

Фадеев Василий Иванович

магистрант, Институт информационных систем и
инженерно-компьютерных технологий
АНО ВО «Российский новый университет»
Fadееv Vassiliy Ivanovich
Master's Student, Institute of Information Systems and
Engineering Computer Technologies
Autonomous Non-profit Organization
of Higher Education "Russian New University"

СРАВНИТЕЛЬНЫЙ АНАЛИЗ КОНТРОЛИРУЕМЫХ И НЕКОНТРОЛИРУЕМЫХ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОБНАРУЖЕНИЯ СЕТЕВЫХ АТАК COMPARATIVE ANALYSIS OF SUPERVISED AND UNSUPERVISED MACHINE LEARNING METHODS FOR NETWORK ATTACK DETECTION

Аннотация. Сравниваются Random Forest, LSTM, 1D-CNN и автоэнкодер на наборе CSE-CIC-IDS2018. Макро-F1 контролируемых моделей составил 0,9693–0,9782; Random Forest показал ROC-AUC 0,9925 и задержку 0,008 мс на образце. Для автоэнкодера макро-F1 составил 0,8243, полнота по атакам – 0,74. Обсуждается совместное применение классификатора и детектора аномалий; выявление zero-day-атак требует отдельной проверки.

Abstract. Four attack-detection methods are compared on CSE-CIC-IDS2018: Random Forest, LSTM, 1D-CNN, and an autoencoder trained on benign traffic. Supervised models achieved macro-F1 of 0.9693–0.9782; Random Forest reached ROC-AUC 0.9925 at 0.008 ms per sample. The autoencoder achieved macro-F1 0.8243 and attack recall 0.74. Combined classification and anomaly detection are discussed; zero-day detection requires separate validation.

Ключевые слова: Обнаружение вторжений; сетевая безопасность; машинное обучение; Random Forest; автоэнкодер; SHAP.

Keywords: Intrusion detection; network security; machine learning; Random Forest; autoencoder; SHAP.

Введение

Развитие цифровой инфраструктуры сопровождается усложнением сетевого взаимодействия и расширением спектра угроз информационной безопасности. Для выявления атак применяются сигнатурные средства, межсетевые экраны и системы обнаружения вторжений, однако такие решения нуждаются в обновлении правил и могут пропускать ранее не наблюдавшиеся варианты атак. Методы искусственного интеллекта позволяют анализировать большие массивы сетевых признаков, выявлять закономерности поведения и поддерживать работу специалистов центров мониторинга безопасности (SOC) [1, 2].

Сравнивать модели обнаружения вторжений только по общей точности недостаточно. В условиях дисбаланса классов метрика Accuracy может скрывать низкую полноту обнаружения редких атак. Практическую значимость также имеют макроусреднённые Precision, Recall и F1-score, ROC-AUC, задержка инференса, стоимость обучения и объяснимость результатов. Для организаций важны и процессы управления рисками на протяжении жизненного цикла моделей, в том числе в контексте систем менеджмента информационной безопасности и искусственного интеллекта [3, 4].

В проведенном исследовании на наборе CSE-CIC-IDS2018 были реализованы и сопоставлены модели Random Forest, LSTM и 1D-CNN. Все три модели показали F1-score (macro) не ниже 0,9693, при этом Random Forest обеспечил более низкую задержку инференса, а SHAP использовался для анализа влияния признаков на предсказания. В дополнительном



эксперименте была разработана модель автоэнкодера, обучаемая только на нормальном трафике. Такой подход позволяет оценивать отклонение наблюдений от нормального поведения и представляет интерес при ограниченном количестве размеченных атак [5].

Цель настоящей работы – обобщить и сопоставить результаты контролируемых моделей и автоэнкодера для обнаружения сетевых атак, выделить сильные стороны и ограничения каждого подхода и сформулировать рекомендации по их совместному применению в корпоративной инфраструктуре. Практическая особенность работы состоит в совместном анализе качества классификации, скорости обработки и интерпретируемости, а также в явном учёте различий экспериментальных протоколов.

1. Материалы и методы исследования

1.1. Набор данных и предварительная обработка

В качестве экспериментального источника использован CSE-CIC-IDS2018, содержащий сетевой трафик с нормальной активностью и несколькими сценариями атак, включая DDoS, brute force, сканирование портов и веб-атаки [6, 7]. В описываемых исследованиях использовалась версия объёмом 9 625 148 записей и 79 признаков [8]. После удаления записей с пропущенными значениями и константных признаков в эксперименте с контролируемым обучением осталось 67 информативных признаков. Числовые показатели нормализовались методом MinMaxScaler до диапазона [0; 1].

Для контролируемых моделей данные перемешивались и разделялись в пропорции 80/20: 7 700 118 записей отводилось на обучение, 1 925 030 – на тестирование. Для LSTM и 1D-CNN дополнительно формировались временные окна длиной 10 шагов. Для снижения влияния дисбаланса классов и с учётом ограничений оперативной памяти экспериментального стенда применялся метод Random Under-Sampling только к обучающей выборке. В результате получено по 2 198 705 образцов каждого класса, всего 4 397 410 образцов. Тестовая выборка не балансировалась.

Для автоэнкодера использовалась другая схема подготовки. После очистки данных применялись SimpleImputer с медианным заполнением и MinMaxScaler. На вход подавались семь признаков, выбранных на основании SHAP-анализа модели Random Forest: dst_port, fwd_seg_size_min, init_fwd_win_byts, fwd_pkt_len_mean, subflow_fwd_byts, rst_flag_cnt и ece_flag_cnt. Обучающая и валидационная выборки содержали только нормальный трафик (Benign); атакующие записи использовались только на этапе итоговой оценки.

1.2. Контролируемые модели

Random Forest представлял собой ансамбль из 100 деревьев решений. LSTM имела один скрытый слой из 128 нейронов и предназначалась для учёта временных зависимостей. Архитектура 1D-CNN включала два свёрточных слоя с 64 и 32 фильтрами, ядро размером 3 и слой глобального пулинга. Эти модели обучались классификации с использованием размеченных примеров нормального трафика и атак.

1.3. Автоэнкодер и правило обнаружения аномалий

Автоэнкодер реализован как полносвязная нейронная сеть с архитектурой «бутылочного горлышка»: 7–32–16–4–16–32–7 (показано на рисунке 1). Кодировщик сжимает входной вектор из семи признаков до латентного представления размерности 4; декодировщик восстанавливает исходные значения. В скрытых слоях используется функция активации ReLU, а выходной слой имеет линейную активацию. Обучение проводилось с оптимизатором Adam и функцией потерь MSELoss, минимизирующей среднеквадратичную ошибку между исходным и восстановленным векторами.



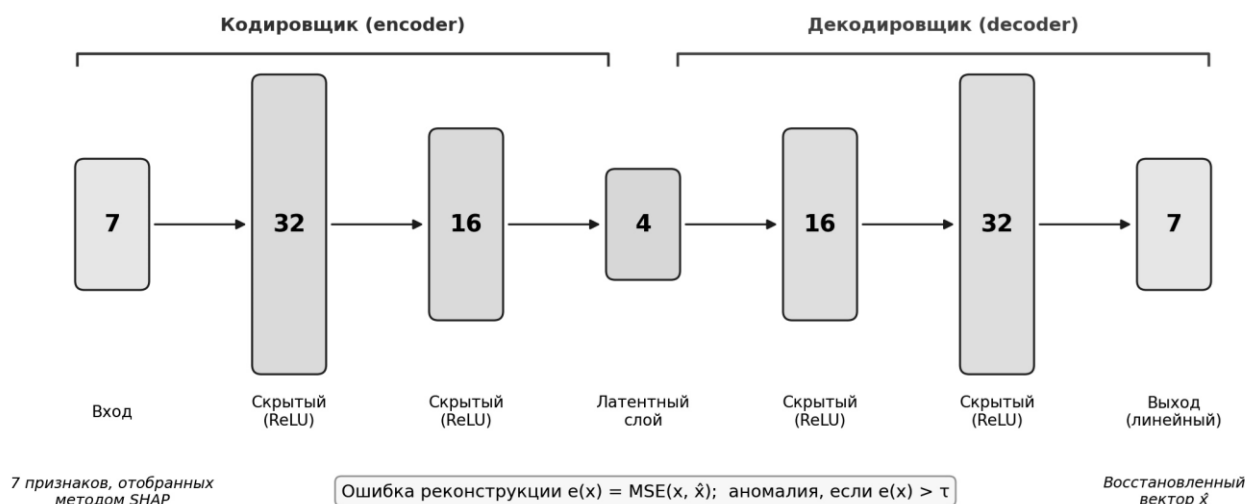


Рисунок 1. Архитектура автоэнкодера (7–32–16–4–16–32–7)

Для каждого наблюдения рассчитывалась ошибка реконструкции. Порог аномалии калибровался по распределению ошибок на валидационной выборке нормального трафика. Рассматривались пороги, соответствующие 95-му и 99-му перцентилям; наилучший из результатов был получен при 95-м перцентиле. Если ошибка реконструкции превышает установленный порог, наблюдение помечается как потенциальная аномалия. Такой выход следует трактовать как сигнал для дополнительной проверки аналитиком, а не как однозначное доказательство злонамеренной активности.

1.4. Метрики и границы сопоставления

Для контролируемых моделей оценивались метрики Accuracy, макроусреднённые Precision, Recall и F1-score, ROC-AUC, время обучения и время инференса на один образец. Для автоэнкодера оценены Precision, Recall и F1-score по классам, макроусреднённые значения, ROC-AUC и задержка инференса.

Сопоставление имеет ориентировочный характер. Во-первых, контролируемые модели использовали 67 признаков, а автоэнкодер только 7. Во-вторых, различались стратегии обучения: контролируемые модели обучались на сбалансированных размеченных данных, автоэнкодер только на нормальном трафике. Поэтому сравнение полезно для выбора архитектурного подхода, но не заменяет контролируемого эксперимента, в котором все модели запускаются на идентичном разбиении данных и аппаратуре.

2. Результаты исследования

2.1. Качество и производительность контролируемых моделей

Результаты базового эксперимента приведены в таблицах 1 и 2. Наибольшее значение F1-score (macro) показала LSTM – 0,9782, однако преимущество относительно Random Forest (0,9779) незначительно. Наибольший ROC-AUC получен у Random Forest (0,9925). Модель Random Forest также продемонстрировала наименьшую задержку инференса: 0,008 мс на образец по сравнению с 0,107 мс у 1D-CNN и 0,121 мс у LSTM. Время обучения Random Forest составило 1 405,04 с, что существенно меньше приведённых для нейросетевых моделей значений.

Таблица 1

Время обучения и задержка инференса контролируемых моделей

Модель	Время обучения, с	Инференс, мс
Random Forest	1 405,04	0,008
LSTM	8 381,64	0,121
1D-CNN	7 495,59	0,107



Таблица 2

Метрики качества контролируемых моделей на CSE-CIC-IDS2018

Модель	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)	ROC-AUC
Random Forest	0,9822	0,9854	0,9711	0,9779	0,9925
LSTM	0,9826	0,9875	0,9699	0,9782	0,9896
1D-CNN	0,9755	0,9807	0,9593	0,9693	0,9833

Полученные значения показывают, что высокая классификационная результативность сама по себе не определяет оптимальный выбор модели. При близких макро-F1-результатах Random Forest имеет преимущество по задержке и ROC-AUC, что делает его обоснованным кандидатом для первичной фильтрации трафика при жёстких требованиях к быстродействию. При этом измеренная задержка относится к условиям исходного эксперимента; переносить её на другую аппаратную конфигурацию или непосредственно на пропускную способность промышленной сети без отдельного тестирования нельзя.

2.2. Интерпретация признаков с использованием SHAP

Для повышения интерпретируемости Random Forest использовался метод SHAP (SHapley Additive exPlanations), позволяющий оценивать вклад признаков в выход модели [9]. Сводная диаграмма из исходного исследования представлена на рисунке 2. Среди признаков с наибольшим влиянием выделяются порт назначения (dst port), минимальный размер сегмента прямого потока (fwd_seg_size_min), начальный размер окна прямого потока (init_fwd_win_byts), средняя длина пакета прямого потока (fwd_pkt_len_mean) и объём данных подпотока прямого направления (subflow_fwd_byts). В диаграмме также заметны вклады счетчиков флагов rst_flag_cnt и ece_flag_cnt.

Направление вклада SHAP зависит от значения признака и поведения конкретной обученной модели. В исходном анализе низкие значения dst_port связывались с повышением оценки атаки, а высокие значения fwd_seg_size_min и счётчиков RST/ECE – с положительным вкладом в решение о наличии атаки. Эти наблюдения помогают формировать объяснения для SOC, но не означают, что отдельный порт или TCP-флаг сам по себе доказывает атаку. Интерпретация должна рассматриваться совместно с контекстом соединения, корреляциями признаков и результатами расследования.

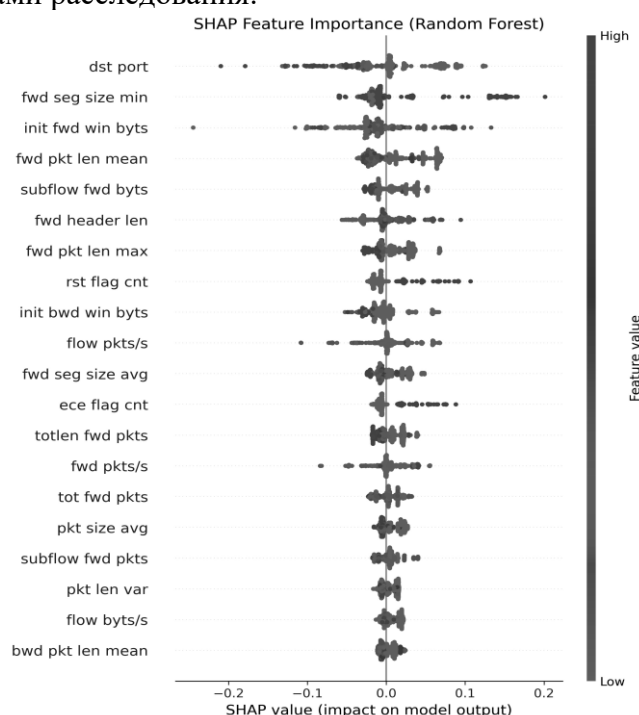


Рисунок 2. Сводная диаграмма важности признаков SHAP для модели Random Forest



2.3. Результаты автоэнкодера

Для автоэнкодера наилучшие результаты в представленном эксперименте получены при пороге ошибки реконструкции, соответствующем 95-му перцентилю ошибок нормального трафика. В таблице 3 приведены показатели по классам. Полнота для класса Benign составила 0,95, а для Attack – 0,74. Таким образом, при выбранном пороге сохранялась способность выделять значительную долю атак, однако около 26% атакующих записей не были обнаружены. Макроусреднённые Precision и Recall составили 0,8342 и 0,8452; макро-F1 – 0,8243, ROC-AUC – 0,8775.

Таблица 3

Метрики автоэнкодера при пороге 95-го перцентиля

Класс	Precision	Recall	F1-score	Число образцов
Benign (0)	0,71	0,95	0,81	1 856 767
Attack (1)	0,96	0,74	0,83	2 748 235

Повышение порога до 99-го перцентиля в проведённых экспериментах сопровождалось снижением F1-score до диапазона 0,55–0,70, это подтверждает влияние калибровки порога на компромисс между чувствительностью детектора и частотой ложных срабатываний. Конкретное же рабочее значение порога нужно подбирать с учётом допустимого уровня ложных тревог, стоимости пропуска атаки и характера трафика предприятия, а после развёртывания ещё и регулярно проверять из-за возможного изменения распределения данных.

2.4. Сводное сравнение четырёх подходов

В таблице 4 объединены результаты двух проведенных исследований. Она иллюстрирует принципиальный компромисс между качеством классификации известных классов и обучением без разметки атак.

Таблица 4

Сопоставление результатов контролируемых моделей и автоэнкодера

Модель	Precision (macro)	Recall (macro)	F1-score (macro)	ROC-AUC	Инференс, мс
Автоэнкодер	0,8342	0,8452	0,8243	0,8775	≈ 0,3 (оценка)
Random Forest	0,9854	0,9711	0,9779	0,9925	0,008
LSTM	0,9875	0,9699	0,9782	0,9896	0,121
1D-CNN	0,9807	0,9593	0,9693	0,9833	0,107

Контролируемые модели демонстрируют более высокие макро-F1 и ROC-AUC в представленных результатах. Это ожидаемо для эксперимента, в котором модель получает размеченные примеры обоих классов и может обучить разделяющую границу. Автоэнкодер решает другую задачу: он приближает восстановление нормального трафика, после чего превышение ошибки реконструкции используется как индикатор отклонения. Поэтому меньшие значения его метрик не позволяют заключить, что подход неэффективен; они отражают различие целей обучения и информационного состава входных признаков.

3. Обсуждение результатов

Полученные результаты в целом подтверждают, что выбор модели зависит от конкретной задачи. Там, где нужна быстрая первичная фильтрация, небольшая задержка и понятное объяснение решения, имеет смысл смотреть в сторону Random Forest вместе с SHAP. LSTM и 1D-CNN лучше подходят для анализа последовательностей и более глубокой обработки событий, если мы можем позволить себе большую вычислительную задержку.



Автоэнкодер же логично использовать как дополнительный индикатор отклонения от нормального поведения, особенно когда размеченных атак не хватает или они не покрывают все возможные случаи [1, 5]. При этом я бы не стал считать доказанной его способность находить неизвестные атаки только потому, что он обучался на нормальном трафике. Здесь нужны дополнительные проверки.

Есть и ещё одно ограничение – разное число признаков у моделей. Семь отобранных признаков упрощают автоэнкодер и могут снизить вычислительные затраты, но есть риск, что какие-то типы атак они просто не описывают. Поэтому дальше стоит проверить несколько вариантов: одинаковые семь признаков для всех моделей, полный набор из 67 признаков для автоэнкодера, а также отбор признаков под конкретные классы угроз.

SHAP, по сути, показывает, какие входные признаки повлияли на выход классификатора, но это ещё не значит доказательства причинно-следственных связей. Для практического SOC-интерфейса, как мне кажется, стоит показывать класс или оценку аномальности, и признаки, которые сильнее всего повлияли на решение. После этого событие всё равно должно уходить на подтверждение и проверку контекста. Такой подход снижает риск, что решение автоматически примут из-за одного эвристического признака.

4. Практические рекомендации по внедрению

На основании проведенного сопоставления предлагается многоуровневая схема обнаружения атак. На первом уровне Random Forest может выполнять низколатентную классификацию сетевых потоков; SHAP используется для объяснения решений и анализа тенденций. На втором уровне LSTM или 1D-CNN могут выполнять более глубокий анализ последовательностей и связанных событий. Параллельно автоэнкодер рассчитывает ошибку реконструкции нормального поведения и помечает нетипичные соединения, в том числе те, для которых контролируемый классификатор демонстрирует низкую уверенность или которые не соответствуют известным шаблонам. Фактическое объединение выходов требует отдельной калибровки и оценки на едином наборе данных.

До выхода в промышленную эксплуатацию стоит заранее зафиксировать, что мы считаем приемлемым результатом. Сюда входят полнота по классам критичных атак, доля ложных срабатываний на нормальном трафике, максимальная задержка, пропускная способность и время реакции аналитика. Кроме того, нужно сразу предусмотреть мониторинг дрейфа данных, регулярную переоценку качества, безопасное обновление модели, журналирование решений и возможность отката на предыдущую версию. Для управления рисками здесь логично опираться на процессы системы менеджмента информационной безопасности по ГОСТ Р ИСО/МЭК 27001-2021 и системы менеджмента искусственного интеллекта по ГОСТ Р ИСО/МЭК 42001-2024 [3, 4].

Отдельно надо позаботиться о защите самой системы машинного обучения. Это значит надо контролировать качество и происхождение данных, ограничивать доступ к обучающим выборкам, проверять устойчивость к специально сформированным входам и не допускать автоматической блокировки критичных соединений без соответствующих процедур подтверждения. Порог аномалии, на мой взгляд, лучше выбирать исходя из цены ошибок в конкретной организации, а не только по максимальному значению F1-score.

Заключение

В работе обобщены результаты применения четырёх моделей машинного обучения для обнаружения сетевых атак на наборе CSE-CIC-IDS2018. В эксперименте с контролируемым обучением Random Forest, LSTM и 1D-CNN показали F1-score (macro) 0,9779, 0,9782 и 0,9693 соответственно. Random Forest обеспечил максимальный ROC-AUC 0,9925 и минимальное время инференса 0,008 мс на образец. Применение SHAP позволило выделить наиболее влиятельные сетевые признаки и повысить интерпретируемость классификационных решений.



Автоэнкодер, обучавшийся только на нормальном трафике по семи признакам, при пороге 95-го перцентиля достиг макро-F1 0,8243 и ROC-AUC 0,8775. Полнота по атакам составила 0,74, что указывает на наличие пропусков и необходимость дальнейшей настройки модели. Одновременно подход реконструкции нормального поведения сохраняет практический интерес для первичного поиска отклонений при дефиците размеченных атак.

Итоговый вывод состоит не в выборе одной универсально лучшей архитектуры, а в целесообразности многоуровневой системы, сочетающей быстрый объяснимый классификатор и дополнительный детектор аномалий

Список литературы:

1. Ahmed M., Khan M. A., Tariq U., Shafiq M. Z., Al-Turjman F. Artificial intelligence based cyber security framework for enterprises // IEEE Systems Journal. 2021. Vol. 15, no. 3. P. 4450–4460.
2. Мещеряков Р. В., Мельников С. Ю., Пересыпкин В. А., Хорев А. А. Перспективные направления применения технологий искусственного интеллекта при защите информации // Вопросы кибербезопасности. 2024. № 4 (62). С. 2–12.
3. ГОСТ Р ИСО/МЭК 27001-2021. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Требования. М.: Стандартинформ, 2021. 42 с.
4. ГОСТ Р ИСО/МЭК 42001-2024. Искусственный интеллект. Система менеджмента. М.: Российский институт стандартизации, 2024. 56 с.
5. Vaiyapuri T., Murugesan K. Unsupervised One-Class Learning for IoT Intrusion Detection: A Review of Methods and Prospects // International Journal of Computer Networks and Applications. 2026. P. 179–199. DOI: 10.22247/ijcna/2026/10.
6. CSE-CIC-IDS2018: dataset / Canadian Institute for Cybersecurity, Centre for Security Science. URL: <https://www.unb.ca/cic/datasets/ids-2018.html> (дата обращения: 09.10.2026).
7. Sharafaldin I., Lashkari A. H., Hakak S., Ghorbani A. A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy // Proceedings of the 2019 IEEE International Conference on Cyber Security and Cloud Computing (CSCloud) / IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom). 2019. P. 1–8.
8. Hoogland, D. CSE-CIC-IDS2018: [network intrusion detection dataset] / D. Hoogland; Canadian Institute for Cybersecurity (CIC) @ UNB Fredericton. – 2021. – URL: <https://www.kaggle.com/datasets/dhoogla/csecicids2018/data> (дата обращения: 09.10.2026).
9. Lundberg S. M., Lee S.-I. A Unified Approach to Interpreting Model Predictions // Advances in Neural Information Processing Systems. 2017. Vol. 30

