ПРИМЕНЕНИЕ МЕТОДОВ ГЛУБОКОГО ОБУЧЕНИЯ В ЗАДАЧАХ КИБЕРБЕЗОПАСНОСТИ. ЧАСТЬ 2

Гайфулина Д.А.[1], Котенко И.В.[2]

Цель статьи: сравнительный анализ методов решения различных задач кибербезопасности, основанных на использовании алгоритмов глубокого обучения.

Метод исследования: *Системный анализ современных методов глубокого обучения в задачах кибербезопасности в различных приложениях кибербезопасности, включая обнаружение вторжений и вредоносного программного обеспечения, анализ сетевого трафика и некоторые другие задачи.*

Полученный результат: *Предложена схема классификации рассмотренных подходов к глубокому обучению в кибербезопасности, а также представлена их сравнительная характеристика по используемым моделям, характеристикам и наборам данных. Проведенный анализ показал, что архитектуры с большим количеством нейронов на каждом слое показывают лучшие результаты. Даны рекомендации по использованию методов глубокого обучения в приложениях кибербезопасности.*

Основной вклад авторов в область исследования методов глубокого обучения для задач кибербезопаности заключается в классификации предметной области, проведении общего и сравнительного анализа существующих подходов, отражающих современное состояние научных исследований.

Настоящая статья является продолжением первой части, опубликованной в журнале «Вопросы кибербезопасности» №3_2020 [1].

Ключевые слова: *наука о данных, машинное обучение, глубокие нейронные сети, обнаружение вторжений, обнаружение вредоносного программного обеспечения.*

DOI: 10.21681/2311-3456-2020-04-11-21

Введение

В первой части статьи [1] было представлено краткое описание методов глубокого обучения и их классификация по используемой архитектуре. Определен понятийный аппарат исследования, его методологическая основа и актуальность. Также в первой части была дана сравнительная характеристика опубликованных аналитических обзоров в области применения методов глубокого обучения в различных приложениях кибербезопасности.

В данной части статьи описываются методы глубокого обучения и возможности их использования для обеспечения безопасности информационно-коммуникационных систем. Основной вклад авторов заключается в классификации предметной области, проведении общего и сравнительного анализа существующих подходов, отражающих современное состояние научных исследований. Статья имеет следующую структуру. В *разделе 1* рассматривается применение различных методов глубокого обучения в зависимости от конкретного приложения кибербезопасности. В *разделе 2* проводится анализ рассмотренных методов глубокого обучения.

1. Применение глубокого обучения для различных задач кибербезопасности

В данном разделе подходы к обеспечению кибербезопасности сгруппированы в соответствии с поставленными задачами, свойствами исходных данных, а также используемыми методами глубокого обучения.

1.1. Обнаружение вторжений

Автокодировщики. Методы обнаружения вторжений с использованием автокодировщиков и ограниченной машины Больцмана (Restricted Boltzmann Machine, RBM) основаны на реконструкции данных, при которой определяется величина различия нормальных и аномальных данных. В статье [2] авторы представили решение AutoIDS, использующее в качестве детекторов два автокодировщика: второй детектор используется для сложных образцов, в которых первый не уверен. Сбой работы нейронных сетей при обработке потока входящих пакетов означает, что такой поток не соответствует нормальному трафику и рассматривается как вторжение. Оценка AutoIDS на наборе данных NSL-KDD [3] показала точность 90,17%. В [4] предлагается метод

1 Гайфулина Диана Альбертовна, аспирант, младший научный сотрудник лаборатории проблем компьютерной безопасности, ФГБУН Санкт-Петербургский институт информатики и автоматизации Российской академии наук, г. Санкт-Петербург, Россия. E-mail: gaifulina@comsec.spb.ru

2 Котенко Игорь Витальевич, доктор технических наук, профессор, главный научный сотрудник и заведующий лабораторией проблем компьютерной безопасности, ФГБУН Санкт-Петербургский институт информатики и автоматизации Российской академии наук, г. Санкт-Петербург, Россия. E-mail: ivkote@comsec.spb.ru

обнаружения сетевых аномалий, основанный на применении шумоподавляющего автокодировщика и метода прореживания (dropout) для предотвращения переобучения. Также системы обнаружения вторжений, управляемые автокодировщиками, представлены в [5-7].

Ограниченные машины Больцмана. Работа [8] посвящена определению DDoS-атак с использованием классификатора RBM. Отбор признаков производится с помощью модели оптимизации поиска с произвольной гармонией (Random Harmony Search, RHS) [9]. Эксперименты на наборе данных KDD Cup 99 [10] показали, что модель RHS-RBM достигает точности 99,92%, а F-мера – 99,93%. Исследование [11] посвящено использованию RBM для обнаружения кибертатак в мобильной облачной среде с точностью 97,11%.

Реккурентные нейронные сети. Отличительной особенностью рекуррентных нейронных сетей (Recurrent Neural Network, RNN) является наличие обратной связи, что позволяет им анализировать последовательные данные, такие как временные ряды. Анализируя последовательность измерений различных параметров текущего процесса, сеть обучается предсказывать его состояние в следующий момент времени. Если предсказанное RNN состояние отличается от текущего, регистрируется аномалия. В [12] RNN применяется для бинарной и мультиклассовой классификации наборов сетевых данных NSL-KDD. Недостатком стандартных RNN являются проблемы с исчезновением градиента и нехватка памяти для использования информации за предыдущие моменты времени. В [13] предлагается метод обнаружения вторжений, основанный на долговременной кратковременной памяти (Long Short-Term Memory, LSTM), который позволяет справляться с данными проблемами, но требует большего количества обучающих параметров. В статье [14] используется RNN с управляемым рекуррентным блоком (Gated Recurrent Units, GRU), которая требует меньше параметров для обучения. Эксперименты на наборе данных KDD Cup 99 показывают точность обнаружения 99,91%.

Сверточные нейронные сети. Сверточные нейронные сети (Convolutional Neural Network, CNN) нацелены на эффективное распознавание образов, что также позволяет использовать их для выявления вторжений. В работе [15] представлен метод классификации вредоносного сетевого трафика в программно-определяемых сетях (Software Defined Networks, SDN) на основе сверточной и реккурентной нейронных сетей. Заголовки пакетов кодируются в двумерную матрицу, которая используется для обучения CNN, на выходе которой применяется RNN в качестве финального классификатора. Оценка подхода проводилась на наборе данных, сгенерированном в симуляторе SDN, и наборе данных CTU-13. Наилучшая точность модели может достигать 99,86% для CTU-13 и 99,84% для сгенерированных данных.

Генеративно-состязательные сети. Одним из подходов к использованию генеративно-состязательных сетей (Generative Adversarial Networks, GAN) является непрерывная генерация поддельных сетевых данных, с целью повышения производительности исходной модели обнаружения. Генератор создает новые экземпляры данных из некоторого скрытого пространства, и дискриминатор, оценивает их на подлинность: относится ли каждый новый экземпляр данных к набору тренировочных данных или нет. Таким образом, GAN могут быть использованы для исследования распределения нормальных данных, чтобы распознавать аномалии по неизвестным данным. Так в [16] данная модель применялась для экспериментального исследования набора данных ботнетов ISCX. В [17] для обнаружения аномалий применяется модель на основе двунаправленной GAN (BiGAN), которая дополнительно проводит обратное отображение реальных данных в скрытое пространство. Помимо экономии времени, BiGAN способствует более эффективному извлечению признаков сетевого трафика. Данная модель на наборе данных KDD Cup 99 показала точность 93,24%.

1.2. Обнаружение вредоносного программного обеспечения

Глубокие сети доверия. На основе использования заражённых вирусами файлов производится обучение глубоких нейросетей для обнаружения вредоносных программ. В рамках этой задачи работа [18] описывает методы применения автокодировщиков, которые проверяются с использованием образцов вредоносного ПО VirusShare и сертифицированного ПО Windows 10, демонстрируя точность обнаружения до 97,5%. Авторы статьи [19] предложили подход к обнаружению на основе совместного использования автокодировщика и RBM и достигли точности 98,82%, используя данные Comodo Cloud Security Center [20]. В [21] обнаружение вредоносных программ осуществляется для Android-платформ с применением автокодировщика.

Реккурентные нейронные сети. RNN применяются для обнаружения вредоносных программ, анализируя последовательности системных вызовов или заголовки исполняемых файлов. В [22] две модели LSTM обучаются последовательностям системных вызовов от вредоносных программ и от сертифицированных приложений, а затем вычисляются две оценки сходства. Классификатор определяет, является ли анализируемое приложение вредоносным или заслуживающим доверия по большей оценке. Эксперименты с применением набора данных Drebin [23] демонстрируют точность 93,7%. Используя последовательные динамические данные, авторы работы [24] применили RNN для прогнозирования вредоносных программ, исполняемых после запуска. Активность компьютера отслеживается с использованием вирусных данных VirusTotal, и точность их обнаружения достигает 94%. В статье [25] авторы оценивают эффективность использования LSTM на основе внимания (Attention-based LSTM, ATT-LSTM) для классификации данных кодов операций и системных вызовов. Точность классификации составляет 95% при статическом анализе и 99% при динамическом анализе.

Сверточные нейронные сети. Необработанные последовательности системных вызовов формализуются как последовательность векторов с фиксированным

размером, что является входными данными CNN. Авторы статьи [26] предлагают подход к обнаружению вредоносных приложений Android на основе статистического анализа и тестируют его на нескольких наборах данных, включая MalGenome [27], Drebin и MalDozer [28], с показателями F-меры до 99%. В [29] предлагается подход VMAnalyzer, который извлекает упорядоченную последовательность системных вызовов отслеживаемых программ и выполняет двухуровневую классификацию. На первом уровне для извлечения и выбора соответствующих последовательностей системных вызовов используется CNN. На втором уровне применяется двунаправленная LSTM для анализа и обнаружения вредоносных последовательностей. Результаты оценки данного подхода на наборах данных University of New Mexico (UNM) [30] показывают точность обнаружения от 81% до 96,67%.

Генеративно-состязательные сети. В исследованиях, посвященных обнаружению вредоносной активности, генеративно-состязательные сети используются, как правило, для генерации сигнатур вредоносного ПО. В [31] предлагается архитектура глубоких сверточных генеративно-состязательных сетей (Deep Convolutional Generative Adversarial Networks, DCGAN) для исследования динамического поведения приложений Android. Предлагаемый в [32] подход к обнаружению вредоносных программ для Android также основан на GAN и применении методов теории игр двух игроков для решения проблемы «бумажных ножниц» с точностью 99,1% и F-мерой 99%.

Рекурсивная нейронная сеть. Авторы работы [33] предлагают систему для сбора HTTP-запросов с динамическим анализом вредоносных программ, применяя рекурсивную нейронную сеть (Recursive Neural Network, ReNN). Данная система в течение короткого периода времени анализирует образец ПО, основываясь на том, что вредоносные коммуникации напоминают естественный язык с точки зрения структуры данных. В экспериментах применялось 42 856 образцов вредоносных программ из набора данных VirusTotal [34]. Точность обнаружения вредоносных программ составила 97,7%.

1.3. Анализ сетевого трафика

Автокодировщик. Сверточная нейронная сеть. Основные проблемы при анализе сетевого трафика связаны с извлечением признаков из потоковых данных и работой с неизвестными сетевыми протоколами. В [35] представлен подход к классификации зашифрованного трафика с применением многослойного персептрона, многоуровневого автокодировщика и CNN. Для экспериментальной оценки использовался набор данных трафика VPN-nonVPN ISCX2012 [36]. В [37] авторы предлагают подход DeepPacket, который может классифицировать сетевой трафик как по основным классам (например, FTP и P2P), так и по приложениям конечного пользователя (например, BitTorrent и Skype). Данный инструмент с CNN в качестве модели классификации достигает точности 98% в задаче идентификации приложения и 94% в задаче категоризации трафика.

Нейронные сети прямого распространения. В работе [38] исследуется возможность применения алгоритмов глубокого обучения для классификации сетевого трафика и использование их для идентификации вредоносного трафика и сетевого управления. Для анализа трафика использовалась нейронная сеть прямого распространения (Feed Forward Neural Network, FFNN). Для тестирования подхода использовался набор данных UNSW-NB15 [39]. Итоговая точность классификации для большинства протоколов превысила 97%.

1.4. Раскрытие утечек данных

Реккурентные нейронные сети. В работе [40] представлен подход к обнаружению внутренних нарушителей путем анализа журнала системных событий. Для выявления шаблонов нормального поведения используется сеть LSTM, что позволило зафиксировать временные шаблоны в поведении пользователей. Для поиска аномалий применяется регрессионная модель, предсказывающая поведение в определенный момент времени на основе вероятностных распределений, соответствующих предшествующим наблюдениям. Подход тестируется на наборе данных CERT v6.2 [41], и полнота обнаружения аномалий составляет около 99%. В [42] RNN используется для классификации атрибутов пользователя. Для расчета отклонений в поведении на основе результатов нескольких классификаторов введен калькулятор аномалий. Результаты эксперимента показывают точность классификации, достигающую 96,97%.

1.5. Выявление спама

Глубокая сеть доверия. Для повышения эффективности классификации веб-спама в [43] используются сети глубокого доверия (Deep Belief Networks, DBN), которые эффективно комбинируются с алгоритмом синтетической избыточной выборки меньшинств (Synthetic Minority Over-Sampling Technique, SMOTE) и алгоритмом подавления помех с помощью автокодировщика. После многочисленных экспериментов с набором данных WEBSPAM-UK2007 [44] результаты показывают, что точность обнаружения предложенного метода классификации, достигает показателя площади под ROC-кривой (AUC) в 98%, при этом F-мера составляет 96,66%.

Реккурентная нейронная сеть. В [45] представлена архитектура рекуррентной нейронной сети с LSTM для классификации спама. Перед использованием LSTM для задачи классификации текст преобразуется в векторы семантических слов с помощью word2vec, WordNet и ConceptNet. Для экспериментальной оценки применялись наборы данных SMS-спама, а также социальной сети Twitter. Точность обнаружения спама при использовании предлагаемого подхода достигает 99%.

1.6. Выявление вредоносных сайтов

Сверточная нейронная сеть. В работе [46] предложен подход к обнаружению вредоносных URL-адресов, путей к файлам и ключей реестра на основе CNN, принимающей на вход необработанные последовательности символов, извлекающей из них признаки и

классифицирующей, являются ли они индикаторами злонамеренного поведения. Последовательность входных символов кодируется в виде двумерного тензора, который применяется компонентом обнаружения признаков для обнаружения шаблонов подпоследовательностей в общих строках и агрегации полученной информации в векторы фиксированной длины с помощью сверточной нейросети. Последний компонент выполняет классификацию полученных признаков. Результаты тестирования предложенного подхода на размеченном наборе данных показали высокую точность классификации, площадь под ROC-кривой (AUC) для каждой из категорий исходных данных достигла 98%. В работе [47] представлена распределенная система обнаружения веб-атак по URL-адресам, использующая CNN. Данная система предназначена для защиты нескольких веб-приложений в распределенной среде Edge of Things (EoT). Эксперименты проводятся в системе с двумя параллельными глубокими моделями сверточных нейронных сетей на наборах данных: HTTP Dataset CSIC 2010 [48] и FWAF [49], демонстрируя результаты с точностью 99,41%.

2. Анализ методов глубокого обучения, используемых в кибербезопасности

В данном разделе приводится сравнительная характеристика рассмотренных подходов. В табл. 1 показаны следующие характеристики используемых методов глубокого обучения (ГО):

- архитектура сети – количество слоев и нейронов в каждом из них (*x* – входной слой, *y* – выходной слой, *h* – скрытый слой, *p* – субдискретизирующий слой, *c* – сверточный слой, *n* – полносвязный слой);
- функция активации (sigm – сигмоида, htang – гиперболический тангенс);
- используемые наборы данных;
- итоговое качество работы предложенного подхода (ACC – аккуратность, F1– F мера, REC – полнота, AUC – площадь под ROC-кривой).

По результатам, представленным в таблице, можно сделать вывод, что большинство подходов дают хорошие результаты для решения задач безопасности, позволяющие применять их на практике. Для отбора признаков чаще всего используются многоуровневые автокодировщики и реккуретные нейронные сети, а для классификации LSTM-RNN. Для активации используются функция сигмоиды, часто комбинируемая с ReLU активацией на предыдущих слоях.

Большое количество реализаций автокодировщиков и RBM, которые являются методами глубокого обучения без учителя, вероятно, связано со сложностью получения размеченных данных. Использование методов глубокого обучения без учителя является одним из способов справиться с анализом большого объема немаркированных данных в области кибербезопасности. Основным недостатком обучения с учителем является то, что оно требует наличия большого количества размеченных данных, а их сбор является трудозатратным. Чтобы обойти эту проблему, большинство архитектур глубоких нейронных сетей используют две фазы обучения: первичную настройку слоев нейросети без учителя, а затем более тонкую настройку на небольшом размеченном наборе данных. Неконтролируемая предварительная подготовка может помочь исследователям и разработчикам в определении начальных весов и смещений.

Частое применение рекуррентных нейронных сетей объясняется тем, что многие данные, связанные с кибербезопасностью, могут быть представлены в виде временных рядов. К ним относятся сетевой трафик, журналы событий, последовательности системных вызовов и т.п. RNN хорошо зарекомендовали себя в обработке последовательных данных, по которым предсказывают состояние процесса в следующий момент времени.

Оптимальное число скрытых слоев и нейронов может быть получено экспериментальным путем, который заключается в нахождении логических соотношений между числом нейронов в каждом слое и размером набора данных. Подход [16], вероятно, показал низкое качество обнаружения из-за использования малого количества признаков по сравнению с остальными подходами. Подходы [5, 6, 12] имеют меньшее качество по сравнению с остальными, что можно связать с использованием недостаточно глубоких архитектур. Для обнаружения вторжений лучшие результаты показывают глубокие сети доверия с большим количеством нейронов [9] и гибридные методы обучения с использованием реккурентных нейронных сетей [14, 15]. Для обнаружений вредоносного ПО – сверточные нейронные сети [26] и генеративно-состязательные [32]. Для прочих приложений кибербезопасности также можно отметить, что сети с большим количеством нейронов на каждом слое показывает большую точность классификации.

С увеличением числа слоев в глубоких нейронных сетях ошибки обучения возрастают, а производительность сети ухудшается. Это указывает на то, что в современных методах обучения нейронной сети с большим числом слоев по-прежнему существует множество проблем.

Можно сделать вывод, что методы глубокого обучения предпочтительно использовать в задачах, где необходимо изучать сложные нелинейные гипотезы с множеством признаков данных, а также в областях, в которых проводится обработка данных в больших объемах. Для линейных задач с малыми объемами данных достаточно применения алгоритмов классического машинного обучения, таких как линейная регрессия или метод опорных векторов, которые обеспечат необходимый исследователю результат. Прежде всего, глубокое обучение представляет собой модель машинного обучения, которая помогает неявно выделить иерархическое и абстрактное представление признаков, в чем оно превосходит прочие алгоритмы машинного обучения. Таким образом, рекомендуется применение глубокого обучения в задачах кибербезопасности с большим количеством необработанных данных, тем самым позволяя избежать предварительной обработки подобной информации.

DOI:10.21681/2311-3456-2020-04-11-21

Таблица 1

Анализ методов глубокого обучения в области кибербезопасности

Задача безопасности	Авторы, год, ссылка	Метод ГО		Архитектура сети (число нейронов)	Функция активации	Набор данных	Оценка
		извлечение признаков	классификация				
Обнаружение вторжений	Gharib et al, 2019 [2]	AE		*h*(140)-*h*(80)	ReLu, sigm	NSL-KDD	ACC=90.17%
	Mohamed et al, 2019 [4]	DAE		*x*(122)-*h*(8)-*y*(122)	ReLu	NSL-KDD	ACC=90.32%
	Ieracitano et al, 2019 [5]	AE		*x*(102)-*h* (50) – *y*(102)	softmax	NSL-KDD	ACC=87%
	Yang Y. et al, 2019 [6]	AE		*x*(122)-*h*(80)-*h*(40)- *h*(20)-*h*(10)- *h*(5)	L2, ReLu	NSL-KDD, UNSW-NB15	ACC=85.97%
	Farahnakian& Heik-konen, 2018 [7]	AE		*h*(32)- *h*(32)-*h*(32)- *h*(32)-	sigm	KDD Cup 1999	ACC=94,71% DR=94.53%
	Mayuranathan et al, 2019 [8]	RHS	RBM	9*h*(1000)	sigm	KDD Cup 1999	ACC=99.92% F1=99.93%
	Nguyen et al., 2018 [11]	RBM	Sm Regr	2*h*(1000)	sigm	NSL-KDD, KDD Cup 1999, UNSW-NB15	ACC=97.11%
	Yin C. et al. 2017 [12]	RNN	Sm Regr	*h*(80)	sigm	NSL-KDD	ACC=83.3%/81,3%
	Zhu M. et al, 2018 [13]	RNN-LSTM		*h*(256)	L2, tangh	CICIDS2017	ACC=91%
	Manavi&Zhang, 2019 [14]	RNN-GRU	genetic algorithm	*h*(?)	dropout	KDD Cup 1999	ACC=99.91%
	QinY. et al, 2019 [15]	CNN+RNN	linear classifier	L2	4*c*-4*p*-4*n*-*h* (300)	Собственный, CTU-13	ACC=99,86%
	Yin C. et al., 2018 [16]	GAN		LSTM(*x*(120)-*h*(80)-*y*(122))+ DNN(*h*(122)-*h*(80)-*h*(20)-*y*(3))	softmax	ISCX botnet dataset	ACC=71,17% F1=70,59%
	Chen&Jiang, 2019 [17]	GAN		discriminator, encoder and generator	L1	KDD Cup 1999	ACC=93,24%

Задача безопасности	Авторы, год, ссылка	Метод ГО: извлечение признаков	Метод ГО: классификация	Архитектура сети (число нейронов)	Функция активации	Набор данных	Оценка
Обнаружение вредоносного ПО	De Paola et al., 2018 [18]	SDAE		*x*(1024)-*h*(1000)-*h*(1000)-*y*(1024)	PReLU, dropout	VirusShare	ACC=97.49%
	Ye Y. et al, 2018 [19]	AE+RBM		*h*(1000)-*h*(100)-*h*(100)	sigm	Comodo Cloud Security Center	ACC=98,82%
	Naway&Li, 2019 [21]	AE		*x*(40)-*h*(200)-*h*(100)-*h*(100)-*y*(40)	sigm	Contagiodump, droidbench, VirusShare, Android malware	ACC=96,81% F1=95,4%
	Xiao X. et al, 2019 [22]	RNN-LSTM		*h*(1000)-*h*(1000)-*h*(1000)-*h*(1000)	sigm	Drebin	ACC=93,7%
	Rhode et al., 2018 [24]	RNN-GRU		*h*(75)	L2	VirusTotal	ACC=94%
	Darabian et al., 2020 [25]	RNN-LSTM		*h*(100)	softmax	VirusTotal	ACC=95%/99%
	Karbab et al., 2018 [26]	CNN		1*c*-1*p*-1*n*	ReLU	MalGenome, Drebin, MalDozer	F1=99%
	Mishra et al., 2019 [29]	CNN	RNN-LSTM	*h*(?)-*h*(?)	sigm	University of New Maxico	ACC=96.97%
	Jan S. et al., 2018 [31]	GAN (DCGAN)		conv discriminator, conv generator	ReLU	Intents Dataset (собственный)	ACC=98.8% F1=99.6%
	Amin M. et al., 2019 [32]	GAN		*h*(?)-*h*(?)	ReLU	Drebin, Android Malware Dataset	ACC=99.1% F1=99%
	Shibahara T. et al., 2019 [33]	ReNN		3*h*(?)	htang	VirusTotal	ACC=97.7% F1=68.3%

Задача безопасности	Авторы, год, ссылка	Метод ГО		Архитектура сети (число нейронов)	Функция активации	Набор данных	Оценка
		извлечение признаков	классификация				
Анализ сетевого трафика	Wang P. et al., 2018 [35]	SAE		*x*(1480)-*h*(740)-*h* (92)-*y*(32)	ReLu, softmax	VPN-nonVPN ISCX2012	REC=99.14% F1=99.05%
	Wang P. et al., 2018 [35]	CNN		3*c*-2*p*-1*n*	ReLu, softmax	VPN-nonVPN ISCX2012	REC=99.3%, F1=98.91%
	Lotfollahi et al., 2020 [37]	SAE		*x*(400)-*h*(300)-*h*(200)-*y*(50)	softmax	VPN-nonVPN ISCX2012	REC=92%, F1=92%
	Lotfollahi et al., 2020 [37]	CNN		2*c*-1*p*-3*n*	softmax	VPN-nonVPN ISCX2012	REC=94%, F1=93%
	Smit et al, 2017 [38]	FFNN		*h*(1600)-*h*(1600)	ReLu, softmax	UNSW-NB15	ACC=97%
Раскрытие утечек данных	Tuor et al, 2017 [40]	RNN-LSTM		7*h*(1000)	htan	CERT insider threat dataset v6.1	REC=99%
	Meng F. et al., 2018 [42]	RNN-LSTM		5*h*(?)	sigm	CERT insider threat dataset v6.2	ACC=93.85%
Выявление спама	Li et al., 2018 [43]	AE+RBM		*h*(64)-*h*(64))-*h*(64)	sigm	WEBSPAM-UK2007	AUC=97.42%
	Jain et al., 2019 [45]	word2vec	RNN-LSTM	*h*(100)	sigm	SMS spam collection dataset, Twitter dataset (собственный)	ACC=99.01%, F1=99.24%
Выявление вредоносных сайтов	Saxe&Berlin, 2017 [46]	CNN		*h*(1024)- *h*(1024)- *h*(1024)- *h*(1024)	ReLU, sigm	Собственный(18 млн.)	AUC=98%, F1 = 96.66%
	Tian et al, 2019 [47]	CNN		4*c*-4*p*-3*n*	dropout	HTTP Dataset, CSIC 2010, FWAF	ACC=99,41%

DOI:10.21681/2311-3456-2020-04-11-21

Заключение

В данной работе был проведен сравнительный анализ методов решения различных задач кибербезопасности, основанных на использовании алгоритмов глубокого обучения. Большинство рассмотренных подходов применяется научным сообществом для создания систем, предотвращающих различные угрозы кибербезопасности. Предложена схема классификации рассмотренных подходов, а также проведен их сравнительный анализ, который показал, что многие более глубокие архитектуры с большим количеством нейронов на каждом слое показывают лучшие результаты. В соответствии с этим выводом, в качестве основного направления будущего развития глубоких нейронных сетей можно отметить увеличение размера сетей, то есть количества нейронов на каждом слое, и глубины сетей. Данный аспект требует мощных вычислительных ресурсов и может привести к риску переобучения. Еще одним направлением развития является использование информации из разных источников и комбинация разных типов признаков, как статических, так и динамических.

Рецензент: *Молдовян Александр Андреевич, доктор технических наук, профессор, главный научный сотрудник лаборатории кибербезопасности и постквантовых систем ФГБУН Санкт-Петербургский институт информатики и автоматизации Российской академии наук, Санкт-Петербург, Россия. E-mail: maal305@yandex.ru*

Работа выполнена при частичной финансовой поддержке РФФИ (проект 18-29-22034) и бюджетной темы 0073-2019-0002.

Литература

1. Гайфулина Д.А., Котенко И.В. Применение методов глубокого обучения в задачах кибербезопасности. Часть 1 // Вопросы кибербезопасности, № 3, 2020. С. 76-86.
2. Gharib M., Mohammadi B., Dastgerdi S.H., Sabokrou M. AutoIDS: Auto-encoder Based Method for Intrusion Detection System // arXiv preprint arXiv:1911.03306. 2019. P. 1-9.
3. Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. A detailed analysis of the KDD CUP 99 data set // Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada, 8–10 July 2009. P. 1-6.
4. Mohamed S., Ejbali R., Zaied M. Denoising Autoencoder with Dropout based Network Anomaly Detection // The Fourteenth International Conference on Software Engineering Advances (ICSEA), 2019. P. 98-104.
5. Ieracitano C., Adeel A., Morabito F.C., Hussain A.A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach // Neurocomputing. 2019. P. 1-12.
6. Yang Y., Zheng K., Wu C., Yang Y. Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network // Sensors. 2019. Vol. 19. No. 11. P. 2528.
7. Farahnakian F., Heikkonen J. A deep auto-encoder based approach for intrusion detection system // 20th International Conference on Advanced Communication Technology (ICACT). IEEE, 2018. P. 178-183.
8. Mayuranathan M., Murugan M., Dhanakoti V. Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment // Journal of Ambient Intelligence and Humanized Computing. 2019. P. 1-11.
9. Geem Z.W., Kim J.H., Loganathan G.V. A new heuristic optimization algorithm: harmony search // Simulation. 2001. Vol. 76. No. 2. P. 60-68.
10. KDD Cup 1999 Data. Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed April 29, 2020).
11. Nguyen K.K., Hoang D.T., Niyato D., Wang P., Nguyen D., Dutkiewicz E. Cyberattack Detection in Mobile Cloud Computing: A Deep Learning Approach // 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2018. P. 1-6.
12. Yin C., Zhu Y., Fei J., He X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks // IEEE Access, 2017. Vol. 5. P. 21954-21961.
13. Zhu M., Ye K., Wang Y., Xu C.Z. A Deep Learning Approach for Network Anomaly Detection Based on AMF-LSTM // IFIP International Conference on Network and Parallel Computing Springer, Cham, 2018. P. 137-141.
14. Manavi M., Zhang Y. A New Intrusion Detection System Based on Gated Recurrent Unit (GRU) and Genetic Algorithm // International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Springer, Cham, 2019. P. 368-383.
15. Qin Y., Wei J., Yang W. Deep Learning Based Anomaly Detection Scheme in Software-Defined Networking // 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, 2019. P. 1-4.
16. Yin C., Zhu Y., Liu S., Fei J., Zhang H. An Enhancing Framework for Botnet Detection Using Generative Adversarial Networks // 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD). IEEE, 2018. P. 228-234.
17. Chen H., Jiang L. GAN-based method for cyber-intrusion detection // arXiv preprint arXiv:1904.02426, 2019. P. 1-6.
18. De Paola A., Favaloro S., Gaglio S., Re G.L., Morana M. Malware Detection through Low-level Features and Stacked Denoising Autoencoders // 2nd Italian Conference on Cyber Security (ITASEC), 2018. P. 1-10.
19. Ye Y., Chen L., Hou S., Hardy W., Li X. DeepAM: A Heterogeneous Deep Learning Framework for Intelligent Malware Detection // Knowledge and Information Systems, 2018. Vol. 54. No. 2. P. 265-285.

20. Comodo Anti-Malware Database. Available at: https://www.comodo.com/home/internet-security/updates/vdp/database.php (accessed April 29, 2020).
21. Naway A., Li Y. Android Malware Detection Using Autoencoder // arXiv preprint arXiv:1901.07315. 2019. P. 1-9.
22. Xiao X., Zhang S., Mercaldo F., Hu G., Sangaiah A.K. Android Malware Detection Based on System Call Sequences and LSTM // Multimedia Tools and Applications. 2019. Vol. 78. No. 4. P. 3979-3999.
23. Arp D., Spreitzenbarth M., Hubner M., Gascon H., Rieck K., Siemens, C.E.R.T. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket // Network and Distributed System Security (NDSS), 2014. Vol. 14. P. 23-26.
24. Rhode M., Burnap P., Jones K. Early-stage Malware Prediction Using Recurrent Neural Networks // Computers & security. 2018. Vol. 77. P. 578-594.
25. Darabian H., Homayounoot S., Dehghantanha A., Hashemi S., Karimipour H., Parizi R. M., Choo K.K.R. Detecting Cryptomining Malware: a Deep Learning Approach for Static and Dynamic Analysis // Journal of Grid Computing. 2020. P. 1-11.
26. Karbab E.B., Debbabi M., Derhab A., Mouheb D. MalDozer: Automatic Framework for Android Malware Detection Using Deep Learning // Digital Investigation. 2018. Vol. 24. P. S48-S59.
27. Zhou Y., Jiang X. Dissecting Android Malware: Characterization and Evolution // 2012 IEEE Symposium on Security and Privacy. IEEE, 2012. P. 95-109.
28. Karbab E.B., Debbabi M., Derhab A., Mouheb D. Android malware detection using deep learning on API method sequences //arXiv preprint arXiv:1712.08996. 2017. P. 1-17.
29. Mishra P., Khurana K., Gupt S., Sharma M.K. VMAnalyzer: Malware Semantic Analysis using Integrated CNN and Bi-Directional LSTM for Detecting VM-level Attacks in Cloud // Twelfth International Conference on Contemporary Computing (IC3). IEEE, 2019. P. 1-6.
30. UNM Dataset, 1998. Available at: https://www.cs.unm.edu/~immsec/systemcalls.htm (accessed April 25, 2020).
31. Jan S., Ali T., Alzahrani A., Musa S. Deep Convolutional Generative Adversarial Networks for Intent-based Dynamic Behavior Capture // International Journal of Engineering & Technology, 2018. Vol. 7. No. 4.29. P. 101-103.
32. Amin M., Shah B., Sharif A., Ali T., Kim K.L., Anwar S.. Android Malware Detection through Generative Adversarial Networks // Transactions on Emerging Telecommunications Technologies. 2019. P. e3675.
33. Shibahara T., Yagi T., Akiyama M., Chiba D., Hato K. Efficient Dynamic Malware Analysis for Collecting HTTP Requests using Deep Learning // IEICE Transactions on Information and Systems, 2019. Vol. 102. No. 4. P. 725-736.
34. VirusTotal. Available at: https://virustotal.com (accessed April 29, 2020).
35. Wang P., Ye F., Chen X., Qian Y. DataNet: Deep Learning Based Encrypted Network Traffic Classification in SDN Home Gateway // IEEE Access, 2018. Vol. 6. P. 55380-55391.
36. ISCX VPN-non-VPN dataset. Available at: https://www.unb.ca/cic/datasets/ids.html (accessed April 29, 2020).
37. Lotfollahi M., Siavoshani M.J., Zade R.S.H., Saberian M. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning // Soft Computing. 2020. Vol. 24. No. 3. P. 1999-2012.
38. Smit D., Millar K., Page C., Cheng A., Chew H.G., Lim C.C. Looking deeper: Using deep learning to identify Internet communications traffic // Australasian Conference of Undergraduate Research (ACUR), Adelaide, 2017. P. 124-144.
39. UNSW-NB15 Dataset. Available at: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (accessed April 29, 2020).
40. Tuor A., Kaplan S., Hutchinson B., Nichols N., Robinson S. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams // Workshops at the Thirty-First AAAI Conference on Artificial Intelligence, 2017. P. 224-231.
41. Insider Threat Test Dataset. Available at: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099 (accessed April 29, 2020).
42. Meng F., Lou F., Fu Y., Tian Z. Deep Learning Based Attribute Classification Insider Threat Detection for Data Security // 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, 2018. P. 576-581.
43. Li Y., Nie X., Huang R. Web spam classification method based on deep belief networks // Expert Systems with Applications. 2018. Vol. 96. P. 261-270
44. WEBSPAM-UK2007 (current dataset). Available at: https://chato.cl/webspam/datasets/uk2007/ (accessed April 29, 2020).
45. Jain G., Sharma M., Agarwal B. Optimizing semantic LSTM for spam detection // International Journal of Information Technology. 2019. Vol. 11. No. 2. P. 239-250.
46. Saxe J., Berlin K. eXpose: A Character-Level Convolutional Neural Network with Embeddings for Detecting Malicious URLs, File Paths and Registry Keys // arXiv preprint arXiv:1702.08568. 2017. P. 1-18.
47. Tian Z., Luo C., Qiu J., Du X., Guizani M. A Distributed Deep Learning System for Web Attack Detection on Edge Devices // IEEE Transactions on Industrial Informatics, 2019. P. 1-8.
48. CSIC 2010 HTTP Dataset in CSV Format Available at: https://petescully.co.uk/research/csic-2010-http-dataset-in-csv-format-for-weka-analysis/ (accessed April 29, 2020).
49. Fwaf-Machine-Learning-driven-Web-Application-Firewall Available at: https://github.com/faizann24/ Fwaf-Machine-Learning-driven-Web-Application-Firewall (accessed April 29, 2020).

APPLICATION OF DEEP LEARNING METHODS IN CYBERSECURITY TASKS. PART 2

Gaifulina D.[3], Kotenko I.[4]

The purpose of the article: comparative analysis of methods for solving various cybersecurity problems based on the use of deep learning algorithms.

Research method: Systematic analysis of modern methods of deep learning in various cybersecurity applications, including intrusion and malware detection, network traffic analysis, and some other tasks.

The result obtained: *classification scheme of the considered approaches to deep learning in cybersecurity, and their comparative characteristics by the used models, characteristics, and data sets. The analysis showed that many deeper architectures with a large number of neurons on each layer show better results. Recommendations are given for using deep learning methods in cybersecurity applications.*

The main contribution of the authors to the research of deep learning methods for cybersecurity tasks is the classification of the subject area; conducting a general and comparative analysis of existing approaches that reflect the current state of scientific research.

Keywords: *data science, machine learning, deep neural networks, intrusion detection, malware detection.*

References

1. Gaifulina D., Kotenko I., Application of deep learning methods in cybersecurity tasks. Part 1 // Voprosy kiberbezopasnosti, No.3, 2020. P. 76-86.
2. Gharib M., Mohammadi B., Dastgerdi S.H., Sabokrou M. AutoIDS: Auto-encoder Based Method for Intrusion Detection System // arXiv preprint arXiv:1911.03306. 2019. P. 1-9.
3. Tavallaee M., Bagheri E., Lu W., Ghorbani A.A. A detailed analysis of the KDD CUP 99 data set // Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), Ottawa, ON, Canada, 8–10 July 2009. P. 1-6.
4. Mohamed S., Ejbali R., Zaied M. Denoising Autoencoder with Dropout based Network Anomaly Detection // The Fourteenth International Conference on Software Engineering Advances (ICSEA), 2019. P. 98-104.
5. Ieracitano C., Adeel A., Morabito F.C., Hussain A.A Novel Statistical Analysis and Autoencoder Driven Intelligent Intrusion Detection Approach // Neurocomputing. 2019. P. 1-12.
6. Yang Y., Zheng K., Wu C., Yang Y. Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network // Sensors. 2019. Vol. 19. No. 11. P. 2528.
7. Farahnakian F., Heikkonen J. A deep auto-encoder based approach for intrusion detection system // 20th International Conference on Advanced Communication Technology (ICACT). IEEE, 2018. P. 178-183.
8. Mayuranathan M., Murugan M., Dhanakoti V. Best features based intrusion detection system by RBM model for detecting DDoS in cloud environment // Journal of Ambient Intelligence and Humanized Computing. 2019. P. 1-11.
9. Geem Z.W., Kim J.H., Loganathan G.V. A new heuristic optimization algorithm: harmony search // Simulation. 2001. Vol. 76. No. 2. P. 60-68.
10. KDD Cup 1999 Data. Available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed April 29, 2020).
11. Nguyen K.K., Hoang D.T., Niyato D., Wang P., Nguyen D., Dutkiewicz E. Cyberattack Detection in Mobile Cloud Computing: A Deep Learning Approach // 2018 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2018. P. 1-6.
12. Yin C., Zhu Y., Fei J., He X. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks // IEEE Access, 2017. Vol. 5. P. 21954-21961.
13. Zhu M., Ye K., Wang Y., Xu C.Z. A Deep Learning Approach for Network Anomaly Detection Based on AMF-LSTM // IFIP International Conference on Network and Parallel Computing Springer, Cham, 2018. P. 137-141.
14. Manavi M., Zhang Y. A New Intrusion Detection System Based on Gated Recurrent Unit (GRU) and Genetic Algorithm // International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Springer, Cham, 2019. P. 368-383.
15. Qin Y., Wei J., Yang W. Deep Learning Based Anomaly Detection Scheme in Software-Defined Networking // 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), IEEE, 2019. P. 1-4.
16. Yin C., Zhu Y., Liu S., Fei J., Zhang H. An Enhancing Framework for Botnet Detection Using Generative Adversarial Networks // 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD). IEEE, 2018. P. 228-234.
17. Chen H., Jiang L. GAN-based method for cyber-intrusion detection // arXiv preprint arXiv:1904.02426, 2019. P. 1-6.

3 Diana Gaifulina, Ph.D. student, Junior Researcher of Laboratory for Computer Security Problems at St. Petersburg Institute for Informatics and Automation of Russian Academy of Science, Moscow, Russia. E-mail: gaifulina@comsec.spb.ru

4 Igor Kotenko, Dr.Sc., Professor, Head of Laboratory of Computer Security Problems at St. Petersburg Institute for Informatics and Automation of Russian Academy of Science, St. Petersburg, Russia. E-mail: ivkote@comsec.spb.ru

18. De Paola A., Favaloro S., Gaglio S., Re G.L., Morana M. Malware Detection through Low-level Features and Stacked Denoising Autoencoders // 2nd Italian Conference on Cyber Security (ITASEC), 2018. P. 1-10.
19. Ye Y., Chen L., Hou S., Hardy W., Li X. DeepAM: A Heterogeneous Deep Learning Framework for Intelligent Malware Detection // Knowledge and Information Systems, 2018. Vol. 54. No. 2. P. 265-285.
20. Comodo Anti-Malware Database. Available at: https://www.comodo.com/home/internet-security/updates/vdp/database.php (accessed April 29, 2020).
21. Naway A., Li Y. Android Malware Detection Using Autoencoder // arXiv preprint arXiv:1901.07315. 2019. P. 1-9.
22. Xiao X., Zhang S., Mercaldo F., Hu G., Sangaiah A.K. Android Malware Detection Based on System Call Sequences and LSTM // Multimedia Tools and Applications. 2019. Vol. 78. No. 4. P. 3979-3999.
23. Arp D., Spreitzenbarth M., Hubner M., Gascon H., Rieck K., Siemens, C.E.R.T. DREBIN: Effective and Explainable Detection of Android Malware in Your Pocket // Network and Distributed System Security (NDSS), 2014. Vol. 14. P. 23-26.
24. Rhode M., Burnap P., Jones K. Early-stage Malware Prediction Using Recurrent Neural Networks // Computers & security. 2018. Vol. 77. P. 578-594.
25. Darabian H., Homayounoot S., Dehghantanha A., Hashemi S., Karimipour H., Parizi R. M., Choo K.K.R. Detecting Cryptomining Malware: a Deep Learning Approach for Static and Dynamic Analysis // Journal of Grid Computing. 2020. P. 1-11.
26. Karbab E.B., Debbabi M., Derhab A., Mouheb D. MalDozer: Automatic Framework for Android Malware Detection Using Deep Learning // Digital Investigation. 2018. Vol. 24. P. S48-S59.
27. Zhou Y., Jiang X. Dissecting Android Malware: Characterization and Evolution // 2012 IEEE Symposium on Security and Privacy. IEEE, 2012. P. 95-109.
28. Karbab E.B., Debbabi M., Derhab A., Mouheb D. Android malware detection using deep learning on API method sequences //arXiv preprint arXiv:1712.08996. 2017. P. 1-17.
29. Mishra P., Khurana K., Gupt S., Sharma M.K. VMAnalyzer: Malware Semantic Analysis using Integrated CNN and Bi-Directional LSTM for Detecting VM-level Attacks in Cloud // Twelfth International Conference on Contemporary Computing (IC3). IEEE, 2019. P. 1-6.
30. UNM Dataset, 1998. Available at: https://www.cs.unm.edu/~immsec/systemcalls.htm (accessed April 25, 2020).
31. Jan S., Ali T., Alzahrani A., Musa S. Deep Convolutional Generative Adversarial Networks for Intent-based Dynamic Behavior Capture // International Journal of Engineering & Technology, 2018. Vol. 7. No. 4.29. P. 101-103.
32. Amin M., Shah B., Sharif A., Ali T., Kim K.L., Anwar S.. Android Malware Detection through Generative Adversarial Networks // Transactions on Emerging Telecommunications Technologies. 2019. P. e3675.
33. Shibahara T., Yagi T., Akiyama M., Chiba D., Hato K. Efficient Dynamic Malware Analysis for Collecting HTTP Requests using Deep Learning // IEICE Transactions on Information and Systems, 2019. Vol. 102. No. 4. P. 725-736.
34. VirusTotal. Available at: https://virustotal.com (accessed April 29, 2020).
35. Wang P., Ye F., Chen X., Qian Y. DataNet: Deep Learning Based Encrypted Network Traffic Classification in SDN Home Gateway // IEEE Access, 2018. Vol. 6. P. 55380-55391.
36. ISCX VPN-non-VPN dataset. Available at: https://www.unb.ca/cic/datasets/ids.html (accessed April 29, 2020).
37. Lotfollahi M., Siavoshani M.J., Zade R.S.H., Saberian M. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning // Soft Computing. 2020. Vol. 24. No. 3. P. 1999-2012.
38. Smit D., Millar K., Page C., Cheng A., Chew H.G., Lim C.C. Looking deeper: Using deep learning to identify Internet communications traffic // Australasian Conference of Undergraduate Research (ACUR), Adelaide, 2017. P. 124-144.
39. UNSW-NB15 Dataset. Available at: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (accessed April 29, 2020).
40. Tuor A., Kaplan S., Hutchinson B., Nichols N., Robinson S. Deep learning for unsupervised insider threat detection in structured cybersecurity data streams // Workshops at the Thirty-First AAAI Conference on Artificial Intelligence, 2017. P. 224-231.
41. Insider Threat Test Dataset. Available at: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099 (accessed April 29, 2020).
42. Meng F., Lou F., Fu Y., Tian Z. Deep Learning Based Attribute Classification Insider Threat Detection for Data Security // 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC). IEEE, 2018. P. 576-581.
43. Li Y., Nie X., Huang R. Web spam classification method based on deep belief networks // Expert Systems with Applications. 2018. Vol. 96. P. 261-270
44. WEBSPAM-UK2007 (current dataset). Available at: https://chato.cl/webspam/datasets/uk2007/ (accessed April 29, 2020).
45. Jain G., Sharma M., Agarwal B. Optimizing semantic LSTM for spam detection // International Journal of Information Technology. 2019. Vol. 11. No. 2. P. 239-250.
46. Saxe J., Berlin K. eXpose: A Character-Level Convolutional Neural Network with Embeddings for Detecting Malicious URLs, File Paths and Registry Keys // arXiv preprint arXiv:1702.08568. 2017. P. 1-18.
47. Tian Z., Luo C., Qiu J., Du X., Guizani M. A Distributed Deep Learning System for Web Attack Detection on Edge Devices // IEEE Transactions on Industrial Informatics, 2019. P. 1-8.
48. CSIC 2010 HTTP Dataset in CSV Format Available at: https://petescully.co.uk/research/csic-2010-http-dataset-in-csv-format-for-weka-analysis/ (accessed April 29, 2020).
49. Fwaf-Machine-Learning-driven-Web-Application-Firewall Available at: https://github.com/faizann24/ Fwaf-Machine-Learning-driven-Web-Application-Firewall (accessed April 29, 2020).