УДК 004.85

СРАВНЕНИЕ РЕКУРРЕНТНОЙ И СВЕРТОЧНОЙ НЕЙРОСЕТЕЙ ДЛЯ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ УГРОЗ КИБЕРБЕЗОПАСНОСТИ

Петров И. А.[1]

Научный руководитель: **Крылов Г. О.**[1], доктор физико-математических наук, кандидат технических наук, кандидат экономических наук, профессор кафедры информационной безопасности

[1]Финансовый университет при Правительстве РФ, Москва, Россия

Статья **поступила** 28.11.2024, **принята** к публикации 25.12.2024. Опубликована онлайн.

Аннотация. Кибератаки становятся все более сложными и изощренными, что требует разработки новых методов анализа и предсказания угроз. Традиционные методы, такие как сигнатурный анализ и статистические модели, часто оказываются неэффективными против современных угроз. Рекуррентные нейронные сети (RNN) представляют собой перспективный инструмент для решения этой задачи благодаря своей способности обрабатывать временные ряды данных и выявлять сложные закономерности. В данной статье рассматриваются возможности применения рекуррентных и сверточных нейросетей для анализа и прогнозирования киберугроз, а также приводятся примеры успешных реализаций этих моделей в области кибербезопасности. Для обучения модели была использована база данных, состоящая из более чем 125 тысяч кибератак. Целью данной статьи является создание и исследование возможностей применения рекуррентных и сверточных нейронных сетей в контексте анализа и прогнозирования угроз информационной безопасности. Научная новизна данной статьи заключается в сравнении результатов работы различных нейросетей с разными параметрами. Методы исследований: системный анализ существующих методов машинного обучения, теоретическая формализация, проведение эксперимента.

Ключевые слова: искусственный интеллект, машинное обучение, кибербезопасность, информационная безопасность, рекуррентная нейросеть, сверточная нейросеть

COMPARISON OF RECURRENT AND CONVOLUTIONAL NEURAL NETWORKS FOR ANALYZING AND PREDICTING CYBER SECURITY THREATS

Petrov I. A.[1]

Scientific supervisor: **Krylov G. O.**[1]**,** Doctor of Physical and Mathematical Sciences, Candidate of Technical Sciences, Candidate of Economic Sciences, Professor of the Department of Information Security Financial

[1]Financial University under the Government of the Russian Federation, Russia, Moscow

Abstract. Cyberattacks are becoming increasingly complex and sophisticated, which requires the development of new methods for analyzing and predicting threats. Traditional methods such as signature analysis and statistical models are often ineffective against advanced threats. Recurrent neural networks (RNN) represent a promising tool for this task due to their ability to process time-series data and identify complex patterns.

Today's information systems face an ever-increasing number of cyber threats. These threats range from simple phishing attacks to sophisticated campaigns carried out by organized criminal groups. Securing information resources requires the development of effective methods to analyze and anticipate potential threats. Current approaches include the use of machine learning and artificial intelligence (AI). In order to use AI as part of information security challenges, particularly cybersecurity, it is necessary to understand which algorithm is best suited for analyzing, predicting and detecting cyber threats in modern and complex information systems. This paper discusses the potential applications of RNN for analyzing and predicting cyber threats and provides examples of successful implementations of these models in the cybersecurity domain. A database consisting of 40000 cyberattacks was used to train the model. The purpose of this paper is to investigate the potential applications of recurrent neural networks in the context of analyzing and predicting information security threats. The scientific novelty of this article lies in the comparison of the results of different neural networks. Research methods: system analysis of existing machine learning methods, theoretical formalization, experimentation.

Keywords: artificial intelligence, machine learning, cybersecurity, information security, recurrent neural network, convolutional neural network

Введение

Современные информационные системы сталкиваются с постоянно растущим числом киберугроз. Эти угрозы варьируются от простых фишинговых атак до сложных кампаний, проводимых организованными преступными группировками. Для обеспечения безопасности информационных ресурсов требуется разработка эффективных методов анализа и предвидения потенциальных угроз. Современные подходы включают использование машинного обучения и искусственного интеллекта (ИИ).

Для того, чтобы использовать ИИ в рамках задач информационной безопасности, в частности, кибербезопасности, необходимо понять, какой алгоритм больше всего подходит для анализа, прогнозирования и выявления киберугроз в современных и сложных информационных системах.

Применение ИИ в области информационной безопасности

Для решения задач кибербезопасности многие исследователи использовали различные алгоритмы машинного обучения. Популярным алгоритмом является Random Forest [1]. Как и все алгоритмы машинного обучения, данный алгоритм обладает как достоинствами, так и недостатками.

Достоинствами Random Forest для задач кибербезопасности являются:

– Высокая устойчивость к переобучению: Random Forest минимизирует риск переобучения благодаря использованию множества деревьев решений, каждое из которых обучено на случайной подвыборке данных. Это особенно важно в условиях ограниченного объема данных или наличия шума в данных, что характерно для задач кибербезопасности.

– Способность работать с большими объемами данных: алгоритм Random Forest хорошо масштабируется и способен эффективно обрабатывать большие объемы данных, такие как журналы событий, сетевые пакеты и другие источники информации. Это делает его полезным для анализа больших объемов данных в режиме реального времени.

– Универсальность: данный алгоритм применим как для задач классификации (например, классификация вредоносных программ), так и для задач регрессии (например, прогнозирование объема трафика). Это позволяет использовать Random Forest в различных аспектах кибербезопасности.

– Простота интерпретации: хотя сам ансамбль деревьев может быть сложным для понимания, отдельные деревья решений относительно просты для интерпретации. Это облегчает понимание принципов работы модели и принятие решений на основе ее выводов.

– Оценка важности признаков: Random Forest предоставляет встроенную оценку важности признаков, что помогает специалистам по кибербезопасности определить, какие факторы имеют наибольшую значимость для обнаружения угроз. Это может способствовать оптимизации процессов мониторинга и реагирования.

Недостатками Random Forest для задач кибербезопасности являются:

– Вычислительная сложность: построение большого числа деревьев решений может требовать значительных вычислительных ресурсов, особенно при работе с большими наборами данных. В условиях необходимости быстрого реагирования на угрозы. Это может стать проблемой.

– Трудности с интерпретацией конечной модели: несмотря на простоту отдельных деревьев решений, ансамбль из множества деревьев может оказаться сложным для полного понимания. Это усложняет объяснение решений модели и может привести к трудностям в принятии обоснованных решений.

– Зависимость от настроек гиперпараметров: эффективность Random Forest сильно зависит от правильных настроек гиперпараметров, таких как количество деревьев, максимальная глубина деревьев и количество признаков для разделения. Неправильная настройка может привести к снижению точности модели.

– Проблемы с балансировкой классов: в задачах кибербезопасности часто встречаются несбалансированные классы данных, например, малое количество аномальных событий по сравнению с нормальными. Random Forest может испытывать трудности с такими данными, что потребует дополнительных техник для балансировки классов.

– Отсутствие гарантии оптимальности: метод Random Forest не гарантирует нахождения глобально оптимальной модели. В некоторых случаях другие алгоритмы, такие как градиентный бустинг или глубокое обучение, могут показать лучшие результаты.

Сверточные нейронные сети

Также для задач кибербезопасности используют сверточные нейронные сети. Например, в работе [2] 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.

Данная работа нацелена на обнаружение вторжений, сверточные нейронные сети (Convolutional Neural Network, CNN) также используются для обнаружения вредоносного программного обеспечения, например, в работе [3], где представлен метод обнаружения вредоносного программного обеспечения для операционной системы Android. CNN также используется для анализа сетевого трафика.

В работе [4] представлен подход к классификации зашифрованного трафика с применением многослойного персептрона, многоуровневого автокодировщика и CNN.

Несмотря на популярность данного метода, у него существует ряд недостатков.

1. Высокая требовательность к ресурсам. Сверточные сети требуют значительных вычислительных мощностей для обучения и тестирования. Это связано с большим количеством слоев и параметров, которые необходимо оптимизировать. В условиях ограниченных ресурсов, таких как мобильные устройства или встраиваемые системы, использование CNN может быть затруднительно.
2. Сложность адаптации к специфическим задачам. Несмотря на успех CNN в задачах обработки изображений и видео, адаптация этих сетей к специфичным задачам кибербезопаности может представлять сложности. Например, для анализа сетевого трафика или обнаружения вредоносного ПО требуются иные представления данных, отличные от изображений, что может снизить эффективность стандартных архитектур CNN.
3. Проблема переноса знаний. CNN, обученные на одних данных, могут плохо переносить знания на другие домены. Например, сеть, обученная на данных из одной компании, может показывать низкую точность при применении к данным другой компании из-за различий в структуре и характеристиках данных. Это требует дополнительного обучения и настройки моделей для каждого конкретного случая.
4. Чувствительность к изменениям в данных. CNN чувствительны к небольшим изменениям в структуре данных. Например, незначительное изменение формата сетевого пакета или кода вредоносной программы может привести к тому, что сеть перестанет правильно классифицировать данные. Это создает необходимость постоянного обновления и переобучения моделей.
5. Трудности с интерпретацией решений. Как и большинство глубоких нейронных сетей, CNN сложно интерпретировать. Это затрудняет понимание причин, по которым сеть приняла то или иное решение. В контексте кибербезопасности это может быть критично, так как специалисты должны понимать, почему система классифицирует определенные данные как угрозу.
6. Ограниченность в обработке структурированных данных. CNN преимущественно ориентированы на обработку двумерных данных, таких как изображения. В задачах кибербезопасности часто приходится иметь дело со структурированными данными, такими как лог-файлы, таблицы и другие форматы, которые не всегда удобно преобразовывать в формат, подходящий для CNN.

Помимо вышеперечисленных алгоритмов машинного обучения существуют также генеративно-состязательные сети, глубокие сети доверия, ограниченные машины Больцмана, автокодировщики, которые применяются для задач информационной и кибербезопасности. Однако, для анализа и выявления угроз кибербезопасности, мной была выбрана рекуррентная нейронная сеть.

Применение RNN в кибербезопасности. Использование рекуррентных нейронных сетей для анализа и предсказания киберугроз имеет ряд преимуществ перед традиционными методами. В работе [5] описано много методов применения данной модели. Однако большинство моделей, описываемых в рамках данных задач, узкоспециализированы [6-11].

Поскольку данная статья написана в рамках написания диссертационной работы на тему «Проблемы безопасности суверенных цифровых активов», в процессе ее написания было установлено, что системы обращения цифровых активов подвержены абсолютно разным рискам, таким как сетевые атаки, вредоносное программное обеспечение, фишинг и тд. Поэтому для обучения модели RNN необходима большая база данных, включающая данные о вышеперечисленных компьютерных атаках. Но для начала необходимо понять, что представляет из себя данный алгоритм, его недостатки и достоинства.

Основой RNN является концепция сохранения состояния на протяжении времени. Вместо обработки независимых данных, как это делают обычные нейронные сети, RNN хранят информацию о предыдущем состоянии и используют её для обработки следующего элемента последовательности.

Базовая формула RNN

Основным уравнением RNN для вычисления скрытого состояния ht в момент времени t является следующее:

$$h_t = f(W_h h_{t-1} + W_x x_t + b), \quad (1)$$

где: f – нелинейная функция активации, например, гиперболический тангенс, W_h – матрица весов для связи между скрытыми состояниями, h_{t-1} – скрытое состояние на предыдущем временном шаге, W_x – матрицы весов для связей между входами и скрытыми состояниями, x_t – вход на текущем временном шаге, b – смещение.

Выход RNN

Выход y_t на временном шаге t вычисляется как:

$$y_t = W_y h_t + c, \quad (2)$$

где: W_y – матрица весов между скрытым состоянием и выходом, c – вектор смещений для выхода.

Рекуррентные нейронные сети обладают рядом уникальных достоинств, которые делают их ценными инструментами в решении задач кибербезопасности. Рассмотрим основные из них:

1. Способность обрабатывать временные ряды. RNN специально спроектированы для работы с последовательными данными, такими как временные ряды. Это позволяет им эффективно анализировать сетевой трафик, лог-файлы и другие виды данных, изменяющиеся во времени. В кибербезопасности такая способность важна для обнаружения аномалий и прогнозирования угроз.
2. Учет контекста. RNN могут запоминать и использовать информацию из предыдущих этапов обработки, что дает им преимущество в учете контекста. Это особенно полезно в задачах, где важно понимать взаимосвязи между событиями, происходящими в разное время.
3. Гибкость и адаптивность. RNN способны адаптироваться к различным видам данных и задачам. Они могут быть использованы для анализа текстовых данных, таких как лог-файлы, а также для обработки числовых данных, например, сетевого трафика. Это делает их универсальными инструментами для решения различных задач кибербезопасности.
4. Возможность обработки длинных последовательностей. Некоторые разновидности RNN, такие как LSTM (Long Short-Term Memory) и GRU (Gated Recurrent Units), способны эффективно обрабатывать длинные последовательности данных. Это полезно в ситуациях, когда необходимо анализировать большие объемы данных или сложные сценарии атак.
5. Поддержка онлайн-обучения. RNN поддерживают онлайн-обучение, что позволяет им адаптироваться к изменениям в данных в реальном времени. Это важно в условиях, когда угроза может эволюционировать или изменяться со временем.
6. Возможность интеграции с другими моделями. RNN можно комбинировать с другими видами нейронных сетей, такими как сверточные нейронные сети (CNN), для создания гибридных моделей. Это расширяет возможности анализа данных и повышает точность и надежность моделей.

Однако их применение в сфере кибербезопасности также сопряжено с рядом недостатков. Рассмотрим основные из них:

1. Долгосрочная зависимость. RNN испытывают трудности с сохранением долгосрочных зависимостей в последовательностях данных. Это связано с эффектом исчезающего градиента, когда информация теряется при передаче через большое количество временных шагов.

В задачах кибербезопасности, где важны длительные последовательности событий, это может приводить к снижению точности модели.

2. Медленная скорость обучения. Обучение RNN занимает значительное время из-за необходимости прохождения всей последовательности данных для обновления весов. Это особенно актуально для длинных последовательностей, таких как сетевой трафик или лог-файлы. В условиях, когда требуется быстрое реагирование на угрозы, медленное обучение может стать серьезным препятствием.

3. Требовательность к объему данных. Для эффективного обучения RNN необходимы большие объемы данных. В сфере кибербезопасности доступ к достаточному количеству качественных данных может быть ограничен, что затрудняет создание точных и надежных моделей.

4. Ограниченная параллельность. RNN работают последовательно, обрабатывая каждую часть последовательности по очереди. Это ограничивает возможности параллельной обработки данных, что замедляет выполнение модели и увеличивает затраты на ресурсы.

5. Чувствительность к шуму. RNN чувствительны к шумовым данным, которые могут повлиять на точность модели. В кибербезопасности шумовые данные, такие как случайные отклонения в сетевом трафике или ошибочные записи в лог-файлах, могут привести к неправильной классификации событий.

6. Ограниченное представление контекста. RNN могут терять контекст при переходе от одной части последовательности к другой. Это может быть проблематично в задачах, где важен учет длительного контекста, например, при анализе сетевого трафика или обнаружении сложных атак.

Сравнение работы нейросетей для задач кибербезопасности

Для построения модели реккурентной нейронной сети для прогноза и обнаружения сетевых атак была использована база данных NSL KDD+. Каждая запись в базе NSL-KDD Dataset представляет собой последовательность пакетов, зафиксированную за промежуток времени. Данная последовательность – это поток данных между источником и адресатом сетевых пакетов в соответствии с IP-адресом в заголовке пакета.

Записи включают 41 информационный признак и промаркированы как «атака» и «не атака». База содержит 36 типов атаки, разделенных на четыре категории:

– Denial of Service (dos): набор атак, в которых злоумышленник ограничивает доступ верифицированным пользователям к конкретному сервису через определенный протокол (Back, Land, Neptune, Pod, Smurf, Teardrop, Apache2, Udpstorm, Processtable, Worm);

– Remote to Local (r2l): набор атак, в которых злоумышленник пытается получить доступ извне к локальной машине пользователя (Guess_Password, Ftp_write, Imap, Phf, Multihop, Warezmaster, Warezclient, Spy, Xlock, Xsnoop, Snmpguess, Snmpgetattack, Httptunnel, Sendmail, Named);

– User to Root (u2r): набор атак, в которых злоумышленник, имея доступ к машине жертвы, пытается получить права более привилегированного пользователя (Buffer_overflow, Loadmodule, Rootkit, Perl, Sqlattack, Xterm, Ps);

– Probe: набор атак, в которых злоумышленник пытается получить сведения об инфраструктуре пользователя (Satan, Ipsweep, Nmap, Portsweep, Mscan, Saint) [12].

Модель была создана в среде Google Colab. Основными показателями, определяющими качество и точность модели, являются:

1. ***Precision*** (точность). Это доля истинных положительных результатов среди всех положительных предсказаний. Она показывает, насколько точно модель идентифицирует положительные классы:

$$\text{Precision}=\frac{TP}{TP+FP},$$

где: TP (True Positives) – количество истинных положительных результатов (правильно предсказанные положительные классы); FP (False Positives) – количество ложных положительных результатов (неправильно предсказанные положительные классы).

2. ***Recall*** – это доля истинных положительных результатов среди всех фактических положительных классов. Она показывает, насколько хорошо модель находит все положительные классы:

$$\text{Recall}=\frac{\text{TP}}{\text{TP+FN}},$$

где FN – (False Negatives) – количество ложных отрицательных результатов (неправильно предсказанные отрицательные классы).

3. ***F1-мера*** – это гармоническое среднее между ***precision*** и ***recall***. Она используется для баланса между этими двумя метриками, особенно в случаях, когда классы не сбалансированы:

$$\text{F1}=2*\frac{\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$$

В NSL-KDD содержатся 125972 записи для обучения и 22182 записи для тестирования.

Также для обучения были использованы параметры ***epochs,*** равная 5, ***batch size*** = 32, LSTM = 128. Результаты обучения модели представлены на рисунке 1.

```
2205/2205 ━━━━━━━━━━━━━━━━━━━━ 30s 14ms/step - accuracy: 0.9910 - loss: 0.0293
788/788 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step
                precision    recall  f1-score   support

          back       0.93      0.98      0.96       185
buffer_overflow      0.33      0.33      0.33         9
   guess_passwd      1.00      0.91      0.95        11
          imap       1.00      1.00      1.00         1
       ipsweep       0.99      0.92      0.95       733
          land       0.50      1.00      0.67         3
       neptune       1.00      1.00      1.00      8228
          nmap       0.95      0.95      0.95       313
        normal       0.99      0.99      0.99     13422
          perl       0.00      0.00      0.00         1
           phf       0.00      0.00      0.00         1
           pod       1.00      0.93      0.96        43
     portsweep       0.99      1.00      1.00       573
       rootkit       0.00      0.00      0.00         1
         satan       0.99      0.95      0.97       738
         smurf       0.99      1.00      0.99       534
           spy       0.00      0.00      0.00         1
      teardrop       1.00      1.00      1.00       188
   warezclient       0.77      0.88      0.82       202
   warezmaster       0.00      0.00      0.00         8

      accuracy                           0.99     25195
     macro avg       0.67      0.69      0.68     25195
  weighted avg       0.99      0.99      0.99     25195
```

Рис.1. Отчет о работе рекуррентной нейронной сети с параметром *LSTM* =128

Полученная модель показывает точность в 99,1%. Чем больше записей с той или иной атакой, тем большая точность достигается. Некоторых атак в базе данных очень мало, поэтому нейросеть не может рассчитать точность. Однако, чем больше количество LSTM-слоев, тем модель лучше справляется с задачей при минимальной выборке. Что особенно заметно при атаке ***buffer_overflow, warezmaster, warezclient.***

Далее создадим сверточную нейронную сеть, обучив ее на той же базе данных и сравним результаты с рекуррентной нейронной сетью (рис.3).

```
2205/2205 ━━━━━━━━━━━━━━━━━━━━ 82s 22ms/step - accuracy: 0.9909 - loss: 0.0280
788/788 ━━━━━━━━━━━━━━━━━━━━ 6s 8ms/step
                precision    recall  f1-score   support

          back       0.96      0.98      0.97       185
buffer_overflow      0.50      0.22      0.31         9
   guess_passwd      1.00      0.91      0.95        11
          imap       1.00      1.00      1.00         1
       ipsweep       0.93      1.00      0.96       733
          land       0.60      1.00      0.75         3
       neptune       1.00      1.00      1.00      8228
          nmap       0.95      0.97      0.96       313
        normal       0.99      0.99      0.99     13422
          perl       1.00      1.00      1.00         1
           phf       1.00      1.00      1.00         1
           pod       1.00      0.93      0.96        43
     portsweep       1.00      1.00      1.00       573
       rootkit       0.00      0.00      0.00         1
         satan       1.00      0.95      0.97       738
         smurf       0.97      1.00      0.98       534
           spy       0.00      0.00      0.00         1
      teardrop       1.00      1.00      1.00       188
    warezclient      0.92      0.83      0.88       202
    warezmaster      0.75      0.75      0.75         8

      accuracy                           0.99     25195
     macro avg       0.83      0.83      0.82     25195
  weighted avg       0.99      0.99      0.99     25195
```

Рис.2. Отчет о работе рекуррентной нейронной сети с параметром LSTM =256

```
788/788 ━━━━━━━━━━━━━━━━━━━━ 3s 4ms/step - accuracy: 0.9870 - loss: 0.0438
Loss: 0.04317694529891014, Accuracy: 0.9867830872535706
788/788 ━━━━━━━━━━━━━━━━━━━━ 2s 2ms/step
                precision    recall  f1-score   support

          back       0.92      0.99      0.96       185
buffer_overflow      0.67      0.22      0.33         9
      ftp_write      0.00      0.00      0.00         0
   guess_passwd      0.91      0.91      0.91        11
          imap       1.00      1.00      1.00         1
       ipsweep       0.88      0.99      0.94       733
          land       0.00      0.00      0.00         3
    loadmodule       0.00      0.00      0.00         0
      multihop       0.00      0.00      0.00         0
       neptune       1.00      1.00      1.00      8228
          nmap       0.97      0.93      0.95       313
        normal       1.00      0.98      0.99     13422
          perl       1.00      1.00      1.00         1
           phf       1.00      1.00      1.00         1
           pod       1.00      0.93      0.96        43
     portsweep       0.98      0.99      0.99       573
       rootkit       0.00      0.00      0.00         1
         satan       0.94      0.97      0.96       738
         smurf       0.91      1.00      0.95       534
           spy       0.00      0.00      0.00         1
      teardrop       1.00      1.00      1.00       188
    warezclient      0.84      0.88      0.86       202
    warezmaster      1.00      0.75      0.86         8

      accuracy                           0.99     25195
     macro avg       0.70      0.68      0.68     25195
  weighted avg       0.99      0.99      0.99     25195
```

Рис.3 Отчет о работе сверточной нейронной сети

Данная сеть также показывает очень достойные результаты, такие как точность в 98,6%, что немного ниже рекуррентной нейронной сети. Поскольку полученные модели не смогли рассчитать точность из-за малой выборки некоторых кибератак, данных о параметрах ***precision*** и ***recall*** нет. Но модернизировав базу данных NSL KDD, можно рассчитать данные параметры.

Модернизировав базу данных, мы получили следующие параметры моделей (таб.1).

Таблица 1

Параметры моделей нейросети

	Precision	Recall	F1-score
Реккурентная модель (LSTM =128)	0,89	0,92	0,9
Реккурентная модель (LSTM =256)	0,91	0,9	0,89
Сверточная модель	0,94	0,89	0,92

Исходя из таблицы 1, можно сказать, что в целом модели показывают очень близкие показатели. Однако, проанализировав показатели каждой кибератаки отдельно, можно сделать вывод, что сверточная модель обладает немного лучшими показателями. Снижение показателей ***recall и F1-score*** при LSTM =256 объясняется тем, что при увеличении LSTM-слоев, модель смогла обработать те кибератаки, которые при LSTM=128 она не смогла обработать.

Заключение

Для написания данной статьи были написаны две модели нейронной сети: рекуррентная и сверточная, а также проанализированы их результаты с различными параметрами. Обе модели обучались на одинаковых базах данных NSL KDD. Модели показали очень близкие результаты. Разница в точности модели составила около 0,5% в пользу рекуррентной.

Однако, сверточная модель точнее определяет кибератаки, которых в выборке мало. Реккурентная модель точнее при большой выборке, также чем выше количество LSTM-слоев, тем точнее модель. Но при увеличении количества LSTM-слоев, скорость работы модели заметно снижается.

Также по результатам проделанной работы можно сказать, что обучение модели очень сильно зависит от базы данных, на которой нейросеть обучается. В ней должно быть достаточно данных для обучения и тестирования, данные должны быть сбалансированно разделены на записи для обучения и тестирования, не должно быть избыточных и дублирующихся записей.

Таким образом, сделан вывод, что обе модели нейросети хорошо подходят для анализа и прогнозирования кибератак, и созданные модели дают высокую точность в 99%.

Список литературы

1. Павлычев, А. В. Использование алгоритма машинного обучения Random Forest для выявления сложных компьютерных инцидентов / А. В. Павлычев, М. И. Стародубов, А. Д. Галимов // Вопросы кибербезопасности. – 2022. – № 5(51). – С. 74-81. – DOI 10.21681/2311-3456-2022-5-74-81. – EDN ZAPFHO

2. 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

3. 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.

4. 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

5. Гайфулина, Д. А. Применение методов глубокого обучения в задачах кибербезопасности. Часть 2 / Д. А. Гайфулина, И. В. Котенко // Вопросы кибербезопасности. – 2020. – № 4(38). – С. 11-21. – DOI 10.21681/2311-3456-2020-04-11-21. – EDN MEZKLH

6. 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.
7. 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.
8. 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.
9. 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.
10. VirusTotal. Available at: https://virustotal.com (accessed November 06, 2024).
11. 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.
12. Зуев, В. Н. Обнаружение аномалий сетевого трафика методом глубокого обучения / В. Н. Зуев // Программные продукты и системы. – 2021. – № 1. – С. 91-97. – DOI 10.15827/0236-235X.133.091-097. – EDN YCVLDE

References

1. Pavlychev, A. V. Using the Random Forest machine learning algorithm to identify complex computer incidents / A. V. Pavlychev, M. I. Starodubov, A. D. Galimov // Voprosy cybersecurity. – 2022. – № 5(51). – С. 74-81. – DOI 10.21681/2311-3456-2022-5-74-81. – EDN ZAPFHO
2. 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
3. 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.
4. 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
5. Gaifulina, D. A. Application of deep learning methods in cybersecurity tasks. Part 2 / D. A. Gaifulina, I. V. Kotenko // Voprosy cybersecurity. – 2020. – № 4(38). – С. 11-21. – DOI 10.21681/2311-3456-2020-04-11-21. – EDN MEZKLH
6. 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.
7. 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.
8. 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.
9. 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.
10. VirusTotal. Available at: https://virustotal.com (accessed November 06, 2024).
11. 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.
12. Zuev, V. N. Detection of the network traffic anomalies by the deep learning method / V. N. Zuev // Software Products and Systems. – 2021. – № 1. – С. 91-97. – DOI 10.15827/0236-235X.133.091-097. – EDN YCVLDE

СВЕДЕНИЯ ОБ АВТОРЕ/ ABOUT THE AUTHOR

Петров Иван Андреевич, аспирант и ассистент кафедры информационной безопасности, Финансовый университет при Правительстве РФ, 125167, Москва, проспект Ленинградский, д. 49/2, iapetrov@fa.ru

Petrov Ivan Andreevich, Postgraduate and Assistant at the Department of Information Security Financial University under The Government of Russian Federation, Russia, Moscow, 49/2 Leningradsky Ave., 125167