DOI: 10.15514/ISPRAS-2023-35(4)-3

Применение глубокого обучения для обнаружения компьютерных атак в сетевом трафике

[1,2,3,4] *А.И. Гетьман, ORCID: 0000-0002-6562-9008 <ever@ispras.ru>*
[5] *М.Н. Горюнов, ORCID: 0000-0003-0284-690X <max.gor@mail.ru>*
[5] *А.Г. Мацкевич, ORCID: 0000-0001-9557-3765 <mag3d.78@gmail.com>*
[5] *Д.А. Рыболовлев, ORCID: 0000-0003-4524-655X <dmitrij-rybolovlev@yandex.ru>*
[5] *А.Г. Никольская, ORCID: 0000-0001-5965-4664 <nikolskaya.a.g@yandex.ru>*

[1] *Институт системного программирования им. В.П. Иванникова РАН,*
109004, Россия, г. Москва, ул. А. Солженицына, д. 25.
[2] *Московский физико-технический институт,*
141700, Россия, Московская область, г. Долгопрудный, Институтский пер., 9
[3] *Национальный исследовательский университет «Высшая школа экономики»,*
101978, Россия, г. Москва, ул. Мясницкая, д. 20
[4] *Московский государственный университет имени М.В. Ломоносова*
119991, Россия, г. Москва, Ленинские горы, д. 1
[5] *Академия ФСО России*
302015, Россия, г. Орел, ул. Приборостроительная, д. 35

Аннотация. В работе рассмотрены вопросы применения методов глубокого обучения для обнаружения компьютерных атак в сетевом трафике. Представлены результаты анализа релевантных исследований и обзоров в области применения глубокого обучения для обнаружения вторжений. Произведено описание и сравнение наиболее используемых методов глубокого обучения, предложена система их классификации. Определены существующие тенденции и проблемы применения методов глубокого обучения для обнаружения компьютерных атак в сетевом трафике. Для оценки применимости методов глубокого обучения для обнаружения вторжений синтезирована нейронная сеть CNN-BiLSTM и представлены результаты её сравнения с разработанной ранее моделью, основанной на использовании классификатора типа «случайный лес». Использование метода глубокого обучения позволило упростить этап конструирования признаков, что вместе с близостью полученных значений метрик для сравниваемых моделей подтверждает перспективность применения методов глубокого обучения для обнаружения вторжений.

Ключевые слова: информационная безопасность; система обнаружения атак; обнаружение вторжений; машинное обучение; глубокое обучение; нейронная сеть; свёрточная нейронная сеть; случайный лес; сетевой трафик; компьютерная атака.

Для цитирования: Гетьман А.И., Горюнов М.Н., Мацкевич А.Г., Рыболовлев Д.А., Никольская А.Г. Применение глубокого обучения для обнаружения компьютерных атак в сетевом трафике. Труды ИСП РАН, том 35, вып. 4, 2023 г., стр. 65–92. DOI: 10.15514/ISPRAS–2023–35(4)–3.

Deep Learning Applications for Intrusion Detection in Network Traffic

[1,2,3,4] *A.I. Getman, ORCID: 0000-0002-6562-9008<ever@ispras.ru>*
[5] *M.N. Goryunov, ORCID: 0000-0003-0284-690X <max.gor@mail.ru>*
[5] *A.G. Matskevich, ORCID: 0000-0001-9557-3765 <mag3d.78@gmail.com>*
[5] *D.A. Rybolovlev, ORCID: 0000-0003-4524-655X <dmitrij-rybolovlev@yandex.ru>*
[5] *A.G. Nikolskaya, ORCID: 0000-0001-5965-4664 <nikolskaya.a.g@yandex.ru>*

[1] *Ivannikov Institute for System Programming of the Russian Academy of Sciences,*
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia.
[2] *Moscow Institute of Physics and Technology (National Research University)*
9 Institutskiy per., Dolgoprudny, Moscow Region, 141701, Russia.
[3] *National Research University «Higher School of Economics»*
20, Myasnitskaya ulitsa, Moscow 101978, Russia
[4] *Lomonosov Moscow State University*
1, Leninskie Gory, Moscow, 119991, Russia
[5] *The Academy of Federal Security Guard Service of the Russian Federation,*
35, Priborostroitelnaya st., Oryol, 302015, Russia

Abstract. The paper discusses the issues of applying deep learning methods for detecting computer attacks in network traffic. The results of the analysis of relevant studies and reviews of deep learning applications for intrusion detection are presented. The most used deep learning methods are discussed and compared. The classification system of deep learning methods for intrusion detection is proposed. Current trends and challenges of applying deep learning methods for detecting computer attacks in network traffic are identified. The CNN-BiLSTM neural network is synthesized to assess the applicability of deep learning methods for intrusion detection. The synthesized neural network is compared to the previously developed model based on the use of the Random Forest classifier. The usage of the deep learning method enabled to simplify the feature engineering stage, and evaluation metrics of Random Forest and CNN-BiLSTM models are close. This confirms the prospects for the application of deep learning methods for intrusion detection.

Keywords: information security; network intrusion detection system; intrusion detection; machine learning; deep learning; neural network; convolutional neural network; random forest; network traffic; computer attack.

For citation: Getman A.I., Goryunov M.N., Matskevich A.G., Rybolovlev D.A., Nikolskaya A.G. Deep Learning Applications for Intrusion Detection in Network Traffic. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 4, 2023. pp. 65-92 (in Russian). DOI: 10.15514/ISPRAS-2023-35(4)-3.

1. Введение

В последние десятилетия наблюдается активное развитие информационно-коммуникационных технологий и оказываемых с их помощью услуг, неизбежный рост объёмов передаваемых данных. Соответственно увеличивается количество угроз и факторов, которые приводят к нарушению функционирования информационных систем и компьютерных сетей. Вследствие этого актуальными являются проблемы обеспечения информационной безопасности в целом и вопросы развития средств обнаружения компьютерных атак в частности.

Для выявления компьютерных атак в сетевом трафике в настоящее время в основном используются сигнатурные анализаторы, являющиеся частью сетевой системы обнаружения атак (СОА, IDS). Анализ опубликованных исследований в области построения современных систем обнаружения атак свидетельствует о возможности применения в них методов искусственного интеллекта и машинного обучения. Одним из основных преимуществ эвристических анализаторов СОА, основанных на использовании методов машинного обучения, по сравнению с традиционными сигнатурными анализаторами является способность выявлять новые виды атак. Вместе с тем при разработке таких эвристических анализаторов исследователи сталкиваются со множеством объективных трудностей:

отсутствие доверенных размеченных данных для обучения, необходимость разработки и/или поддержания актуальными генераторов атак, сложность формирования оптимального признакового пространства для решения задачи переноса обучения, необходимость защиты модели машинного обучения против состязательных атак и др.

В настоящей работе предпринята попытка провести анализ релевантных исследований и обзоров в области применения глубокого обучения для обнаружения вторжений. Новизна исследования заключается в описании и сравнении наиболее используемых методов глубокого обучения, предложенной системе их классификации. Представленные результаты отражают современное состояние научных исследований в рассматриваемой предметной области.

В исследовании [1] отмечается, что для использования методов машинного обучения в СОА обычно требуется преобразование данных в набор признаков, то есть формирование признакового пространства, и подчеркивается важность отбора подходящих признаков для точности данных методов. Для решения задач обработки признакового пространства большой мощности, выбора оптимального признакового пространства и ряда других всё чаще используют так называемые методы глубокого обучения нейронных сетей. Данные методы позволяют повысить степень автоматизации обработки данных большой размерности и добиться лучшего качества обнаружения атак по сравнению с другими методами машинного обучения.

Сегодня не существует чётко определённых стандартов в применении методов глубокого обучения для решения задачи обнаружения компьютерных атак. Рассматриваемая предметная область активно развивается, что обусловливает актуальность и подтверждает необходимость проведения системного анализа применяемых методов, их классификации и сравнения, исследования имеющихся проблем и возможных способов их решения. Многообразие и вариативность характеристик методов глубокого обучения при решении задачи обнаружения компьютерных атак также обусловливает актуальность задачи их классификации.

Для оценки применимости методов глубокого обучения для обнаружения вторжений в исследовании синтезирована нейронная сеть CNN-BiLSTM [2] и представлены результаты её сравнения с разработанной ранее моделью [3], основанной на использовании классификатора типа «случайный лес» (Random Forest), на публичном наборе данных обучения CICIDS2017 [4]. Разработанная модель нейронной сети предназначена для использования в сетевой СОА.

2. Глубокое обучение и архитектуры нейронных сетей

Глубокое обучение применяется к искусственным нейронным сетям, состоящим из объединённых в слои искусственных нейронов. В такой сети входные данные обрабатываются в процессе прохождения от первого слоя, называющегося входным, через промежуточные (скрытые) слои до последнего (выходного) слоя. Если промежуточных слоёв больше одного, такая искусственная нейронная сеть называется глубокой.

Функция, в виде которой представляются нейроны, называется функцией активации. Значение функции активации зависит от взвешенной суммы входов нейрона и порогового значения, при этом выходом нейрона является результат применения функции активации к скалярному произведению входного вектора и вектора весов нейрона, смещенное на заданное расстояние [5]. Примерами нелинейных функций, используемых в качестве функции активации, являются сигмоида, функция softmax, линейный выпрямитель (Rectified linear unit, ReLU), гиперболический тангенс.

В обучении нейронной сети используется функция потерь (ошибки), которая характеризует разницу между правильным значением целевой переменной и значением, предсказанным нейронной сетью.

Различные конфигурации нейронов, слоёв, их связей между собой порождают различные архитектуры нейронных сетей. Методы глубокого обучения, таким образом, можно классифицировать по используемой архитектуре нейронной сети.

Кроме того, как и традиционные методы машинного обучения, методы глубокого обучения можно разделить на основные группы: методы глубокого обучения с учителем, с частичным привлечением учителя, без учителя, с подкреплением. Возможность создавать сложные архитектуры нейронных сетей, объединяя разные архитектуры в одной сети, приводит к существованию гибридных методов.

Далее приведено краткое описание основных методов глубокого обучения, которые фигурируют в проведённом анализе релевантных работ.

Искусственная нейронная сеть (Artificial Neural Network, ANN), как упоминалось выше, может быть как *глубокой (Deep Neural Network, DNN)*, так и *неглубокой (Shallow Neural Network, S-NN)* в зависимости от числа скрытых слоёв. Базовый вариант таких сетей представляет собой сеть прямого распространения (Feed forward neural network), в которой сигнал распространяется строго от входа к выходу. Обучение обычно осуществляется методом обратного распространения ошибки.

Многослойный перцептрон (Multilayer Perceptron, MLP) представляет собой разновидность перцептрона – простейшего вида нейронных сетей, основанного на математической модели восприятия информации мозгом, предложенной Ф. Розенблаттом [6]. Перцептроны состоят из сенсоров, ассоциативных и реагирующих элементов. MLP является полносвзяной ANN, при этом между входным и выходным слоями может быть один или несколько скрытых слоев. Многослойный перцептрон обучается с учителем при помощи алгоритма обратного распространения ошибки. Более современной разновидностью MLP по Розенблатту является многослойный перцептрон по Румельхарту [7].

Свёрточная нейронная сеть (Convolutional Neural Network, CNN) представляет собой однонаправленную многослойную сеть с чередующимися свёрточными (convolution layers) и субдискретизирующими (subsampling layers или pooling layers) слоями. Свёрточный слой формирует карту признаков путём поэлементного умножения матрицы весов (ядра свёртки) на каждый фрагмент входного слоя и суммирования результата, который записывается в аналогичную позицию выходного слоя [5]. Субдискретизирующий слой уменьшает размерность карт признаков на отдельных слоях путём выбора отдельного нейрона карты среди соседних. CNN позволяет выделять в данных иерархию абстрактных признаков. Изначально CNN применялись для обработки изображений, но в настоящий момент используются и в других задачах. Использование CNN накладывает ограничение на входные данные: они должны быть представлены в виде нормализованного «изображения». В плане применения к задаче обнаружения вторжений это значит, что каждый вектор признаков из набора данных должен быть преобразован в условное «изображение», имеющее формат «сетки» или таблицы, и нормализован. CNN может обучаться с учителем или без учителя, чаще всего для обучения используется алгоритм обратного распространения ошибки [8].

Рекуррентная нейронная сеть (Recurrent Neural Network, RNN) представляет собой сеть, где связи между нейронами образуют направленную последовательность, при этом нейроны имеют внутреннюю память и могут передавать данные самим себе. Такая архитектура сети позволяет RNN иметь динамическое поведение во времени и обуславливает способность обрабатывать последовательные данные произвольной длины. Однако такая способность приводит к высоким требованиям к ресурсам и проблеме исчезающего (или взрывного) градиента: долгосрочная информация должна последовательно проходить через все ячейки и может быть повреждена многократным умножением на слишком малые (или большие) числа, происходящем при вычислении весов сети методом обратного распространения ошибки во времени (Backpropagation through time, BPTT) [9]. Рекуррентная нейронная сеть может обучаться с различной степенью привлечения учителя, как и её разновидности – LSTM и

GRU, описанные ниже. RNN и её разновидности чаще всего обучаются при помощи BPTT или метода рекуррентного обучения в реальном времени (Real-time recurrent learning, RTRL).

Длинная цепь элементов краткосрочной памяти (Long Short-Term Memory, LSTM) предназначена для решения проблемы исчезающего градиента, свойственной RNN. Сохранение данных в LSTM обеспечивается рекуррентным LSTM-модулем, представляющим собой ячейку памяти, содержащую специальные структуры, реализующие функцию активации – «вентили» или «гейты» (gates). LSTM-модуль позволяет запоминать значения как на короткие, так и на длинные промежутки времени.

Управляемый рекуррентный блок или нейрон (Gated Recurrent Unit, GRU) использует похожий на LSTM вариант ячейки, но с меньшим количеством «вентилей». Таким образом, GRU имеет меньше параметров, чем LSTM, и требует меньше ресурсов для обучения.

Двунаправленная GRU (Bidirectional GRU, BGRU) и *двунаправленная LSTM (Bidirectional LSTM, BiLSTM)* являются разновидностями GRU и LSTM соответственно, позволяющими сети предсказывать результат на основании не только уже обработанных данных, но и всей последовательности целиком. Такие сети имеют два направления вычислений: выходные блоки нейронов вычисляют представление, зависящее как от прошлого, так и от будущего.

Автокодировщик или автоэнкодер (Autoencoder, AE) состоит из входного слоя, кодирующего данные (энкодер), скрытого слоя и выходного слоя, декодирующего данные (декодер). Автокодировщики применяются для автоматического кодирования информации: нейросеть ищет обобщения и корреляцию в поступающих на вход данных и выполняет их сжатие, при этом на выходном слое должен получаться отклик, наиболее близкий к входным данным. Необходимы некоторые ограничения, чтобы AE не обучался простому копированию результатов [8]: например, в стандартном автокодировщике промежуточный слой имеет меньшую размерность, чем входной и выходной слои. Ограничение количества одновременно активных нейронов промежуточного слоя позволяет получить разновидность автокодировщика – *разряженный автокодировщик (Sparse Autoencoder)*. В разряженном автокодировщике размерность промежуточного слоя превышает размерность входного и выходного слоёв. Автокодировщики обучаются без учителя и используют метод обратного распространения ошибки.

Последовательное объединение автокодировщиков позволяет получить *многослойный автокодировщик (Stacked Autoencoder, SAE)*.

Глубокая сеть доверия (Deep Belief Network, DBN) представляет собой композицию подсетей, в которой скрытый слой каждой подсети служит видимым слоем для следующей подсети, и при обучении каждая подсеть должна научиться кодировать предыдущую. Подсетью в данном случае является автокодировщик или ограниченная машина Больцмана. При этом нейроны внутри скрытых слоев не связаны друг с другом, но связаны с нейронами соседнего слоя. DBN обучаются без учителя при помощи жадного послойного обучения.

3. Анализ релевантных работ

В данном разделе приведён анализ релевантных работ и сравнение описанных в них методов глубокого обучения.

Большинство проанализированных в данной работе исследований были отобраны при помощи программы «Publish or Perish» [10] из базы Google Scholar [11]. По запросам «"intrusion detection system" AND "neural network"» и «"intrusion detection system" AND "neural networks"» были отобраны по 50 работ с 2017 по 2021 годы. После объединения полученных списков работ и удаления дубликатов, было получено 94 работы, из которых были выбраны 12 релевантных работ с количеством цитирований не менее 10.

В работе [12] для решения задачи обнаружения вторжений используется CNN-BiLSTM: CNN выявляет пространственные признаки, BiLSTM – временны́е. Применяется гибридный сэмплинг (OSS вместе со SMOTE) для уменьшения времени обучения и балансировки

наборов данных. После гибридного сэмплинга время обучения сократилось для всех сравниваемых моделей, при этом CNN-BiLSTM уступила в скорости обучения LeNet-5, но показала лучшие результаты по всем остальным метрикам. Сравнение проводилось не на самых современных наборах данных: NSL-KDD и UNSW-NB15.

В статье [13] при решении задачи обнаружения вторжений глубокая нейронная сеть (DNN) с 3 скрытыми слоями показала наилучшие результаты по сравнению с классическими алгоритмами машинного обучения. Сравнение проводилось на наборе данных KDD Cup 99 для DNN с 1-5 скрытыми слоями и алгоритмами Ada Boost, Decision Tree, K-Nearest Neighbour, Linear Regression, Navie Bayes, Random Forest, SVM*-Linear, SVM*-rbf. Однако авторами отмечается, что необходимо проводить исследования на более современных наборах данных и в реальных условиях, в том числе в состязательных средах (adversarial environment). Для данной статьи имеется общедоступный репозиторий с кодом и используемым набором данных [14], ссылка на который, однако, не указана в самом исследовании.

В работе [15] для задачи обнаружения вторжений разработана гибридная IDS на основе свёрточно-рекуррентной нейронной сети: CNN выявляет пространственные признаки, RNN – временны́е. Для балансировки набора данных использовалось дублирование примеров миноритарного класса (oversampling). Для оценки производительности в нейросеть перед слоями CNN и RNN были добавлены слои с гауссовским шумом для улучшения обобщающей способности и уменьшения переобучения нейросети. Оценка эффективности проводилась на наборе данных CSE-CIC-IDS2018: использовалось как проведённое авторами сравнение методов, так и данные из других исследований. В проведённом в данном исследовании эксперименте предложенная модель показала лучшие результаты по сравнению с деревом решений, логистической регрессией и алгоритмом XGBoost. Стоит отметить, что, хотя в данном исследовании приводятся два разных сравнения, используются разные наборы оценок и указанные в них оценки не совпадают. Кроме того, авторы указывают на важность тестирования разработанной IDS на более современных данных. Для указанного исследования имеется общедоступный репозиторий с кодом [16], который, однако, на момент написания настоящей работы пуст.

В исследовании [17] для решения задачи обнаружения вторжений используется CNN с предварительным преобразованием данных из векторного формата в «изображение» (матрицу). Для уменьшения размерности пространства признаков использовались метод главных компонент (PCA) и автокодировщик (AE), для оптимизации обучения использовалась пакетная нормализация (batch normalization, BN). По сравнению с традиционными алгоритмами машинного обучения (Naive Bayes, Logistic Regression, Decision Tree, Random Forest, SVM, Adaboost), RNN и трёхслойной DNN, предложенная модель позволяет значительно сократить время обнаружения и показывает лучшие результаты на наборе данных KDD Cup 99. Однако модель показывает низкий уровень обнаружения для атак типов User to Root (U2R) и Remote to Local (R2L) – 20.61% и 18.96% соответственно, в силу малого количества примеров этих атак в используемом наборе данных. В дальнейших исследованиях авторы планируют решать данную проблему путём генерации примеров атак при помощи генеративно-состязательной сети (GAN).

В работе [18] для решения задачи обнаружения вторжений предложена IDS на иерархических пространственно-временных признаках (HAST-IDS), которая сначала «выучивает» низкоуровневые пространственные признаки при помощи CNN, а затем – высокоуровневые временные признаки при помощи двунаправленной LSTM. Исследовались 2 варианта предложенной IDS: HAST-I, которая обучалась только на пространственных признаках, и HAST-II, которая использует и CNN, и LSTM. Выделение признаков происходило автоматически из необработанных данных трафика, производительность оценивалась на наборах данных DARPA 1998 и ISCX 2012. Исследование также определяет оптимальные

длины обрабатываемых данных сетевого соединения и отдельного пакета, которые позволяют провести классификацию. HAST-IDS уменьшает уровень ложных срабатываний (FAR) с помощью улучшения набора признаков, которые не требуется конструировать вручную. Авторы отмечают необходимость усовершенствования предложенного решения для применения к несбалансированным наборам данных, где есть атаки с малым количеством образцов. Также отмечается возможность повышения эффективности решения при обогащении данных признаками, сконструированными вручную.

В статье [19] для решения задачи обнаружения вторжений предложен метод конвертации данных из набора данных NSL-KDD в бинарные векторы, из которых строят «изображение» (матрицу) для классификации при помощи CNN, что позволяет избежать этапа отбора признаков. На подмножествах конвертированного набора данных NSL-KDD протестированные CNN-сети (ResNet 50 и GoogLeNet) показали результаты лучше стандартных классификаторов, но не сильно лучше state-of-the-art решений (сравнение проводилось с J48, Naive bayes, NB Tree, Random forest, Random tree, Multi-layer perceptron, SVM). В статье отмечается необходимость усовершенствования техник репрезентации данных в виде «изображения» для сохранения структурных характеристик данных, к которым чувствительны CNN.

В исследовании [20] при решении задачи обнаружения вторжений применяется метод, позволяющий обнаруживать вторжения на прикладном (application) уровне. Данный метод использует SAE или DBN для конструирования признаков из биграмм символов HTTP-запросов нормального трафика, поступающего на межсетевой экран веб-приложений (web application firewall, WAF). Выполняется параллельное слияние 100 или 30 сконструированных признаков и признаков, полученных из униграмм символов HTTP-запросов нормального трафика (parallel-feature-fusion). Всего в статье проверяется три сценария: простое конструирование признаков при помощи N-грамм, извлечение признаков при помощи нейронных сетей (SAE и DBN) или методов понижения размерности (PCA, KPCA, FICA), параллельное слияние признаков. Эффективность вариантов в данных сценариях сравнивается при помощи одноклассовых классификаторов: One-Class SVM, изоляционный лес, эллиптический конверт (Elliptic Envelope). Сравнение проводится на наборах данных CSIC 2010 и ECML/PKDD 2007. Предложенный метод на основе модели глубокого обучения и слияния признаков показал лучший результат точности и обобщения при разумном времени обнаружения. Варианты с DNN оказались наиболее сбалансированы по точности, обобщению и скорости.

В работе [21] предлагается использовать рекуррентные нейронные сети (RNN) для решения задачи обнаружения вторжений. И для бинарной, и для многоклассовой классификации RNN показывает производительность лучше, чем традиционные алгоритмы (J48, ANN, RF, SVM и др.) на наборе данных NSL-KDD, хоть и требует больше времени на обучение. Предложенный метод также показывает лучшие результаты, чем RNN уменьшенного размера [22] на наборе данных KDD CUP 1999. В статье также изучают выбор гиперпараметров: влияние количества нейронов и скорости обучения на точность метода. Авторы данной работы отмечают имеющиеся у данного метода проблемы исчезающего и взрывающегося градиентов и собираются в дальнейшем исследовать LSTM и Bidirectional RNN для решения данных проблем.

В исследовании [23] протестировали различные архитектуры RNN-сетей (RNN, LSTM, GRU) при решении задачи обнаружения вторжений. Подробно рассматриваются процесс выбора гиперпараметров и проведённые на наборах данных KDD Cup 99 и UNSW-NB15 эксперименты, в том числе эксперименты с малым количеством признаков (4, 8, 11). По сравнению с не-рекуррентными сетями, RNN-сети показали меньший уровень ошибок первого рода (false positives). LSTM давала лучшие результаты, GRU – близкие к ней. На наборе данных UNSW-NB15 результаты были хуже, чем на KDD Cup 99, поскольку в UNSW-NB15 содержится больше различных типов атак. Авторы исследования отмечают, что RNN-

сети хорошо выделяли динамические паттерны, но тестирования только на синтетических наборах данных недостаточно.

В статье [24] предложена модель обнаружения вторжений, объединяющая искусственные нейронные сети с отбором признаков, основанном на корреляции (Correlation based Feature Selection, CFS). Модель была реализована при помощи инструмента RapidMiner и проверена на наборах данных NSL-KDD и UNSW-NB15. Использование CFS позволило за счёт сокращения размерности данных повысить точность, специфичность и чувствительность модели, сократить вычислительное время. Было проведено сравнение реализованной модели с другими современными подходами: данная модель показывает лучшие результаты, однако требует больше вычислительного времени. Предложенный подход может применяться в различных сетях связи, для защиты серверов интернета вещей (Internet of Things, IoT).

В статье [25] для решения задачи обнаружения вторжений предложена модель, состоящая из двух нейронных сетей: Shallow Neural Network (S-NN) и Deep-Optimized Neural Network (D-ONN). Сеть S-NN более простая и быстрая, D-ONN – более сложная и медленная. Отбор признаков осуществлялся методом корреляционного анализа и с применением энтропийного подхода. Данная модель показала наилучший результат на наборе данных KDD Cup 99. Авторы отмечают возможность использования данного метода для защиты беспроводных сетей и IoT.

В статье [26] предложена модель BGRU+MLP для решения задачи обнаружения вторжений. В экспериментах использовались наборы данных KDD Cup 99 и NSL-KDD. По результатам экспериментов, GRU показывает результаты лучше, чем LSTM, BGRU – лучше, чем GRU в отдельности, а сочетание BGRU и MLP даёт лучшие результаты по сравнению с отдельным использованием RNN (GRU или LSTM) или MLP. BGRU+MLP показывает лучшие результаты по точности, количеству обнаруженных инцидентов и доле ложноположительных примеров (FPR), однако имеются проблемы с выявлением атак типа R2L и U2R. Авторы отмечают, что данная проблема свойственна и системам других исследователей в силу малой доли данных атак в используемых наборах данных. Кроме того, используемая RNN-сеть имеет больше преимуществ при работе с временными рядами, а у атак R2L и U2R меньше характеристик, очевидно связанных со временем.

В исследовании [27] для решения задачи обнаружения вторжений предложена модель SFSDT+RNN. Данная модель предназначена для улучшения точности выявления атак, в том числе отдельных типов атак – в частности, упомянутых ранее R2L и U2R. Для отбора признаков используется гибридный алгоритм SFSDT: при помощи алгоритма последовательного прямого выбора (SFS) отбираются наиболее релевантные наборы признаков, среди которых определяется лучший набор признаков при помощи дерева принятий решений (DT). Эксперименты проводились на наборах данных NSL-KDD и ISCX 2012. Модель с использованием LSTM показала лучшую точность среди трёх видов RNN (RNN, LSTM, GRU). Благодаря отбору признаков с помощью SFSDT уменьшились время вычисления и использование памяти. Стоит отметить, что в данном исследовании не указаны подробности конкретных реализаций использованных в экспериментах архитектур (RNN, LSTM, GRU).

Отдельно стоит остановиться на исследованиях, представляющих собой аналитические обзоры в области применения глубокого обучения в IDS.

В статье [28] представлен обзор литературы по использованию нейронных сетей в IDS за 2015-2019 годы. В исследование вошли обзоры литературы, предложения новых методов и обучающие статьи. Рассматриваются наиболее используемые в системах обнаружения атак архитектуры нейронных сетей, распространённые и частные наборы данных, особенности их использования. Поднимается проблема последствий для безопасности при использовании нейронных сетей в IDS.

В работе [29] представлен обзор литературы по использованию машинного обучения и нейронных сетей в IDS с точки зрения предложенной авторами таксономии IDS. В исследование вошли обзоры классификаций IDS, часто используемых в IDS алгоритмов машинного обучения, метрик, наборов данных. Отмечены имеющиеся проблемы предметной области и будущие направления исследований.

В исследованиях [5, 30] представлен аналитический обзор в области глубокого обучения для задач кибербезопасности. Рассматривается применение различных методов глубокого обучения в зависимости от конкретного приложения кибербезопасности. Авторами сделаны выводы по применимости и особенностям применения методов глубокого обучения в задачах кибербезопасности.

4. Классификация методов глубокого обучения

Для осуществления анализа современных методов глубокого обучения в области обнаружения вторжений было выполнено сравнение рассмотренных выше методов. Стоит отметить, что исследователи в своих работах фокусируются на различных аспектах, не всегда предоставляя полную информацию о реализации предложенных ими методов и проведённых экспериментах.

На основании проведённого анализа релевантных работ была разработана классификация методов глубокого обучения в области обнаружения вторжений. Выделенные классификационные признаки условно могут быть разделены на две группы: отражающие основные характеристики и особенности метода и отражающие результаты использования данного метода в практическом применении (в экспериментах).

Таким образом, предлагается следующая классификация методов глубокого обучения.

1) Основные характеристики метода:
   - предложенный метод;
   - уровень обнаружения атак (сетевой, хостовой, прикладной);
   - сочетание с другими методами (метод глубокого обучения используется вместе с не DL-методами);
   - особенности архитектуры предложенного метода;
   - предобработка данных;
   - конструирование признаков;
   - вспомогательные приёмы (методы оптимизации, алгоритмы сэмплирования, техники регуляризации и т.п.).
2) Результаты экспериментальных исследований:
   - методы, с которыми производилось сравнение в исследовании;
   - набор данных;
   - вид классификации (бинарная, многоклассовая);
   - оценка;
   - оборудование и программное обеспечение;
   - время обучения и/или выполнения.

Предложенная классификация позволяет систематизировать знания в рассматриваемой предметной области и может быть использована для проведения дальнейших исследований и проведения сравнительного анализа методов глубокого обучения согласно предложенной классификации, результаты которого представлены в табл. 1 и табл. 2. Как этот анализ показывает, обнаружение атак на прикладном уровне или уровне хоста встречается всего в

двух примерах, а подавляющее большинство реализованных исследователями методов глубокого обучения (11 из 13) обнаруживают атаки на сетевом уровне.

Табл. 1. Сравнение методов глубокого обучения в области обнаружения вторжений по основным характеристикам

Table 1. Comparison of deep learning methods for intrusion detection by main features

Работа, год, ссылка	Уровень обнаружения	Сочетание с другими методами	Предложенный метод	Особенности архитектуры	Предобработка данных	Конструирование признаков	Вспомогательные приёмы
K. Jiang et al, 2020 [12]	Сетевой (network)	+	CNN-BiLSTM	2 свёрточных слоя CNN, 2 скрытых слоя BiLSTM	- OHE; - min-max нормализация; - данные преобразуются в матрицу (ч/б изображение)	- CNN: пространственные признаки; - BiLSTM: временны́е признаки	- гибридный сэмплинг (OSS + SMOTE)
Rahul Vigneswaran et al, 2018	Сетевой (network)	-	DNN	От 1 до 5 скрытых слоёв (оптимально: 3)			- Back propagation; - Dropout
M.A. Khan, 2021 [15]	Сетевой (network), на уровне хоста (host)	+	HCRNN (CNN-RNN)	2 свёрточных слоя CNN, 2 скрытых слоя RNN	данные преобразуются в матрицу (ч/б изображение)	- удаление части признаков (IP-адреса и временные метки); - CNN: пространственные признаки; - RNN: временны́е признаки	- Oversampling; - слои с Гауссовским шумом; - оптимизация гипер-параметров при помощи случайного поиска
Yihan Xiao et al, 2019 [17]	Сетевой (network)	+	CNN	Lenet-5 + слой Dropout	- OHE; - min-max нормализация; - данные преобразуются в матрицу (ч/б изображение)	PCA/AE для уменьшения размерности пространства признаков	- пакетная нормализация (Batch normalization); - Dropout
Wei Wang et al, 2017 [18]	Сетевой (network)	-	HAST-IDS (CNN-BiLSTM)	*HAST-I:* 2 свёрточных слоя CNN *HAST-II:* 4 свёрточных слоя CNN, 2 скрытых слоя BiLSTM	- OHE; - данные преобразуются в матрицу (ч/б изображение)	- CNN: пространственные признаки; - BiLSTM: временны́е признаки	
Zhipeng Li et al, 2017 [19]	Сетевой (network)	-	CNN	ResNet50, GoogLeNet	- OHE; - min-max нормализация; - данные преобразуются в матрицу (ч/б изображение)		- градиентный спуск; - кросс-энтропия как функция потерь

Работа, год, ссылка	Уровень обнаружения	Сочетание с другими методами	Предложенный метод	Особенности архитектуры	Предобработка данных	Конструирование признаков	Вспомогательные приёмы
Ali Moradi Vartouni et al, 2019 [20]	Прикладной (application)	+	parallel-feature-fusion + SAE/DBN + 1-SVM/IF/Elliptic	От 3 до 4 скрытых слоёв (оптимально: 4)		SAE/DBN + parallel-feature-fusion	
C. Yin et al, 2017 [21]	Сетевой (network)	-	RNN	От 20 до 240 скрытых узлов (оптимально: 80 с NSL-KDD, 20 – с KDD CUP 1999)	- OHE; - min-max нормализация		- кросс-энтропия; - оптимизация коэффициента скорости обучения
Vinayakumar R. et al, 2017 [23]	Сетевой (network)	-	RNN (RNN, LSTM, GRU)	От 1 до 4 скрытых слоёв с 32 ячейками памяти			- BPTT; - ADAM; - кросс-энтропия
Sumaiya Thaseen et al, 2020 [24]	Сетевой (network)	+	CFS + ANN	6 скрытых слоёв	min-max нормализация	CFS	
Mangayarkarasi Ramaiah et al, 2021 [25]	Сетевой (network)	+	S-NN, D-ONN	S-NN без скрытых слоёв, D-ONN с 2 скрытыми слоями	- LabelEn-coder; - стандарти-зация; - диаграмма размаха для выявления выбросов	- корреляция; - RF	ADAM
Congyuan Xu et al, 2018 [26]	Сетевой (network)	-	BGRU + MLP	MLP: 3 слоя, 48 скрытых узлов; BGRU: 128 скрытых блоков	- 1-to-N encoding; - min-max нормализация		- SGD; - Backpropa-gation; - BPTT; - кросс-энтропия
Thi-Thu-Huong Le et al, 2019 [27]	Сетевой (network)	+	SFSDT + RNN (RNN, LSTM, GRU)	RNN, LSTM, GRU	восстанов-ление пропущенных значений	SFSDT: SFS + DT	BPTT

Табл. 2 Сравнение методов глубокого обучения в области обнаружения вторжений по результатам экспериментальных исследований

Table 2. Comparison of deep learning methods for intrusion detection by experimental results

Работа, год, ссылка	Предложенный метод	Сравниваемые методы	Набор данных	Вид классификации	Оценка, %	Оборудование и ПО	Время обучения (применения), с
K. Jiang et al, 2020 [12]	CNN-BiLSTM	RF, AlexNet, LeNet-5, CNN, BiLSTM	NSL-KDD	много-классовая	ACC=83.58, Precision= 85.82, Recall=84.49, F1=85.14	Intel i3-7100U, 12 GB RAM; Windows 10, TensorFlow, Keras 2.2, Python	341.69 (-)
			UNSW-NB15	много-классовая	ACC=77.16, Precision=82.63 , Recall=79.91, F1=81.25		2750.47 (-)
Rahul Vigneswaran et al, 2018 [13]	DNN	Ada Boost, Decision Tree, K-NN, Linear Regression, Naive Bayes, Random Forest, SVM*-Linear, SVM*-rbf	KDD CUP 1999	бинарная	*DNN (3 слоя):* ACC=93, Precision=99.7, Recall=91.5, F1=95.5	Nvidia GK110BGL Tesla k40; Keras, TensorFlow	
M.A. Khan, 2021 [15]	HCRNN (CNN-RNN)	LR, XGB, Decision Tree, HCRNN	CSE-CIC-IDS 2018	бинарная	Precision= 0.9633, Recall=0.9712, F1=0.976, DR=0.97, FAR=2.5	NVIDIA GTX 1080ti; 64-bit, 32 GB RAM, 32-core processor, desktop computer Core I7; Java (JDK) 12, Deeplearning4j 1.0.0. alpha, Spark v2.3.0	
		DBN, DNN, Deep learning, LSTM, IDS using DL, CNN IDS			ACC=97.75, FAR=1.4		
Yihan Xiao et al, 2019 [17]	CNN	Naive Bayes, LR, Decision Tree, Random Forest, SVM, Adaboost, RNN, DNN (3 слоя)	KDD CUP 1999	много-классовая	ACC= 94, DR=93, FAR=0.5	NVIDIA GTX 1080ti; Intel i7-8700k, 32 GB RAM; Windows 10, Keras 1.2	20 (11)

Работа, год, ссылка	Предложенный метод	Сравниваемые методы	Набор данных	Вид классификации	Оценка, %	Оборудование и ПО	Время обучения (применения), с
Wei Wang et al, 2017 [18]	HAST-IDS (CNN-BiLSTM)	PLSSVM, Multi-Classifier, Random Forest, Bayes Net, JRip, SVM, Naïve Bayes, ID3, EID3, MARK-ELM	DARPA1998	много-классовая	*HAST-I:* ACC=99.68, FAR=0.07, DR=97.78	NVIDIA Tesla K40m; Ubuntu 16.04 64-bit OS с Keras и TensorFlow, сервер DELL R720, 16 CPU ядер, 16GB RAM	58 мин (1.7)
		MHCVF, ALL-AGL, KMC + NBC, AMGA2-NB	ISCX 2012	много-классовая	*HAST-I:* ACC=99.69, FAR=0.22, DR=96.91 *HAST-II:* ACC=99.89, FAR=0.02, DR=96.96		
Zhipeng Li et al, 2017 [19]	CNN	J48, Naive Bayes, NB Tree, Random Forest, Random Tree, MLP, SVM	NSL-KDD Test^{+}	бинарная	*ResNet50:* ACC=79.14, Precision= 91.97, Recall=69.41, F1=79.12 *GoogLeNet:* ACC=77.04, Precision= 91.66, Recall=65.64, F1=76.5	TITAN X Pascal; Dell 7910, TensorFlow	
			NSL-KDD Test^{-21}	бинарная	*ResNet50:* ACC=81.57, Precision= 81.81, Recall=99.63, F1=89.85 *GoogLeNet:* ACC=81.84, Precision= 81.84, Recall=100, F1=90.01		
Ali Moradi Vartouni et al, 2019 [20]	parallel-feature-fusion + SAE/DBN + 1-SVM/ IF/Elliptic	1-gram/2-gram + 1-SVM/IF/ Elliptic, PCA/KPCA/ FICA + 1-SVM/IF/ Elliptic, SAE/DBN + 1-SVM/IF/ Elliptic, SAE100/ DBN100 + Fusion + 1-SVM/IF/ Elliptic, SAE30/DBN 30 + Fusion + IF/Elliptic	CSIC 2010	бинарная	*SAE30 + Fusion + IF:* ACC=89.24, DR=89.48, PR=81.58, Specificity= 89.11, F1= 85.35	Intel Xeon 2.2 GHz (2 Processors), 64.0 GB RAM, Windows 7 (64-bit), Python 3.6, Tensorflow	14.14 (14.05)
			ECML/PKDD 2007	бинарная	*DBN30 + Fusion + Elliptic:* ACC=84.02, DR=89.75, PR=80.61, Specificity= 78.25, F1= 84.93		284.54 (0.25)

Работа, год, ссылка	Предложенный метод	Сравниваемые методы	Набор данных	Вид классификации	Оценка, %	Оборудование и ПО	Время обучения (применения), с
C. Yin et al, 2017 [21]	RNN	J48, Naive Bayesian, NB Tree, Random Forest, Random Tree, MLP, SVM	NSL-KDD Test^{+}	бинарная	ACC=83.28	ThinkPad E450, Intel Core i5-5200U CPU @ 2.20 GHz, 8 GB RAM, Theano	5516 (-)
				много-классовая	ACC=81.29		11444 (-)
			NSL-KDD Test^{-21}	бинарная	ACC=68.55		5516 (-)
				много-классовая	ACC=64.67		11444 (-)
		reduced-size RNN	KDD CUP 1999	много-классовая	DR=97.09		1765 (-)
Vinayakumar R. et al, 2017 [23]	RNN (RNN, LSTM, GRU)	RNN (4, 8, 11); LSTM (4, 8, 11); GRU (4, 8, 11); LR; NB; KNN; DT; RF; AB; RNN, LSTM, GRU	KDD CUP 1999	бинарная	*RNN:* ACC=94.2, Precision=100, Recall=92.8, F1=96.2, Loss=0.22 *LSTM:* ACC=99.9, Precision=100, Recall=99.9, F1=99, Loss=0.01 *GRU:* ACC=99.7, Precision=100, Recall=99.7, F1=99.8, Loss=0.01	Nvidia GK110BGL Tesla k40, Tensorflow, Scikit-learn	
				много-классовая	*RNN:* ACC=95.7, Loss=0.33 *LSTM:* ACC=96.98, Loss=0.31 *GRU:* ACC=95.37, Loss=0.63		
		RNN, LSTM, GRU	UNSW-NB15	бинарная	*RNN:* ACC=88.3, Precision=87.6, Recall=96.5, F1=91.8, Loss=0.25 *LSTM:* ACC=89.9, Precision=88.9, Recall=97.3, F1=92.9, Loss=0.22 *GRU:* ACC=89.7, Precision=88.6, Recall=97.3, F1=92.8, Loss=0.23		
				много-классовая	*RNN:* ACC=58.5, Loss=0.89 *LSTM:* ACC=67.5, Loss=0.79 *GRU:* ACC=64.8, Loss=0.84		

Работа, год, ссылка	Предложенный метод	Сравниваемые методы	Набор данных	Вид классификации	Оценка, %	Оборудование и ПО	Время обучения (применения), с
Sumaiya Thaseen et al, 2020 [24]	CFS + ANN	Naïve Bayes, REP Tree, Decision Tree, SVM, Random Tree, RF, Bagging, Randomizable, AlexNet, BiLSTM, CNN, CNN+BiLSTM, Stacking Ensemble, Pelican, TSDL, Neural Network with reduced feature	UNSW-NB15	много-классовая	ACC=96.44, Specificity=98.4		660 (-)
		SVM, CNN, MLP, Bi-LSTM, Naïve Bayes, C4.5, CART, Random Forest, CNN-BiLSTM, Ensemble, SVM-RBF, SAE-SVM-RBF	NSL-KDD	много-классовая	ACC=97.49, Specificity=99.31		500 (-)
Mangayarkarasi Ramaiah et al, 2021 [25]	Shallow Neural Network Model (S-NN), Deep-Optimized Neural Network Model (D-ONN)	SVM, PIO-SVM, GA-SVM, PSO-SVM, DNN-1, DNN-5, Naïve Bayes, KNN, RF, CNN 3 layer, CNN 1/3 layer.LSTM CNN 1-3 layer.GRU CNN 3 layer.RNN DNN 2-5 layer	KDD CUP 1999	много-классовая	*S-NN:* ACC=91, Precision=93, Recall=93 *D-ONN:* ACC=98, Precision=93, Recall=93, F1=98	Google Colab, Keras Tensorflow from Python 3.7	

Работа, год, ссылка	Предложенный метод	Сравниваемые методы	Набор данных	Вид классификации	Оценка, %	Оборудование и ПО	Время обучения (применения), с
Congyuan Xu et al, 2018 [26]	BGRU + MLP	LSTM, LSSVM-FMIFS, LSTM-RNN, Pruning VELM, GA+FLN, PSO+FLN	KDD CUP 1999	много-классовая	ACC=99.84, DR=99.42, FPR=0.05	Intel Core i7 @ 3.4 GHz, 64 GB RAM, NVIDIA TESLA K40. Ubuntu 16.04 LTS, CUDA 8.0, cuDNN 6.0, TensorFlow 1.4.1	
		OS-ELM, LSSVM-FMIFS, TVCPSO-MCLP, VELM	NSL-KDD	много-классовая	ACC=99.24, DR=99.31, FPR=0.84		
Thi-Thu-Huong Le et al, 2019 [27]	SFSDT + RNN (RNN, LSTM, GRU)	SCDNN, STL, DNN, Gaussian-Bernoulli RBM, Naive Bayes, J48, ANN, CART, MDPCA-DBN, Zscore + Kmeans, RNN	NSL-KDD	много-классовая	*RNN:* ACC=89.6 *LSTM:* ACC=92 *GRU:* ACC=91.8	Windows 10, Python	63 (-)
		NB, Bagged-NB, Boosted-NB, AMGA2-NB, TCM-KNN, Zscore + Kmeans	ISCX 2012	бинарная	*RNN:* ACC=94.75 *LSTM:* ACC=97.5 *GRU:* ACC=97.08		120.83 (-)

Практически в половине (7 из 13) работ нейронная сеть используется в сочетании с другими, не DL-методами. Данные методы применяются для конструирования признаков (предназначены для выделения или сокращения признакового пространства), оптимизации процесса обучения (например, методы для балансировки набора данных или уменьшения переобучения), классификации.

Наиболее часто встречаются RNN с разновидностями и CNN (7 и 5 работ соответственно). При этом в 4 из 13 рассмотренных работ используется не один вид нейронной сети, а их сочетание (например, CNN-RNN). Сочетания различных архитектур в одном методе призваны устранить недостатки конкретных методов или в целом улучшить степень автоматизации всего процесса выявления атак.

Стоит отметить, что не во всех исследованиях указаны подробности архитектуры реализованных методов глубокого обучения. Также заметно отсутствие общепринятой

краткой нотации описания слоёв нейронной сети: в различных работах используются визуализации, словесные описания, математическая нотация или принятые в конкретных фреймворках машинного обучения обозначения. В связи с этим, для каждого рассмотренного метода были отмечены конкретные особенности, указанные в исследовании, например, используемые в работе вариации архитектуры или число скрытых слоёв.

В качестве метода предобработки данных чаще всего встречается быстрое кодирование (One-Hot Encoding, OHE), минимаксная (min-max) нормализация и преобразование данных в изображение (в случае использования CNN).

Преобладание работ, в которых исследователи обращают внимание на методы конструирования признаков, предобработки данных или различные вспомогательные приёмы (например, методы оптимизации), позволяют говорить о важности данных этапов для достижения нейронными сетями полученных в исследованиях высоких результатов. Стоит отметить, что не каждое исследование касается такого важного вопроса, как выбор гиперпараметров.

В экспериментах, проводимых в рассмотренных исследованиях, сравнение предложенных методов глубокого обучения производится в основном с другими методами глубокого обучения (в том числе с вариациями самого предложенного метода) или с различными методами машинного обучения.

Чаще всего сравнение методов проводилось на наборах данных KDD Cup 1999 и NSL-KDD 2009 (по 6 работ соответственно), UNSW-NB15 (3 работы) и ISCX 2012 (2 работы). Самый современный набор данных в проанализированных исследованиях – CSE-CIC-IDS2018; потерявший актуальность – DARPA 1998. Стоит отметить проблему методов глубокого обучения, свойственную также классическим методам машинного обучения: использование для их обучения устаревших несбалансированных наборов данных, не соответствующих современным данным.

Применение методов глубокого обучения для задачи обнаружения вторжения в рассмотренных исследованиях чаще всего проводится для задачи многоклассовой классификации (9 работ). Указанные в исследованиях проблемы с недостаточно качественными результатами в основном наблюдаются для задачи многоклассовой классификации атак и решаются аккуратной подготовкой набора данных (например, балансировкой набора данных) и/или тщательным отбором признаков.

Использованное оборудование и программное обеспечение указано практически в каждом изученном исследовании, однако время обучения оказалось указано примерно в половине проанализированных статей (7 из 13 работ), а время применения – только в 3 исследованиях. При этом для программного обеспечения не всегда указывается его версия. Данные особенности не позволяют провести качественное сравнение рассмотренных методов глубокого обучения для задачи обнаружения вторжения по критерию времени.

Для оценки качества во всех исследованиях используется доля правильных ответов (accuracy, ACC). Другие часто встречающиеся метрики оценки качества – это точность (precision), полнота (recall), F-мера (F1) и частота обнаружения (DR), встречающиеся примерно в половине работ. Следует отметить разнообразие используемых метрик оценки качества.

Отдельно стоит отметить, что только для одной проанализированной работы из 13 имеется общедоступный репозиторий с кодом, и ещё для одной работы наличие такого кода объявлено, однако репозиторий на настоящий момент пуст. Отсутствие общедоступного кода не позволяет однозначно верифицировать результаты экспериментов, изложенные в исследованиях.

5. Экспериментальные данные

Как было указано ранее, в большинстве проанализированных исследований используются устаревшие наборы данных и отсутствуют общедоступные репозитории с кодом. Устранение

данных недостатков позволит обеспечить возможность работы с современными реальными данными и повысить верифицируемость результатов экспериментов.

С учётом данных замечаний был проведён эксперимент, в котором для оценки применимости методов глубокого обучения для обнаружения вторжений была синтезирована нейронная сеть, состоящая из CNN и двунаправленной LSTM. Выбранная архитектура нейронной сети была предложена в исследованиях [31, 32]. Разработанная модель нейронной сети предназначена для использования в сетевой СОА.

Для проверки синтезированной модели был выбран один из наиболее актуальных публичных наборов данных обучения – Intrusion Detection Evaluation Dataset (CICIDS2017). Набор данных CICIDS2017 подготовлен Канадским институтом кибербезопасности по результатам анализа сетевого трафика в изолированной среде, в которой были смоделированы действия 25 легальных пользователей, а также вредоносные действия нарушителей. Набор объединяет более 50 Гб «сырых» данных в формате PCAP и включает 8 предварительно обработанных файлов в формате CSV, содержащих размеченные сетевые сессии с выделенными признаками в разные дни наблюдения. Сетевые сессии относятся к одному из 15 классов: классу нормального трафика («BENIGN») или одному из 14 классов атак («DoS Hulk», «PortScan», «DDoS», «DoS GoldenEye», «FTP-Patator», «SSH-Patator», «DoS slowloris», «DoS Slowhttptest», «Bot», «Infiltration», «Heartbleed», «Web Attack – Brute Force», «Web Attack – XSS», «Web Attack – SQL Injection»).

Синтезированная модель нейронной сети решает задачу бинарной классификации: определяет, относится ли конкретная сетевая сессия к классу нормального трафика или к классам атак, без определения конкретного класса атаки.

Оценка качества классификаторов производилась на разработанной ранее [3] сбалансированной и предварительно обработанной подвыборке веб-атак WebAttacks набора данных CICIDS2017 с 7267 записями и соотношением нормального и аномального трафика 70% / 30%. В подвыборке содержатся 4 класса: «BENIGN» (5087 записей), «Web Attack - Brute Force» (1507 записей), «Web Attack - Sql Injection» (21 запись), «Web Attack - XSS» (652 записи). Каждая запись в наборе данных WebAttacks представляет собой сетевую сессию и характеризуется 84 признаками, например, IP-адресами источника и приёмника потока данных («Source IP» и «Destination IP»), скоростью потока данных («Flow Bytes/s») и др.

Для обучения нейронной сети использовались 76 из 84 признаков сессий, содержащихся в подвыборке. Из признакового пространства были исключены признаки «Flow ID», «Source IP», «Source Port», «Destination IP», «Destination Port», «Protocol», «Timestamp» в силу относительной лёгкости их подделки злоумышленником и предположении, что признаки «формы» (соответствующие статистикам сетевого трафика) являются более значимыми для общего случая [33].

Данные прошли следующую предварительную обработку:

- минимаксная нормализация признаков: данные, поступающие на вход CNN, лежат в пределах [0, 1];
- преобразование категориальных значений меток при помощи быстрого кодирования (OHE).

На рис. 1 представлено описание слоёв модели синтезированной нейронной сети. Модель является последовательной, входным слоем нейросети является одномерный свёрточный слой, принимающий на вход 76 признаков сессии, выходным – полносвязный слой с 2 нейронами, к выходам которых применяется функция softmax. Между свёрточным и полносвязным слоем имеются два слоя BiLSTM, предварённые слоями пакетной нормализации. Между слоями BiLSTM расположен слой, изменяющий размерность выхода для корректной работы слоёв. После данного слоя, а также свёрточного слоя, применяется операция субдискретизации по максимальному значению. Перед полносвязным слоем

выполняется операция прореживания для предотвращения переобучения. В качестве функции потерь используется категориальная перекрёстная энтропия, в качестве оптимизатора нейронной сети – алгоритм ADAM [34].

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 76, 64)            2112
_________________________________________________________________
max_pooling1d_2 (MaxPooling1D)  (None, 15, 64)         0
_________________________________________________________________
batch_normalization_2 (BatchNormalization)  (None, 15, 64)  256
_________________________________________________________________
bidirectional_2 (Bidirectional)  (None, 128)           66048
_________________________________________________________________
reshape_1 (Reshape)          (None, 128, 1)            0
_________________________________________________________________
max_pooling1d_3 (MaxPooling1D)  (None, 25, 1)          0
_________________________________________________________________
batch_normalization_3 (BatchNormalization)  (None, 25, 1)  4
_________________________________________________________________
bidirectional_3 (Bidirectional)  (None, 256)           133120
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 514
_________________________________________________________________
activation_1 (Activation)    (None, 2)                 0
=================================================================
Total params: 202,054
Trainable params: 201,924
Non-trainable params: 130
```

Рис. 1. Краткое описание модели CNN-BiLSTM
Fig. 1. Summary of the CNN-BiLSTM model

Тестирование модели нейронной сети было проведено в сравнении с разработанной ранее моделью, использующей классификатор Random Forest [3, 35], также предназначенной для решения задачи бинарной классификации. При этом для модели Random Forest использовались только 10 признаков, на которых данная модель была обучена:

- «Average Packet Size», средняя длина поля данных пакета TCP/IP;
- «Flow Bytes/s», скорость потока данных;
- «Max Packet Length», максимальная длина пакета;
- «Fwd Packet Length Mean», средняя длина переданных в прямом направлении пакетов;
- «Fwd IAT Min», минимальное значение межпакетного интервала (IAT, inter-arrival time) в прямом направлении;
- «Total Length of Fwd Packets», суммарная длина переданных в прямом направлении пакетов;
- «Flow IAT Mean», среднее значение межпакетного интервала;
- «Fwd Packet Length Max», максимальная длина переданного в прямом направлении пакета;
- «Fwd IAT Std», среднеквадратическое отклонение значения межпакетного интервала в прямом направлении пакетов;
- «Fwd Header Length», суммарная длина заголовков переданных в прямом направлении пакетов.

Тестирование модели проводилось в экспериментальной среде со следующими характеристиками: Windows 8.1 64 bit, 2-ядерный CPU Intel Core i5-3317U 1.7 GHz, ОЗУ 4 GB, Python 3.9.7, Tensorflow 2.6.0. Время обучения составило около 801 с.

Качество ответов модели оценивалось с использованием следующих метрик:

- доля правильных ответов (accuracy);
- точность (precision, насколько можно доверять классификатору);
- полнота (recall, как много объектов класса «есть атака» определяет классификатор);
- F1-мера (F1-measure, гармоническое среднее между точностью и полнотой).

В табл. 3 представлены полученные значения метрик моделей CNN-BiLSTM и Random Forest.

Табл. 3. Результаты оценки качества классификаторов
Table 3. Results of evaluation of quality of classifiers

Модель (алгоритм)	Accuracy	Precision	Recall	F1
CNN-BiLSTM	0.948	0.959	0.862	0.908
Random Forest	0.983	0.975	0.968	0.971

На рис. 2 представлен график сравнения метрик качества моделей CNN-BiLSTM и Random Forest.

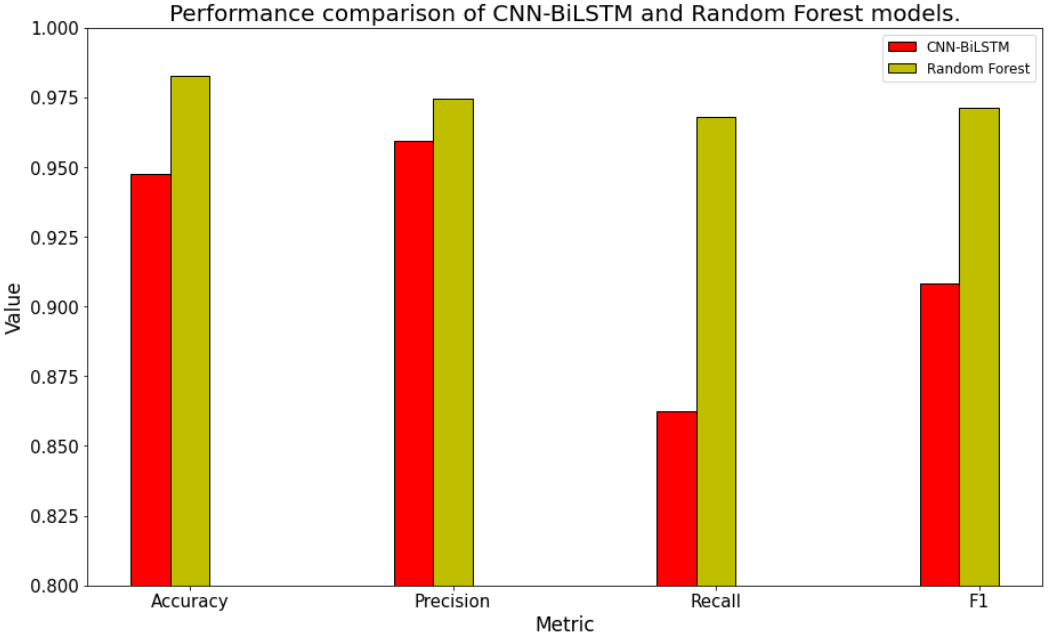

Рис. 2. Сравнение производительности моделей CNN-BiLSTM и Random Forest
Fig. 2. Performance comparison of CNN-BiLSTM and Random Forest models

Как видно из полученных результатов, модель CNN-BiLSTM имеет достаточно высокое качество, однако по всем метрикам уступает модели Random Forest.

На рис. 3 представлена матрица ошибок модели CNN-BiLSTM, а на рис. 4 – такая же матрица модели Random Forest.

Несмотря на то, что модель CNN-BiLSTM показала себя менее эффективной по сравнению с моделью Random Forest, использование метода глубокого обучения позволило исключить этап явного, «ручного» конструирования признаков. Близость полученных значений метрик для сравниваемых моделей подтверждает перспективность применения методов глубокого обучения для обнаружения вторжений. Для достижения необходимого уровня эффективности модели требуются проведение дальнейших исследований и более точная оптимизация нейронной сети для решения поставленной задачи.

Исходный код проекта доступен для выполнения в среде Google Colaboratory: https://colab.research.google.com/github/fisher85/ml-cybersecurity/blob/master/python-web-attack-detection/web-attack-detection-using-CNN-BiLSTM.ipynb.

6. Выводы

В настоящем исследовании проанализированы наиболее используемые методы глубокого обучения в области обнаружения вторжений и предложена система их классификации, основанная на двух группах классификационных признаков: отражающих основные характеристики и особенности метода и отражающих результаты использования данного метода в практическом применении, то есть экспериментах. На основании предложенной классификации было проведено сравнение методов глубокого обучения в области обнаружения вторжений. По результатам проведённого анализа методов глубокого обучения определены существующие тенденции и проблемы рассматриваемой предметной области.

Как видно из анализа релевантных работ, большинство используемых в исследованиях последних лет архитектур показывают хорошие результаты, независимо от того, используется какой-то один вид нейронной сети (например, RNN) или их сочетание (например, CNN-RNN). Сочетания призваны устранить недостатки конкретных методов или в целом улучшить степень автоматизации всего процесса выявления атак.

Отметим, что не столько применение методов глубокого обучения даёт преимущество над классическими ML-методами, сколько именно грамотное их использование со всеми подготовительными и вспомогательными приёмами. Применение только нейронных сетей или нейронных сетей в сочетании с другими, не DL-методами, даёт, в целом, сопоставимо хорошие результаты.

Из собранной выборки релевантных работ можно сделать вывод, что значительной популярностью у исследователей пользуются RNN с разновидностями, CNN и сочетания указанных архитектур. В настоящий момент можно говорить, что данные архитектуры хорошо себя зарекомендовали. Возможно, имеется проблема в том, что в охватываемый рассмотренными исследованиями временной период данные архитектуры пользовались повышенным интересом по сравнению с другими архитектурами, которым уделялось меньше внимания со стороны исследователей. Например, более поздние статьи, но ещё мало цитируемые и потому не попавшие в обзор, широко используют автокодировщики, графовые нейронные сети, трансформеры.

При описании полученных результатов авторы столкнулись с проблемой отсутствия общепринятой краткой нотации описания слоёв нейронной сети: в различных работах используются визуализации, словесные описания, математическая нотация или принятые в конкретных фреймворках машинного обучения обозначения.

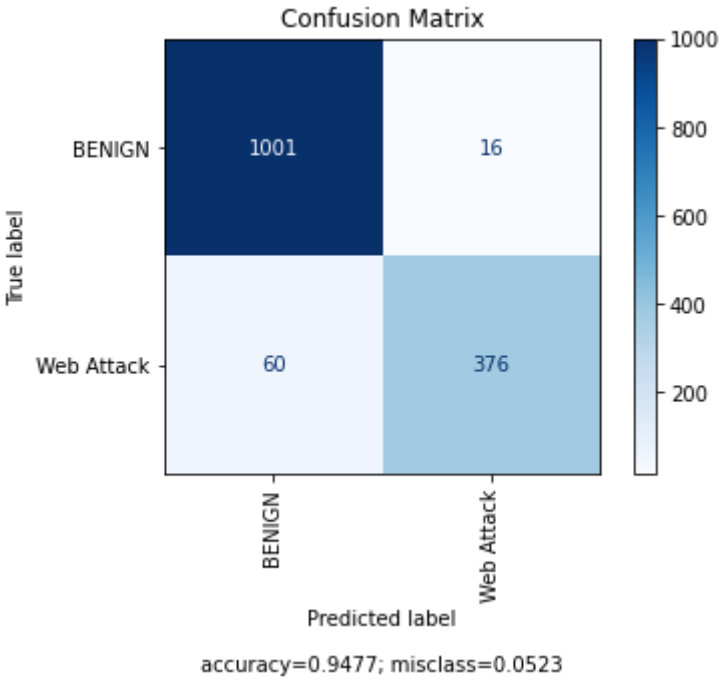

Рис. 3. Матрица ошибок модели CNN-BiLSTM
Fig. 3. Confusion matrix of the CNN-BiLSTM model

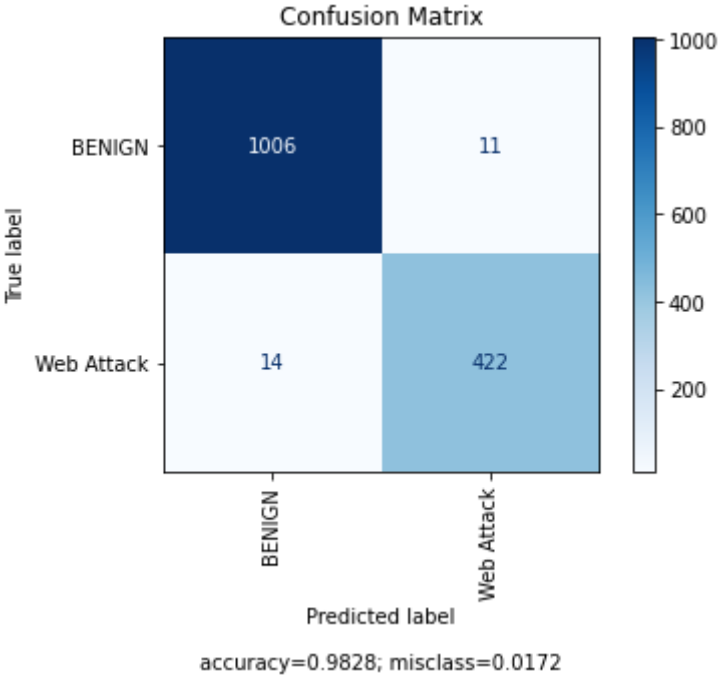

Рис. 4. Матрица ошибок модели Random Forest
Fig. 4. Confusion matrix of the Random Forest model

Стоит отметить, что подавляющее большинство реализованных исследователями методов глубокого обучения обнаруживают атаки на сетевом уровне, а методы обнаружения атак на прикладном уровне или уровне хоста изучены недостаточно.

Кроме того, не в каждом исследовании уделяется достаточное внимание вопросу выбора гиперпараметров. Либо подробные рассуждения просто оставались за рамками статей, либо им не всегда уделялось должное внимание.

Проблемы в основном наблюдаются с многоклассовой классификацией атак, что решается аккуратной подготовкой набора данных – балансировкой, тщательным отбором признаков. В целом, можно отметить наличие тех же проблем, которые имеются и у классических методов машинного обучения: устаревшие несбалансированные наборы данных, не соответствующие реальным данным, что делает сравнение методов проблематичным. Выбор различных метрик для оценки также не способствует качественному сравнению.

Использованное оборудование указано практически в каждом изученном исследовании, однако время обучения или применения оказалось указано только в части проанализированных исследований. Это привело к невозможности качественного сравнения рассмотренных архитектур по критерию времени. Также необходимо отметить, что отсутствие общедоступного кода в большинстве работ не позволяет однозначно верифицировать результаты экспериментов, изложенные в исследованиях.

Для оценки применимости использования методов глубокого обучения для обнаружения компьютерных атак была синтезирована нейронная сеть CNN-BiLSTM, обученная и протестированная на разработанной ранее сбалансированной и предварительно обработанной подвыборке веб-атак WebAttacks набора данных CICIDS2017. Исходный код проекта опубликован для общего доступа, что позволяет обеспечить верифицируемость результатов. При решении задачи бинарной классификации модель CNN-BiLSTM имеет достаточно высокое качество, однако по всем метрикам уступает разработанной ранее модели, использующей классификатор Random Forest. Несмотря на меньшую эффективность разработанной модели CNN-BiLSTM по сравнению с моделью Random Forest, использование метода глубокого обучения позволило исключить этап «ручного» конструирования признаков, что вместе с близостью полученных значений метрик для сравниваемых моделей подтверждает перспективность применения методов глубокого обучения для обнаружения вторжений. Для достижения необходимого уровня эффективности модели требуются проведение дальнейших исследований и настройка параметров нейронной сети для решения поставленной задачи.

7. Список сокращений

ОЗУ – оперативное запоминающее устройство

СОА – система обнаружения атак

ACC – Accuracy, доля правильных ответов

ADAM – Adaptive Moment Estimation, метод адаптивной оценки моментов

AE – Autoencoder, автокодировщик (автоэнкодер)

ANN – Artificial Neural Network, искусственная нейронная сеть

BGRU – Bidirectional GRU, двунаправленная GRU

BiLSTM – Bidirectional LSTM, двунаправленная LSTM

BN – Batch normalization, пакетная нормализация

BPTT – Backpropogation Through Time, метод обратного распространения ошибки во времени

CFS – Correlation Based Feature Selection, основанный на корреляции отбор признаков

CNN – Convolutional Neural Network, свёрточная нейронная сеть

CPU – Central Processing Unit, центральный процессор

CSV – Comma-Separated Values, разделённые запятыми значения

DBN – Deep Belief Network, глубокая сеть доверия
DL – Deep Learning, глубокое обучение
DNN – Deep Neural Network, глубокая нейронная сеть
D-ONN – Deep-Optimized Neural Network, глубоко оптимизированная нейронная сеть
DR – Detection Rate, частота обнаружения
DT – Decision Tree, дерево принятия решений
FAR – False Alarm Rate, уровень ложных срабатываний
FICA – Fast Independent Component Analysis, быстрый метод независимых компонент
FPR – False Positive Rate, доля ложноположительных примеров
GAN – Generative Adversarial Network, генеративно-состязательная сеть
GPU – Graphics Processing Unit, графический процессор
GRU – Gated Recurrent Unit, управляемый рекуррентный блок (нейрон)
HTTP – HyperText Transfer Protocol, протокол передачи гипертекста
IAT – Inter-Arrival Time, межпакетный интервал
IDS – Intrusion Detection System, система обнаружения вторжений
IF – Isolation Forest, изоляционный лес
IoT – Internet of Things, Интернет вещей
IP – Internet Protocol, Интернет-протокол
K-NN – k-Nearest Neighbors Algorithm, метод k-ближайших соседей
KPCA – Kernel Principal Component Analysis, ядерный метод главных компонент
LR – Logistic Regression, логистическая регрессия
LSTM – Long Short-Term Memory, длинная цепь элементов краткосрочной памяти
ML – Machine Learning, машинное обучение
MLP - Multilayer Perceptron, многослойный перцептрон
NB – Naive Bayes Classifier, Наивный байесовский классификатор
NB Tree – Naïve Bayes Tree, наивное байесовское дерево
OHE – One-Hot Encoding, быстрое кодирование
OSS – One-Sided Selection, односторонний сэмплинг
PCA – Principal Component Analysis, метод главных компонент
R2L – Remote to Local, атака типа Remote to Local
ReLU – Rectified Linear Unit, усеченное линейное преобразование
RF – Random Forest, случайный лес
RNN – Recurrent Neural Network, рекуррентная нейронная сеть
RTRL – Real-time recurrent learning, метод рекуррентного обучения в реальном времени
SAE – Stacked Autoencoder, многослойный автокодировщик
SFS – Sequence Forward Selection, последовательный прямой выбор
SGD – Stochastic Gradient Descent, стохастический градиентный спуск
SMOTE – Synthetic Minority Over-sampling Technique, техника передискретизации синтетического меньшинства
S-NN – Shallow Neural Network, неглубокая нейронная сеть
SQL – Structured Query Language, язык структурированных запросов
SVM – Support Vector Machine, метод опорных векторов
U2R – User to Root, атака типа User to Root
WAF – Web Application Firewall, файрвол веб-приложений
XSS – Cross-Site Scripting, межсайтовый скриптинг

Список литературы / References

[1]. Mohammadi S., Namadchian A. Anomaly-based Web Attack Detection: The Application of Deep Neural Network Seq2Seq With Attention Mechanism. The ISC International Journal of Information Security, vol. 12, issue 1, 2020, pp. 44-54. DOI: 10.22042/isecure.2020.199009.479.

[2]. Web attack detection using CNN-BiLSTM neural network and CICIDS2017 dataset. Доступно по ссылке: https://github.com/fisher85/ml-cybersecurity/blob/master/python-web-attack-detection/web-attack-detection-using-CNN-BiLSTM.ipynb, 04.10.2023.

[3]. Горюнов М.Н., Мацкевич А.Г., Рыболовлев Д.А. Синтез модели машинного обучения для обнаружения компьютерных атак на основе набора данных CICIDS2017. Труды ИСП РАН, том 32, вып. 5, 2020 г., стр. 81-94 / Goryunov M.N., Matskevich A.G., Rybolovlev D.A. Synthesis of a machine learning model for detecting computer attacks based on the CICIDS2017 dataset. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 5, 2020, pp. 81-94 (in Russian). DOI: 10.15514/ISPRAS–2020–32(5)–6.

[4]. Intrusion Detection Evaluation Dataset (CICIDS2017). Available at: https://www.unb.ca/cic/datasets/ids-2017.html, accessed 04.10.2023.

[5]. Гайфулина Д.А., Котенко И.В. Применение методов глубокого обучения в задачах кибербезопасности. Часть 1 // Вопросы кибербезопасности, вып. №3 (37), 2020 г., стр. 76-86. DOI: 10.21681/2311-3456-2020-03-76-86.

[6]. Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review, vol. 65, issue 6, 1958, pp. 386-408. DOI: 10.1037/H0042519.

[7]. Rumelhart D.E., Hinton G.E., Williams R.J. Learning Internal Representations by Error Propagation. In: Rumelhart, D.E. and McClelland, J.L., The PDP Group, Eds., Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1, Foundations, MIT Press, Cambridge, 1985, pp. 318-362.

[8]. Goodfellow I., Bengio Y., Courville A. Deep Learning. MIT Press, 2016. Available at: http://www.deeplearningbook.org.

[9]. Culurciello E. The fall of RNN / LSTM (2018). Available at: https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0.

[10]. Harzing A.W. Publish or Perish (2007). Available at: https://harzing.com/resources/publish-or-perish.

[11]. Google Scholar. Available at: https://scholar.google.com, accessed 04.10.2023.

[12]. Jiang K., Wang W., Wang A., Wu H. Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network. IEEE Access, vol. 8, 2020, pp. 32464-32476. DOI: 10.1109/ACCESS.2020.2973730.

[13]. Vigneswaran R.K., Vinayakumar R., Soman K.P., Poornachandran P. Evaluating Shallow and Deep Neural Networks for Network Intrusion Detection Systems in Cyber Security. 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2018, pp. 1-6. DOI: 10.1109/ICCCNT.2018.8494096.

[14]. Intrusion-Detection-Systems. Available at: https://github.com/rahulvigneswaran/Intrusion-Detection-Systems, accessed 04.10.2023.

[15]. Khan M.A. HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System. Processes, vol. 9, issue 5: 834, 2021, 14 p. DOI: 10.3390/pr9050834.

[16]. Hybrid-Convolutional-Recurrent-Neural-Network-Based-Network-IDS. Available at: https://github.com/Ashfaqjiskani/Hybrid-Convolutional-Recurrent-Neural-Network-Based-Network-IDS, accessed 04.10.2023.

[17]. Xiao Y., Xing C., Zhang T., Zhao Z. An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks. IEEE Access, vol. 7, 2019, pp. 42210-42219. DOI: 10.1109/ACCESS.2019.2904620.

[18]. Wang W., Sheng Y., Wang J., Zeng X., Ye X., Huang Y., Zhu M. HAST-IDS: Learning Hierarchical Spatial-Temporal Features Using Deep Neural Networks to Improve Intrusion Detection. IEEE Access, vol. 6, 2018, pp. 1792-1806. DOI: 10.1109/ACCESS.2017.2780250.

[19]. Li Z., Qin Z., Huang K., Yang X., Ye S. Intrusion Detection Using Convolutional Neural Networks for Representation Learning. In: Liu D., Xie S., Li Y., Zhao D., El-Alfy ES. (eds) Neural Information Processing. ICONIP 2017. Lecture Notes in Computer Science, vol. 10638. Springer, Cham, 2017, pp. 858-866. DOI: 10.1007/978-3-319-70139-4_87.

[20]. Vartouni A.M., Teshnehlab M., Kashi S.S. Leveraging Deep Neural Networks for Anomaly-Based Web Application Firewall. IET Information Security, vol. 13, issue 4, 2019, pp. 352-361. DOI: 10.1049/iet-ifs.2018.5404.

[21]. Yin C., Zhu Y., Fei J., He X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. IEEE Access, vol. 5, 2017, pp. 21954-21961. DOI: 10.1109/ACCESS.2017.2762418.
[22]. Sheikhan M., Jadidi Z., Farrokhi A. Intrusion detection using reduced-size RNN based on feature grouping. Neural Computing and Applications - NCA, vol. 21, no. 6, 2012, pp. 1185–1190. DOI: 10.1007/s00521-010-0487-0.
[23]. Vinayakumar R., Soman K.P., Poornachandran P. Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). International Journal of Information System Modeling and Design, vol. 8, no. 3, 2017, pp. 43-63. DOI: 10.4018/IJISMD.2017070103.
[24]. Sumaiya Thaseen I., Saira Banu J., Lavanya K., Rukunuddin Ghalib M., Abhishek K. An integrated intrusion detection system using correlation-based attribute selection and artificial neural network. Transactions on Emerging Telecommunications Technologies, vol. 32, issue 2: e4014, 2021, 15 p. DOI: 10.1002/ett.4014.
[25]. Ramaiah M., Chandrasekaran V., Ravi V., Kumar N. An intrusion detection system using optimized deep neural network architecture. Transactions on Emerging Telecommunications Technologies, vol. 32, issue 4: e4221, 2021, 17 p. DOI: 10.1002/ett.4221.
[26]. Xu C., Shen J., Du X., Zhang F. An Intrusion Detection System Using a Deep Neural Network With Gated Recurrent Units. IEEE Access, vol. 6, 2018, pp. 48697-48707. DOI: 10.1109/ACCESS.2018.2867564.
[27]. Le T.-T.-H., Kim Y., Kim H. Network Intrusion Detection Based on Novel Feature Selection Model and Various Recurrent Neural Networks. Applied Sciences, vol. 9, no. 7: 1392, 2019, 29 p. DOI: 10.3390/app9071392.
[28]. Drewek-Ossowicka A., Pietrołaj M., Rumiński J. A survey of neural networks usage for intrusion detection systems. Journal of Ambient Intelligence and Humanized Computing, vol. 12, 2021, pp. 497–514. DOI: 10.1007/s12652-020-02014-x.
[29]. Liu H., Lang B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. Applied Sciences, vol. 9, no. 20: 4396, 2019, 28 p. DOI: 10.3390/app9204396.
[30]. Гайфулина Д.А., Котенко И.В. Применение методов глубокого обучения в задачах кибербезопасности. Часть 2 // Вопросы кибербезопасности, вып. №4 (38), 2020 г., стр. 11-21. DOI: 10.21681/2311-3456-2020-04-11-21
[31]. Sinha J., Manollas M. Efficient Deep CNN-BiLSTM Model for Network Intrusion Detection. Proceedings of the 2020 3rd International Conference on Artificial Intelligence and Pattern Recognition (AIPR 2020), Association for Computing Machinery, New York, NY, USA, 2020, pp. 223–231. DOI: 10.1145/3430199.3430224.
[32]. Efficient-CNN-BiLSTM-for-Network-IDS. Available at: https://github.com/razor08/Efficient-CNN-BiLSTM-for-Network-IDS/blob/master/NSL_KDD_Final.ipynb, accessed 04.10.2023.
[33]. Kostas K. Anomaly Detection in Networks Using Machine Learning. Master's Thesis. University of Essex, 2018, 70 p.
[34]. Kingma D.P., Ba J. Adam: A Method for Stochastic Optimization. The International Conference on Learning Representations (ICLR), San Diego, 2015, 15 p. DOI: 10.48550/arXiv.1412.6980.
[35]. Web attack detection using CICIDS2017 dataset. Доступно по ссылке: https://github.com/fisher85/ml-cybersecurity/blob/master/python-web-attack-detection/web-attack-detection.ipynb, 04.10.2023.

Информация об авторах / Information about authors

Александр Игоревич ГЕТЬМАН – кандидат физико-математических наук, старший научный сотрудник ИСП РАН, доцент ВШЭ. Сфера научных интересов: анализ бинарного кода, восстановление форматов данных, анализ и классификация сетевого трафика.

Aleksandr Igorevich GETMAN – Cand. Sci. (Phys.-Math.), senior researcher at ISP RAS, associate professor at HSE. Research interests: binary code analysis, data format recovery, network traffic analysis and classification.

Максим Николаевич ГОРЮНОВ – кандидат технических наук. Сфера научных интересов: информационная безопасность, системы обнаружения вторжений, системы анализа защищенности, машинное обучение, безопасная разработка программного обеспечения.

Maxim Nikolaevich GORYUNOV – Cand. Sci. (Tech.). Research interests: information security, intrusion detection systems, security analysis systems, machine learning.

Андрей Георгиевич МАЦКЕВИЧ – кандидат технических наук, доцент. Сфера научных интересов: информационная безопасность, системы обнаружения вторжений, системы антивирусной защиты, машинное обучение, криптографические методы защиты информации.

Andrey Georgievich MATSKEVICH – Cand. Sci. (Tech.), associate professor. Research interests: information security, intrusion detection systems, anti-virus protection systems, machine learning, cryptographic methods for protecting information.

Дмитрий Александрович РЫБОЛОВЛЕВ – кандидат технических наук. Сфера научных интересов: информационная безопасность, системы обнаружения вторжений, машинное обучение, криптографические методы защиты информации.

Dmitry Aleksandrovich RYBOLOVLEV – Cand. Sci. (Tech.). Research interests: information security, intrusion detection systems, machine learning, cryptographic methods for protecting information.

Анастасия Григорьевна НИКОЛЬСКАЯ - Сфера научных интересов: информационная безопасность, системы обнаружения вторжений, машинное обучение, искусственные нейронные сети.

Anastasiya Grigorevna NIKOLSKAYA. Research interests: information security, intrusion detection systems, machine learning, artificial neural networks.