный на базе трехуровневой архитектуры с использованием Flask, SQLite и Chart.js, позволяет эффективно обрабатывать и визуализировать сетевые данные для выявления подозрительной активности.

Применение алгоритма Isolation Forest с временной сложностью $O(n \log n)$ обеспечивает теоретическую возможность обработки значительных объемов данных (до $\sim 10^5$ записей в минуту на стандартном оборудовании при обработке данных в часовых окнах), что является важным для систем, работающих в условиях интенсивного трафика. Хотя текущая реализация прототипа работает с меньшими пакетами синтетических данных, заложенная архитектура и выбор алгоритма ориентированы на масштабируемость.

Многоуровневая классификация серьезности аномалий (0–1, с порогом >0,8 для критических) в сочетании с интерактивными визуализациями (временные ряды, распределение протоколов, диаграммы серьезности) значительно повышает интерпретируемость результатов. Это позволяет сетевым администраторам не только идентифицировать аномалии, но и быстрее понимать их контекст, что критически важно для оперативного реагирования [15]. Визуализация нормальных и аномальных паттернов в пространстве признаков (рис. 3) помогает демистифицировать решения модели.

Несмотря на потенциальные преимущества, реализация и эксплуатация таких систем сталкиваются с рядом существенных вызовов, влияющих на точность и производительность. Гетерогенность трафика, наличие пропущенных значений и несогласованность выборок в реальных сетях остаются серьезным препятствием. Необходимы надежные механизмы предварительной обработки для нормализации данных и сохранения индикаторов аномалий.

В сетевом трафике может быть более 40 признаков [6], и оптимальный их набор варьируется в зависимости от типа атаки. Это требует разработки адаптивных механизмов выбора признаков для повышения точности обнаружения различных угроз.

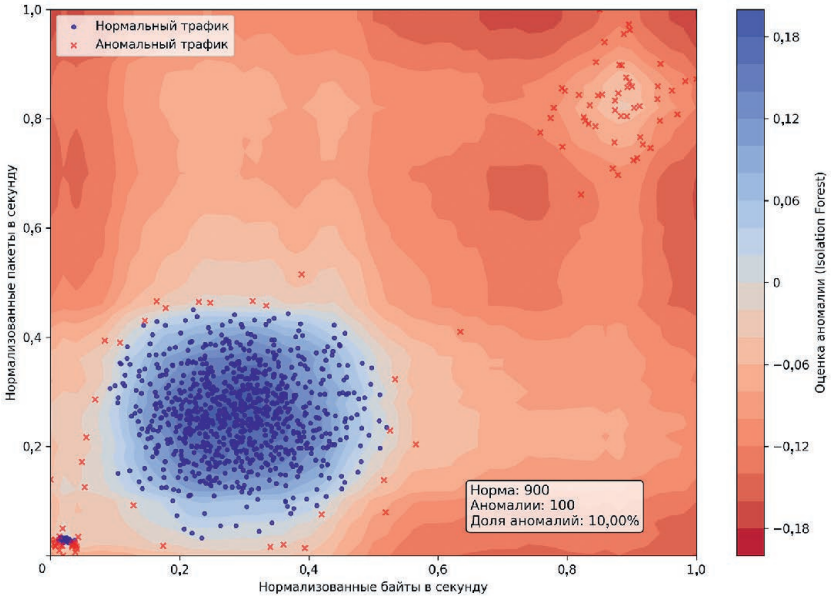

Рис. 3. Концептуальная визуализация нормальных и аномальных паттернов трафика
Fig. 3. Conceptual visualization of normal versus anomalous traffic patterns

Хотя алгоритм Isolation Forest эффективен, реальные сетевые потоки, превышающие 100 Гбит/с [16], могут создавать значительную нагрузку на ресурсы. Даже оптимизированные системы могут испытывать трудности во время пиковых нагрузок, что требует эффективных алгоритмов и стратегий управления ресурсами. Потребление памяти в текущем прототипе масштабируется линейно с объемом данных, что следует учитывать при обработке больших объемов трафика.

Настройка параметров алгоритмов, таких как Isolation Forest (например, количество деревьев, фактор контаминации), существенно влияет на точность (до 15 % [12]). Эта задача усложняется динамическим характером сетевого трафика и отсутствием размеченных данных для валидации. Для эффективной работы системы требуется механизм адаптивной настройки.

Несмотря на усилия по визуализации, модели машинного обучения, включая Isolation Forest, могут оставаться «черными ящиками». Это может увеличивать время, необходимое для расследования оповещений (до 73 % [13]), и снижать доверие к системе. Необходимы дальнейшие исследования в области объяснимого искусственного интеллекта (XAI) для сетевой безопасности.

Текущий прототип имеет некоторые ограничения, такие как требование наличия минимум десяти записей для эффективного обнаружения аномалий, проблемы с отображением при высокой кардинальности данных в визуализациях и редкие ошибки конечных точек API, указывающие на необходимость улучшения механизмов обновления.

Заключение

1. Представлены теоретический фреймворк и архитектурные решения для прототипа системы обнаружения аномалий в сетевом трафике, который интегрирует машинное обучение без учителя с интерактивными средствами визуальной аналитики. Предложенный модульный подход, реализованный с использованием веб-фреймворка Flask, алгоритма Isolation Forest и библиотеки визуализации Chart.js, направлен на преодоление ограничений традиционных сигнатурных методов за счет алгоритмической адаптивности и гибкости архитектуры.

2. Система демонстрирует потенциал для эффективного выявления аномальных паттернов благодаря временной сложности $O(n \log n)$ основного алгоритма и многоуровневой системе классификации серьезности угроз. Интеграция визуальной аналитики решает одну из ключевых проблем машинного обучения – интерпретируемость результатов, предоставляя администраторам интуитивно понятные инструменты для анализа и принятия решений. Однако, как показал анализ, на пути к созданию полнофункциональных и надежных систем стоят существенные вызовы. К ним относятся обеспечение высокого качества входных данных, динамическая оптимизация

набора признаков, достижение масштабируемости для обработки высокоскоростных потоков, точная настройка параметров алгоритмов в условиях отсутствия размеченных данных и дальнейшее повышение интерпретируемости моделей. Текущий прототип также имеет ограничения, связанные с «холодным стартом», линейным масштабированием использования памяти и необходимостью улучшения стабильности API.

3. Дальнейшая работа будет сосредоточена на эмпирической валидации предложенного подхода с использованием реальных операционных данных, доработке и уточнении алгоритмов на основе метрик производительности, а также на исследовании путей решения идентифицированных проблем. Это включает разработку автоматизированных механизмов планирования анализа, повышение надежности API, внедрение обратной связи для улучшения модели и возможное использование WebGL для ускорения визуализаций.

4. Представленная разработка закладывает теоретическую и архитектурную основы для создания более совершенных систем обнаружения аномалий, способных адаптироваться к постоянно меняющемуся ландшафту сетевых угроз. Углубленное изучение обозначенных вызовов будет способствовать дальнейшему прогрессу в области применения машинного обучения без учителя для задач сетевой безопасности.

Список литературы / References

1. Sommer R., Paxson V. (2010) Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *IEEE Symposium on Security and Privacy*. 305–316.
2. García-Teodoro P., Díaz-Verdejo J., Maciá-Fernández G., Vázquez E. (2009) Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges. *Computers & Security*. 28 (1–2), 18–28.
3. Buczak A. L., Guven E. (2016) A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*. 18 (2), 1153–1176.
4. Bhuyan M. H., Bhattacharyya D. K., Kalita J. K. (2014) Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys & Tutorials*. 16 (1), 303–336.
5. D'Amico A., Whitley K. (2008) The Real Work of Computer Network Defense Analysts. *VizSEC 2007 (Mathematics and Visualization)*. 19–37.
6. Ring M., Wunderlich S., Grüdl D., Landes D., Hotho A. (2017) Flow-Based Benchmark Data Sets for Intrusion Detection. *Proc. of the 16th European Conf. on Cyber Warfare and Security*. 361–369.
7. Sharafaldin I., Lashkari A. H., Ghorbani A. A. (2018) Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. *Proc. of the 4th International Conf. on Information Systems Security and Privacy*. 108–116.
8. Cordero C. G., Vasilomanolakis E., Milanov N., Koch C., Hausheer D., Mühlhäuser M. (2015) An Overview of the Botnet Simulation Framework. *IEEE Conf. on Communications and Network Security (CNS)*. 739–740.
9. Liu F. T., Ting K. M., Zhou Z. H. (2012) Isolation-Based Anomaly Detection. *ACM Transactions on Knowledge Discovery from Data*. 6 (1), 1–39.
10. Ding Z., Fei M. (2013) An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data Using Sliding Window. *IFAC Proceedings Volumes*. 46 (20), 12–17.
11. Apruzzese G., Colajanni M., Ferretti L., Guido A., Marchetti M. (2018) On the Effectiveness of Machine and Deep Learning for Cyber Security. *2018 10th International Conference on Cyber Conflict (CyCon)*. 371–390. IEEE.
12. Wang H., Bah M. J., Hammad M. (2019) Progress in Outlier Detection Techniques: A Survey. *IEEE Access*. 7, 107964–108000.
13. Ribeiro M. T., Singh S., Guestrin C. (2016) "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 1135–1144.
14. Perdisci R., Ariu D., Fogla P., Giacinto G., Lee W. (2009) McPAD: A Multiple Classifier System for Accurate Payload-Based Anomaly Detection. *Computer Networks*. 53 (6), 864–881.
15. Staheli D., Yu T., Crouser R. J., Damodaran S., Nam K., O'Gwynn D., et al. (2014) Visualization Evaluation for Cyber Security: Trends and Future Directions. *VizSec '14: Proceedings of the Eleventh Workshop on Visualization for Cyber Security*. 49–56.
16. Perdisci R., Lee W., Feamster N. (2010) Behavioral Clustering of HTTP-Based Malware and Signature Generation Using Malicious Network Traces. *NSDI'10: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation.* 12.

Поступила 29.05.2025
Received: 29 May 2025

Принята в печать 18.06.2025
Accepted: 18 June 2025

Вклад авторов

Ван С. реализовал программный прототип системы, разработал модули генерации данных и базовой визуализации.

Сайманов И. М. определил методику предварительной обработки данных, сформулировал критерии отбора признаков для анализа.

Кабулов А. В. обосновал выбор методов машинного обучения, провел экспертизу теоретических положений.

Прудник А. М. разработал архитектуру системы, определил требования к ее масштабируемости и интеграции.

Authors' contribution

Wang X. implemented the system's software prototype, developed data generation and basic visualization modules.

Saymanov I. M. defined the data preprocessing methodology, formulated feature selection criteria for analysis.

Kabulov A. V. justified the selection of machine learning methods, validated the theoretical foundations.

Prudnik A. M. designed the system architecture, defined the requirements for its scalability and integration.

Сведения об авторах

Ван С., магистрант каф. защиты информации и эргономики, Белорусский государственный университет информатики и радиоэлектроники (БГУИР)

Сайманов И. М., канд. техн. наук, доц., доц. каф. информационной безопасности, Национальный университет Узбекистана имени Мирзо Улугбека (НУУ)

Кабулов А. В., д-р техн. наук, проф., проф. каф. информационной безопасности, НУУ

Прудник А. М., канд. техн. наук, доц., доц. каф. инженерной психологии и эргономики, БГУИР

Information about the authors

Wang X., Master's Student at the Information Security Department, Belarusian State University of Informatics and Radioelectronics (BSUIR)

Saymanov I. M., Cand. Sci. (Tech.), Associate Professor, Associate Professor at the Information Security Department, National University of Uzbekistan named after Mirzo Ulugbek (NUU)

Kabulov A. V., Dr. Sci. (Tech.), Professor, Professor at the Information Security Department, NUU

Prudnik A. M., Cand. Sci. (Tech.), Associate Professor, Associate Professor at the Engineering Psychology and Ergonomics Department, BSUIR

Адрес для корреспонденции

220013, Республика Беларусь,
Минск, ул. П. Бровки, 6
Белорусский государственный университет
информатики и радиоэлектроники
Тел.: +375 17 293-85-24
E-mail: aleksander.prudnik@bsuir.by
Прудник Александр Михайлович

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovki St., 6
Belarusian State University
of Informatics and Radioelectronics
Tel.: +375 17 293-85-24
E-mail: aleksander.prudnik@bsuir.by
Prudnik Aleksander Mikhailovich