ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ТЕЛЕКОММУНИКАЦИИ
INFORMATION TECHNOLOGY AND TELECOMMUNICATIONS

УДК 004.896

DOI: 10.21822/2073-6185-2024-51-2-137-153 Оригинальная статья /Original article

Создание и обучение искусственной нейронной сети с целью детектирования, классификации и блокировки сетевых DDos–атак («отказ в обслуживании»)

П.В. Разумов, Л.В. Черкесова, Е.А. Ревякина
Донской государственный технический университет
344000, г. Ростов-на-Дону, пл. Гагарина, 1, Россия

Резюме. Цель. Целью исследования является разработка искусственной нейронной сети (ИНС) для детектирования и нейтрализации сетевых DDoS–атак. **Метод.** Исследование основано на применении языка программирования Python в среде, поддерживающей функции обучения нейронных сетей PyCharm. **Результат.** Проведён анализ существующих искусственных нейронных сетей для определения их оптимальной структуры; изучены существующие методы детектирования сетевых DDoS– атак; собраны и доработаны датасеты для улучшения качества обучения; создана структура искусственной нейронной сети классификатора и проведено её обучение, создано демонстрационное программное средство, иллюстрирующее процесс классификации и блокировки и нейтрализации DDoS–атак. **Вывод.** Наличие систем для мониторинга трафика, брандмауэр Web–приложений, ограничение скорости, страница состояния и лицо компании, которое отвечает на вопросы в социальных сетях, — все это поможет обеспечить максимально эффективную защиту от DDoS–атак.

Ключевые слова: кибератака, DDoS–атака, мониторинг трафика, брандмауэр Web–приложений, искусственный интеллект, искусственные нейронные сети, машинное обучение

Для цитирования: П.В. Разумов, Л.В. Черкесова, Е.А. Ревякина. Создание и обучение искусственной нейронной сети с целью детектирования, классификации и блокировки сетевых DDos–атак («отказ в обслуживании»). Вестник Дагестанского государственного технического университета. Технические науки. 2024; 51(2):137-153. DOI:10.21822/2073-6185-2024-51-2-137-153

Creation and Training of artifical neural network for Detection and Neutralization of Network DDos–attacks(“Denial of Service”)

P.V. Razumov, L.V. Cherckesova, E.A. Revyakina
Don State Technical University,
1 Gagarin Square, Rostov-on-Don 344000, Russia

Abstract. Objective. The goal of the research is to develop an artificial neural network (ANN) to detect and neutralize network DDoS attacks. **Method.** The research is based on the use of the Python programming language in an environment that supports the training functions of PyCharm neural networks. **Result.** An analysis of existing artificial neural networks was carried out to determine their optimal structure; Existing methods for detecting network DDoS attacks have been studied; Datasets were collected and refined to improve the quality of training; The structure of the artificial neural network of the classifier was created and its training was carried out, a demonstration software was created that illustrates the process of classification and blocking and neutralizing DDoS attacks. **Conclusion.** Having systems to monitor traffic, a Web application firewall, speed limiting, a status page, and a company face to answer questions on social media will all help ensure the most effective protection against DDoS attacks.

Keywords: cyberattack, DDoS attack, traffic monitoring, web application firewall, artificial intelligence, artificial neural networks, machine learning

For citation: P.V. Razumov, L.V. Cherckesova, E.A. Revyakina. Creation and Training of artifical neural network for Detection and Neutralization of Network DDos–attacks ("Denial of Service"). Herald of Daghestan State Technical University. Technical Sciences. 2024; 51(2):137-153. DOI:10.21822/2073-6185-2024-51-2-137-153.

Введение. В течение последних лет во Всемирной глобальной сети Интернет участились случае осуществления кибератак, вызывающих недоступность ресурсов, и наиболее значимыми из них являются DDoS–атаки – *отказ в обслуживании.* Подобное положение дел может быть связано с геополитической ситуацией, и с тем фактом, что многие организации приняли решение перейти в дистанционный формат осуществления своей деятельности. По мере развития Интернета растёт и количество DDoS–атак, а также злоумышленников, которые их совершают.

Защита от сетевых DDoS–атак стала обычной рутиной для многих крупных компаний, и они вынуждены бороться с этой проблемой повседневно. Сегодня важно иметь хорошую защиту от подобных кибератак ещё до того, как нападение произойдет, иначе это может повлечь огромные потери для компании. Если Web–приложение или Web–сайт выйдут из строя, то это может создать у пользователей негативное отношение к ресурсу или организации в целом. Может появиться ощущение отсутствия профессионализма организации, а это, в свою очередь, может повлиять на множество других решений, таких, как потеря доверия к компании, утрата денежных средств, клиентов и их лояльности.

DDoS, сокращение от английского выражения «Distributed Denial of Service, распределенный отказ в обслуживании», представляет собой кибератаку, направленную на то, чтобы сделать сеть или Web–ресурс недоступными для обычных пользователей. Обычно это выполняется путём переполнения целевого хоста (сетевого интерфейса устройства, подключённого к IP–сети по протоколу TCP/IP) многочисленными запросами или обращениями. В DDoS–атаке требуется не один источник запросов, а от сотен до тысяч IP–адресов, в отличие от простых DoS-атак, в которых обычно участвует один пользователь. DDoS–атаки труднее отразить, чем DoS–атаки, из-за огромного количества устройств, участвующих в атаке.

Существуют различные типы кибератак типа «отказ в обслуживании»: на уровне приложений и на сетевом уровне. По данным Arbor Network(компании, занимающейся сетевой безопасностью), почти все DDoS-атаки, около 93%, в 2023 году были связаны с атаками на уровне приложений. В 2023 году целевой службой чаще всего является DNS, а не HTTP(-S).

Еще один важный аспект, когда речь заходит о DDoS–атаках, — это вопрос, как справляются с ними с точки зрения PR. Если переходить в социальные сети и распространять информацию о том, что Web–ресурс вашей компании временно не работает, то данные слухи быстро расползутся по сети, что может нанести компании огромные убытки и вызовет потерю репутации. Как правило, в таких случаях лучше быть открытым и прозрачным в отношении проблем, и информировать пользователей о том, что происходит. Защита от DDoS–атак сегодня необходима, как никогда, поскольку их количество продолжает расти немыслимыми темпами. Невозможно гарантировать безопасность на 100%, но можно заранее подготовиться к атаке. Наличие систем для мониторинга трафика, брандмауэр Web–приложений, ограничение скорости трафика, присутствие страницы состояния и кого-то из персонала, кто отвечает на вопросы в социальных сетях, — все это поможет обеспечить максимально эффективную защиту от DDoS-атак и их последствий.

Постановка задачи. Целью данного исследования является разработка искусственной нейронной сети для детектирования и нейтрализации сетевых DDoS–атак.

Для достижения цели необходимо решить следующие задачи: проанализировать существующие искусственные нейронные сети для определения их оптимальной структуры; исследовать современные методы детектирования сетевых DDoS–атак; собрать и доработать наборы данных (датасеты) для улучшения качества обучения ИНС; создать структуру искусственной нейронной сети классификатора; обучить ИНС и выполнить тестирование разработанного программного средства.

Методы исследования. Для успешного противодействия сетевым кибератакам разрабатываются методы и механизмы защиты,которые используют практически все современные программные и аппаратные средства защиты. Однако из-за высокой стоимости средств защиты многие компании воздерживаются от их покупки и эксплуатации, что приводит к значительным финансовым и клиентским потерям при сетевых атаках.

Одним из наиболее эффективных и перспективных методов обнаружения DDoS–атак является механизм искусственных нейронных сетей, который широко используется в современных защитных мерах.

Искусственная нейронная сеть – это набор алгоритмов, которые пытаются распознать основные взаимосвязи в наборе данных с помощью процесса, имитирующего работу человеческого мозга [1]. В этом смысле ИНС относятся к интеллектуальным системам, органическим или искусственным по своей природе. Искусственные нейронные сети могут адаптироваться к изменению входных данных; поэтому такая сеть может выдавать наилучший возможный результат, без необходимости переопределять выходные критерии. Концепция ИНС, основанная на искусственном интеллекте, сегодня стремительно развивается и захватывает всё новые сферы деятельности человека. Искусственная нейронная сеть работает аналогично нейронной сети человеческого мозга. «Нейрон» в нейронной сети – это математическая функция, которая собирает и классифицирует информацию в соответствии с определенной архитектурой.

Работа сети похожа на статистические методы – подгонку кривой и регрессионный анализ. ИНС содержит слои взаимосвязанных узлов. Каждый узел известен как *персептрон*, и напоминает множественную линейную регрессию. Персептрон передает сигнал, полученный с помощью этого метода, в функцию активации, которая является нелинейной. Нейронная сеть состоит из 3 основных элементов:

- набора синапсов–входных значений, предаваемых в нейрон, каждый синапс характеризуется своим весом;
- сумматора, складывающего значения входных сигналов, взвешенных относительно синапсов нейрона;
- функции активации, ограничивающей амплитуду выходного значения сигнала нейрона.

Random Forest – известный метод ансамблевого обучения в сфере Data Science, имеющий много преимуществ по сравнению с другими алгоритмами обработки данных. Этот метод прост в интерпретации, результативен и стабилен и часто используется в задачах регрессии и классификации. Таким образом, он охватывает большинство задач, решаемых с помощью искусственных нейронных сетей.

Этот алгоритм машинного обучения, предложенный в 1995 г. Тином Камом Хо и расширенный Лео Брейманом и Адель Катлер, основан на *деревьях*, которые часто называют *деревьями принятия решений*. Как следует из названия, дерево решений помогает специалисту по Data Science принять решение с помощью ряда вопросов (тестов), ответы на которые (да / нет) приводят к окончательному решению. Random Forest является методом ансамбля, он «собирает» или объединяет результаты, чтобы получить окончательный результат. Количество деревьев – это параметр, который настраивается путем *перекрестной проверки* – метода оценки алгоритма машинного обучения, который основывается на обучении и тестировании модели на фрагментах исходного ряда данных. Каждое дерево обучается подмножеству ряда данных и возвращает результат (да / нет).

Затем результаты всех деревьев решений объединяются, чтобы получить окончательный ответ. Каждое дерево «верно», и окончательный ответ – тот, у которого наибольшее количество голосов [2]. Это называется методом упаковки. Согласно ему, необходимо разделить выбранный набор данных на несколько случайно составленных подмножеств выборок, отсюда появляется «случайная» структура.

Для каждого подмножества обучается одна модель: существует столько же моделей, сколько и подмножеств. Все результаты моделей объединяются (например, с системой голосования), что дает нам конечный результат.

Этот алгоритм очень популярен из-за его способности комбинировать результаты своих деревьев для получения более надежного конечного результата. Благодаря своей эффективности он может использоваться во многих областях, например, в телефонном маркетинге для прогнозирования поведения клиентов, или даже в управлении финансовыми рисками. Пример схемы RandomForest представлен на рис. 1.

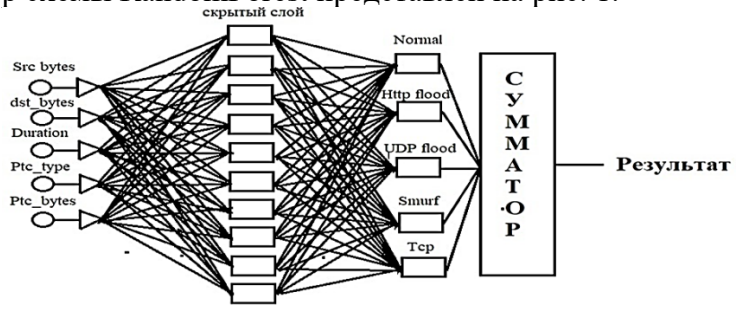

Рис. 1. Схема RandomForest
Fig. 1. RandomForest scheme

Таким образом, из нескольких моделей, не обязательно надежных, строится общая надежная модель, пригодная для использования в различных областях. На рис. 1 представлена схема работы сети. Данные из набора (датасета) подаются на вход искусственной нейронной сети. Количество подаваемых данных влияет на качественные характеристики ИНС. Чем больше подаваемых входных запросов, тем выше шанс определить вредоносный трафик. Затем данные попадают на скрытый слой, где «деревья» производят обработку данных и делаю первичную классификацию полученных данных на основе модели обучения и полученных данных по направлению возможных DDoS атак.

После этого все результаты со всех деревьев попадают в сумматор, где происходит «голосование», где каждое дерево выдает свой результат для каждого элемента множества и на основе большинства голосов сеть делает вывод, является ли запрос опасным или это обычный пользовательский запрос.

Машинное обучение (ML) – это технология искусственного интеллекта (ИИ), которая фокусируется на разработке систем, обучающихся или улучшающих производительность труда на основе наборов данных (датасетов). Управляют машинным обучением алгоритмы. Обычно используются два типа алгоритмов машинного обучения: обучение с учителем и обучение без учителя. Разница между ними определяется тем способом, с помощью которого каждый узнает о данных, на основе которых делаются прогнозы.

Контролируемое машинное обучение. Наиболее часто используются алгоритмы контролируемого машинного обучения. В этой модели специалист по обработке данных выступает в роли преподавателя и обучает алгоритм делать выводы. Когда алгоритм тренируется с учителем (преподавателем), он обучается с использованием набора данных, который уже обучен и имеет заранее определенный результат. Примерами контролируемого машинного обучения являются такие алгоритмы, как линейная и логистическая регрессия, многоклассовая классификация и методы эталонных векторов. При обучении алгоритмов с преподавателем создается модель прогнозирования на основе входных и выходных данных. Ключевое слово «под наблюдением» исходит из идеи наличия предварительно помеченного и классифицированного набора данных, то есть набора примеров, который уже знает, к какой группе, значению или категории принадлежат примеры. Эта группа данных,

которую специалист называет обучающими данными, корректирует предложенную исходную модель. Так, алгоритм «учится» классифицировать входные выборки, сравнивая результат модели с фактической меткой выборки, и выполняя соответствующие компенсации модели для каждой ошибки в оценке результата. Контролируемое обучение, например, может использоваться в робототехнике для программирования роботов, автономных (беспилотных) транспортных средств и др.

Неконтролируемое машинное обучение. В неконтролируемом машинном обучении используется более независимый подход, при котором компьютер учится выявлять сложные процессы и закономерности без пристального и постоянного руководства со стороны человека. Неконтролируемое машинное обучение включает обучение на основе данных, которые не имеют ни меток, ни определенного результата. Примерами бесконтрольных алгоритмов машинного обучения являются кластеризация *k-ближайших соседей*, анализ основных и независимых компонентов, а также законы ассоциации. Этот метод обучения состоит из искусственной нейронной сети, которая анализирует и использует большой объем информации для определения контекстов, закономерностей и сходств между данными. Этот процесс основан на различных процедурах, среди которых одним из методов, используемых в этой форме обучения, является групповой анализ или кластеризация.

В этом случае алгоритмы отвечают за автономное формирование групп, чтобы в конечном итоге сопоставить их с данными. Например, если данные являются фотографиями собак и кошек, программа сортирует все фотографии собак в одну категорию, а все фотографии кошек – в другую, во время обучения без присмотра. Однако, в отличие от занятий с преподавателем (учителем), эта классификация не дается заранее. При неконтролируемом обучении алгоритмы принимают эти решения независимо друг от друга на основе сходства и различий между фотографиями. Таким образом, задача алгоритмов состоит в том, чтобы найти отношения между объектами без необходимости какого-либо сходства между ними. Опять же, можно взять пример с изображениями собак: алгоритм обучения без присмотра не группирует всех собак, используя ассоциации, но, например, сопоставляет собак с поводками.

По статистике, около 33% компаний попадают под DDoS-атаки. Предсказать атаку невозможно, а некоторые из них могут быть действительно мощными и достигать 300–500 Гбит/с. Чтобы защититься от DDoS–атак, можно воспользоваться услугами специализированных сервисов.*Cloudflare.* Один из самых известных защитных сервисов. При использовании тарифов FREE и PRO можно рассчитывать на базовую защиту от DDoS-атак. Для более надёжной защиты от атак 3, 4 и 7 уровней модели OSI нужно подключить бизнес или корпоративный аккаунт. В Cloudflare используется фиксированная оплата, то есть, независимо от того, сколько и каких уровней будут атаки, клиент платит одну и ту же сумму. Среди клиентов этого сервиса такие крупные компании, как Nasdaq, DigitalOcean, Cisco, Salesforce и Udacity. Сеть Cloudflare представлена в 102 дата–центрах с пропускной способностью более 10 Тбит/с и способна устранять любые DDoS–атаки, в том числе DNS и Smurf–атаки. У этого сервиса есть круглосуточная служба экстренной помощи, куда пользователь может обратиться, если против него происходит DDoS–атака.

Incapsula. Этот сервис предлагает комплексную защиту от DDoS–атак различных типов. Сервис может работать постоянно или по запросу, и обнаруживать любые DDoS–атаки. Сеть Incapsula состоит из 32 дата-центров с пропускной способностью в 3 Тбит/с.

У Incapsula есть пробная версия для бизнеса с SSL протоколом, CDN и WAF, которая защитит от DDoS–атак. На случай, если атака уже совершена и счёт идёт на минуты, можно воспользоваться сервисом экстренной помощи UnderAttack. Этот сервис стоит достаточно дорого относительно своих аналогов и на данный момент недоступен в России.

Akamai. Этот сервис является одним из лидеров в сфере обеспечения безопасности. По заявлениям руководства Akamai, сервис может справиться с атакой в 1,3 терабита в секунду. Сервис построен на интеллектуальной платформе Akamai и оказывает

поддержку круглосуточно. Защитить сайты можно от всех известных DDoS–атак, включая зашифрованный трафик. Имеется 1300 сетевых адресов в более 100 странах.

Qrator. Это один из наиболее известных российских ресурсов по борьбе с DDoS, ведет блог на Хабре. Особенностью работы системы является «прозрачность» для легитимных пользователей – их она выявляет с помощью алгоритмов умной фильтрации, не заставляя вводить *капчу*, или другим образом подтверждать свою легитимность [3]. Даже при необходимости отражения атаки на уровне приложений (7 уровень модели OSI) не требуется тонкой настройки продукта при обнаружении атаки система перейдет в нужный режим автоматически [4]. Для подключения защиты системы Qrator нужно изменить А-запись сайта. К недостаткам этой системы можно отнести высокую стоимость использования продукта, но компания предоставляет SLA (если уровень услуги не обеспечен, платить не обязательно) и бесплатный тестовый период в семь суток.

*AWS Shield Advanced.*Сервис Shield предназначен для защиты приложений, работающих на платформе AWS. Он бесплатен, но для улучшенной защиты следует перейти на тариф Shield Advanced. В продвинутую версию добавлена проверка сетевого потока и мониторинг трафика прикладного уровня, защита от большего количества DDoS–атак, блокировка нежелательного трафика, анализ состояния информационной системы после DDoS–атаки, при круглосуточном режиме работы. Все эти службы защиты от DDoS–атак предназначены для блогеров, электронной коммерции, малого и среднего бизнеса. Для бизнеса более крупных масштабов существуют другие сервисы, с более высоким уровнем защиты и большим функционалом.

В качестве источника DDoS–атак киберпреступники тайно используют целые сети зараженных устройств. Часто их владельцы даже не знают, что их устройства и IP-адреса участвуют в подобной атаке. Устройства *Интернета Вещей* (IoT) особенно уязвимы для таких атак. Их количество постоянно растет, но уровень безопасности остается недостаточным. Почти половина всех DDoS–атак является смешанными атаками, но в их классификации выделяется три основные категории:

Объемные атаки (Mass Flood – «массовое затопление»), выполняют большое количество запросов к серверу. После этого всплывающий трафик переполняет всю пропускную способность сети, со скоростью до нескольких терабит в секунду, и вскоре компьютерная система перестает отвечать на запросы.

Протокольные атаки – это особый тип DDoS–атак, который использует уязвимости в сетевых протоколах, таких как TCP, UDP или ICMP. Они перегружают сеть избирательными действиями [5].

Атаки на уровне приложений блокируют Web–серверы и приложения. Внешние запросы вызывают большое количество внутренних запросов, что приводит к недоступности сервера.

UDP Flood (*User Datagram Protocol,* протокол пользовательских дейтаграмм) – это атака типа «отказ в обслуживании», при которой большое количество UDP–пакетов отправляется на целевой сервер, чтобы подавить обработку и отклик этого устройства. Брандмауэр, защищающий целевой сервер, также может выйти из строя из-за переполнения UDP, что приведет к отказу службы в законном трафике [6].

Атака UDP, в основном, работает с шагами, которые сервер выполняет при ответе на UDP–пакет, отправленный на один из его портов. При нормальных обстоятельствах, когда сервер получает пакет UDP на определенном порту, он в ответ проходит два шага [7]: сервер проверяет, запущены ли программы, которые в настоящее время ожидают запросов на указанном порту. Если программа не получает пакеты на этом порту, то сервер отвечает пакетом ICMP (Ping), чтобы сообщить отправителю, что пункт назначения недоступен.

Атаку UDP можно сравнить с переадресацией вызовов администраторами отеля. Секретарь получает телефонный звонок, в котором вызывающий абонент просит подключиться к определенной комнате. Затем секретарь должен просмотреть список всех комнат, чтобы убедиться, что гость свободен и находится в комнате, чтобы ответить на звонок.

Как только секретарь обнаружит, что гость не отвечает на звонки, он должен снова снять трубку и сообщить вызывающему абоненту, что гость не отвечает на звонок. Если вдруг все телефонные линии загораются одновременно с одними и теми же запросами, то они быстро перегорают. Когда сервер получает каждый новый UDP–пакет, то он проходит этапы обработки запроса, используя ресурсы сервера.

При передаче UDP–пакетов каждый пакет содержит IP–адрес исходного устройства. В этом типе DDoS–атаки злоумышленник обычно не использует свой собственный реальный IP–адрес, а вместо этого подделывает исходный IP–адрес UDP–пакетов, предотвращая раскрытие истинного местоположения хакера и, возможно, загромождение пакетов ответа целевого объекта, что отображено на рис. 2. Этот тип атаки сегодня является одним из распространённых, потому что хакеру его легко организовать.

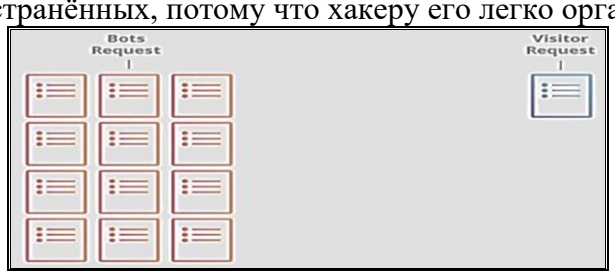

Рис. 2. Пример атаки UPD–Flood
Fig. 2. Example of a UDP-Flood attack

Поскольку целевой сервер использует ресурсы для проверки, а затем отвечает на каждый полученный UDP–пакет, ресурсы назначения могут быстро истощаться при получении большого потока пакетов UDP, что приводит к отказу в обслуживании для обычного трафика. Большинство операционных систем частично ограничивают скорость отклика пакетов ICMP, чтобы избежать DDoS-атак, требующих ответа ICMP. Недостатком этого типа является то, что законные пакеты также могут быть отфильтрованы во время атаки. Если поток UDP имеет достаточный объём для заполнения таблицы состояний брандмауэра целевого сервера, то любое смягчение атаки, происходящее на уровне сервера, будет недостаточным.

HTTP–Flood. Кибератаки этого типа HTTP основаны на запросах клиента GET или POST. Клиент(то есть браузер), который хочет получить доступ к Web–сайту, должен отправить один из этих запросов. Сервер обрабатывает его, и, в свою очередь, отправляет ответ клиенту. Запросы GET извлекают статический контент, такой, как изображения или текстовые блоки. Вместо этого, запросы POST используются для доступа к динамическим ресурсам. Другими словами, метод GET получает данные с сервера, а метод POST отправляет данные на сервер. Оба могут использоваться для выполнения такого рода атак, хотя метод POST используется чаще, поскольку он требует сложной обработки сервером.

При атаке HTTP–Flood многие из этих запросов выполняются одновременно и в течение длительного периода времени. Сеть ботов, или ботнет, часто используется для увеличения количества запросов. Атака типа HTTP–Flood предназначена для того, чтобы сервер выделял как можно больше ресурсов для каждого запроса. В обычной ситуации это желательно, потому что обычно сервер не получает тысячи или сотни тысяч запросов в минуту, как в этом случае.

Таким образом, в этом случае злоумышленнику просто нужно подождать, пока сервер не переполнится, а затемWeb–приложение или Web–сайт выйдут из строя. Во время атаки тысячи ботов отправляют несколько запросов в секунду и отключают сервер или сеть. Страницы иногда временно генерируют большой трафик, поэтому трудно сказать, увеличивается ли количество посещений из-за атаки, или только из-за хорошей маркетинговой кампании. Если обнаружится, что это атака наводнения HTTP, брандмауэры могут идентифицировать и блокировать подозрительные IP–адреса. В этой ситуации, первым шагом является отправка клиенту так называемой задачи вычисления JavaScript, метода, который позволяет определить, является ли клиент частью ботнета или обычного

пользователя. Этот тест, в отличие от роботов, может пройти любой браузер обычного посетителя Web–сайта. HTTP Flood может быть произведен с обычного устройства, имеющего выход в глобальную сеть Интернет.

Зная модель злоумышленника, можно ввести в брандмауэр несколько простых правил, чтобы автоматически блокировать IP–адреса ботнета. HTTP–Flood обычно может быть обнаружен и остановлен в течение нескольких минут, если он был идентифицирован, как причина сбоя системы. Защита от наводнения HTTP очень сложна, потому что запросы злоумышленника могут выглядеть как обычный трафик Web–сайта. Этот тип атаки не отправляет вредоносное ПО на сервер и не предназначен для использования каких-либо нарушений безопасности, но сервер насыщен законными требованиями. Поскольку для этого требуется гораздо меньшая пропускная способность, чем любое проникновение кода на страницу, этот тип атаки может сначала остаться незамеченным. Чтобы защитить себя, большинство Web–сайтов используют тест *Captcha*, который может пройти только человек–пользователь. Таким образом, можно узнать, поступает ли запрос из ботнета, и, таким образом, заблокировать соответствующие IP–адреса. Существуют также брандмауэры приложений и Web–сайтов, которые проверяют и анализируют трафик. Эти системы могут немного замедлить работу Web–сайта, но гарантируют его защиту и стабильность. Если сама Web–страница требует обработки большого объема данных, можно просмотреть страницу загрузки, пока главная страница загружается в фоновом режиме.

SYN–Flood – это метод, который позволяет пользователю враждебной клиентской программы выполнить атаку типа «отказ в обслуживании» (DoS) на компьютерный сервер. Вражеский клиент повторно отправляет пакеты SYN (синхронизации) на каждый порт сервера, с поддельными IP–адресами, что приводит к перегрузке системы и замедлению реакции сервера на законный трафик или полному прекращению ответа.

Это возможно потому, что SYN–атаки атакуют процесс взаимодействия с TCP–соединением. При нормальных обстоятельствах, TCP–соединение указывает на три различных процесса, которые необходимо выполнить. Клиент сначала отправляет SYN–пакет на сервер, чтобы инициировать соединение. Затем сервер отвечает на этот исходный пакет пакетом SYN/ACK для подтверждения связи.

Наконец, клиент возвращает пакет ACK, чтобы подтвердить получение пакета от сервера. После завершения этой последовательности отправки и получения пакетов открывается TCP- соединение, и тогда можно отправлять и получать данные. Клиент злоумышленника делает все запросы SYN действительными, но поскольку IP–адреса являются поддельными, сервер не может закрыть соединение, отправив первые пакеты обратно вражескому клиенту. Вместо этого соединение остается открытым. До истечения тайм–аута от враждебного клиента приходит еще один SYN–пакет. Такое соединение называется *полуоткрытым соединением*. В этих условиях сервер полностью или почти полностью занят враждебным клиентом. Общение с законными клиентами затруднено или невозможно. Клиент злоумышленника может воспользоваться полуоткрытыми соединениями, и, возможно, получить доступ к файлам на Web–сервере. Отправка пакетов SYN вражескому клиенту для поиска открытых портов и взлома одного или нескольких из них, называется анализом SYN. Вражеский клиент всегда знает, открыт ли порт, когда сервер отвечает пакетом SYN/ACK.

Существует три основных типа SYN–атак:

- прямая атака - злоумышленник не скрывает свой оригинальный IP;
- спуфинг–атака, при которой создается несколько поддельных IP–адресов;
- распределенная атака (DDoS) – используются боты, чтобы затруднить отслеживание источника атаки.

SYN–атаки можно предотвратить следующими способами:

1) Увеличение очереди журналов. Увеличивая максимальное отставание, можно гарантировать место для законного пользовательского трафика в середине кибератаки.

2) Повторное использование TCP–соединения, относится к перезаписыванию самого старого полуоткрытого соединения после завершения отставания, чтобы установить соединения с законными пользователями за меньшее время, чем потребовалось бы для заполнения отставания пакетами SYN.

3) Файлы cookie – эти файлы удаляют большую часть невыполненной работы и восстанавливают ее только после подтверждения законности запроса.

ICMP. Атака ICMP (Internet Control Message Protocol): как следует из названия, она предназначена для отправки потока ping-пакетов на целевой хост, что делает целевой хост занятым обработкой ping–пакетов и неспособным обрабатывать другие обычные запросы. Это напоминает переполнение узла назначения ping–пакетами. Для выполнения ICMP–атаки необходимы 3 резерва знаний: принцип DOS–атаки; глубокое понимание работы протокола ICMP; навыки программирования для необработанных сокетов.

Атака ICMP формируется на основе ping, но сам запрос ping редко вызывает проблемы с целью и временем простоя. Это связано с тем, что скорость отправки ping–пакетов слишком низка. Ограничения отправки запросов ping в 1 секунду, что более чем достаточно для хоста назначения для обработки ping– пакетов. Чтобы вызвать явление «наводнения», существует 3 метода атаки ICMP–Flood.

Ping of Death. Следует отметить, что кибератака *Ping of Death(эхо-запрос смерти)* является одной из старейших атак на корпоративную сеть. Злоумышленник отправляет вредоносный пакет данных. Как только система получателя попытается его обработать, появится ошибка, которая приведет к сбою компьютера.Эта атака создает IP–диаграмму, превышающую максимально допустимый размер 65 536 байт. Он отправляется в систему с уязвимым стеком TCP/IP и приводит к зависанию системы. Следует отметить, что сегодня эта проблема больше не может затронуть большинство компьютеров. Этот тип атаки был широко распространен ранее, особенно в 1990-х и 2000-х годах, и с тех пор операционные системы исправили эту уязвимость.

Отправляя эхо-запросы такого размера, злоумышленник лишает компьютеры возможности отвечать. Этаситуация может произойти, если будут отправлены фрагменты пинга (эхо–размера) такого размера, которые превышают 65 536 байт. Когда команда пытается собрать воедино этот полученный пакет и все его фрагменты, происходит переполнение сетевого стека компьютера. Это приводит к сбою системы. Эта уязвимость ранее присутствовала во всех типах операционных систем и устройств, и повлияла на несколько версий операционных систем Microsoft Windows, Unix, Linux, Mac и др., и даже на маршрутизаторы и принтеры. Со временем, для решения этой проблемы в ОС были выпущены исправления. Однако «*смертный приговор*» может быть вынесен в более позднем случае. В 2013 году Microsoft выпустила исправления безопасности для протокола ICMPv6, которое затронуло все версииоперационной системы, за исключением Windows XP и Windows Server 2012. Эта уязвимость была зарегистрирована как CVE-2013-3183.

Команда Ping появилась в 1980-х годах, использовала протокол ICMP и использовалась для сбора сетевой информации. Несколько лет спустя, многие пользователи поняли, что можно отправить несколько пакетов ICMP размером более 65 536 байт. Это перегружало систему и выводило её из строя. Операционные системы не были готовы к такой атаке. Это означало, что любой пользователь может посылать эхо-запросы и вызывать смерть системы, что могло отключать многие компьютеры за короткое время. Эта проблема затронула многие компьютеры по всему миру. Операционные системы исправили эту уязвимость, но на данный момент нет гарантий, что снова не произойдет ошибка, аналогичная той, что была произошла в Windows в 2013 году. Можно сказать, что этот тип угрозы стал источником DDoS-атак, наблюдаемых сегодня в сети. Эти новые и более сложные атаки могут парализовать важные Web–серверы. Чтобы избежать таких проблем,

лучше всегда правильно обновлять свой компьютер, хотя сегодня такие атаки являются исключением. Во многих случаях возникают ошибки, которые могут быть использованы злоумышленниками. Чрезвычайно важно всегда иметь последние версии операционных систем и быть максимально защищенным.

Обсуждение результатов. Программное средство и модель искусственной нейронной сети разрабатывались с использованием языка программирования Python в среде, поддерживающей функции обучения нейронных сетей PyCharm.

PyCharm – это самая интеллектуальная интегрированная среда разработки (IDE) языка Python с полным набором средств эффективной разработки. Существует в двух вариантах – бесплатная версия PyCharm Community Edition, и поддерживающая больший набор возможностей PyCharm Professional Edition. PyCharm выполняет инспекцию кода на лету, обеспечивает автодополнение, в том числе основываясь на информации, полученной во время исполнения кода, реализовывает навигацию по коду и перепроектирование исходного кода (рефакторинг). Для разработки данного проекта использовалась версия PyCharm Community Edition, поскольку в ней находятся все основные функции, которые необходимы для дальнейшей разработки.

Преимущества такой среды разработки – облегченная IDE для разработки на языке Python, являющаяся бесплатной версией PyCharm с открытым кодом под лицензией Apache 2; внутренний текстовый редактор, понимающий контекст, отладчик с возможностью рефакторинга и инспекций, интеграция с системой контроля версий (VCS); навигация по проекту, поддержка тестирования, настраиваемый интерфейс пользователя (UI), горячие клавиши специализированного текстового редактора Vim (Vi Improved) [8].

Отбор данных. На рис. 3 можно увидеть подключение необходимых модулей через Import, там используются основные библиотеки, такие как Sklearn, Pandas. Numpy, Matplotlib, Seaborne.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import itertools


from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression


from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
```

Рис. 3. Подключение модулей
Fig. 3. Connecting modules

На рис.4 продемонстрирован программный код,отвечающий за подключение датасетов, а также присвоение колонкам названия, поскольку в самом датасете они отсутствуют [9].

```
file_path_20_percent = 'KDDTrain+_20Percent.txt'
file_path_full_training_set = 'KDDTrain+.txt'
file_path_test = 'KDDTest+.txt'


df = pd.read_csv(file_path_full_training_set)
test_df = pd.read_csv(file_path_test)

columns = (['duration','protocol_type','service','flag','src_bytes','dst_bytes','land','wrong_fragment'
,'urgent','hot','num_failed_logins','logged_in','num_compromised','root_shell','su_attempted','num_root',
'num_file_creations','num_shells','num_access_files','num_outbound_cmds','is_host_login','is_guest_login','count',
'srv_count','serror_rate','srv_serror_rate','rerror_rate','srv_rerror_rate','same_srv_rate','diff_srv_rate',
'srv_diff_host_rate','dst_host_count','dst_host_srv_count','dst_host_same_srv_rate','dst_host_diff_srv_rate',
'dst_host_same_src_port_rate','dst_host_srv_diff_host_rate','dst_host_serror_rate','dst_host_srv_serror_rate',
'dst_host_rerror_rate','dst_host_srv_rerror_rate','attack','level'])
```

Рис. 4. Подключение датасетов и название колон
Fig. 4. Connecting datasets and naming columns

На рис. 5 показан датафрейм с названием колонок в отличии от исходных данных.

	duration	protocol_type	service	...	attack	level	attack_flag
0	0	udp	other	...	normal	15	0
1	0	tcp	private	...	neptune	19	1
2	0	tcp	http	...	normal	21	0
3	0	tcp	http	...	normal	21	0
4	0	tcp	private	...	neptune	21	1

Рис. 5. Датафрейм данных
Fig. 5. Dataframe data

На рис. 6 показано создание списка для хранения классификаций атак. Первые преобразования касаются поля атаки. Сначала нужно добавить столбец, в котором кодируются «нормальные» значения 0, причём любое другое значение интерпретируется как 1. Программа будет использовать его в качестве классификатора для простой бинарной модели, которая идентифицирует любую атаку. Далее необходимо классифицировать каждую из кибератак в соответствии с типом DDoS–атаки для создания более детализированной модели прогнозирования.

```
# Классификация атак

dos_attacks = ['apache2', 'back', 'land', 'neptune', 'mailbomb', 'pod', 'processtable', 'smurf', 'teardrop', 'udpstorm',
               'worm']
probe_attacks = ['ipsweep', 'mscan', 'nmap', 'portsweep', 'saint', 'satan']
privilege_attacks = ['buffer_overflow', 'loadmdoule', 'perl', 'ps', 'rootkit', 'sqlattack', 'xterm']
access_attacks = ['ftp_write', 'guess_passwd', 'http_tunnel', 'imap', 'multihop', 'named', 'phf', 'sendmail',
                  'snmpgetattack', 'snmpguess', 'spy', 'warezclient', 'warezmaster', 'xclock', 'xsnoop']


attack_labels = ['Normal', 'DoS', 'Probe', 'Privilege', 'Access']


# Функция для перехода к фреймам данных
def map_attack(attack):
    if attack in dos_attacks:
        attack_type = 1
    elif attack in probe_attacks:
        attack_type = 2
    elif attack in privilege_attacks:
        attack_type = 3
    elif attack in access_attacks:
        attack_type = 4
    else:
        attack_type = 0

    return attack_type


# сравнение и присоединение
attack_map = df.attack.apply(map_attack)
df['attack_map'] = attack_map

test_attack_map = test_df.attack.apply(map_attack)
test_df['attack_map'] = test_attack_map
```

Рис. 6.Классификация атак
Fig. 6. Classification of attacks

На рис. 7 продемонстрирован датафрейм с двумя новыми столбцами таблицы.

	duration	protocol_type	service	...	level	attack_flag	attack_map
0	0	udp	other	...	15	0	0
1	0	tcp	private	...	19	1	1
2	0	tcp	http	...	21	0	0
3	0	tcp	http	...	21	0	0
4	0	tcp	private	...	21	1	1

Рис. 7. Датафрейм
Fig. 7. Dataframe

Некоторые предварительные исследования того, что есть в наборе, показывают, что: во–первых, это простая таблица DDoS–атак по протоколам. Протокол анализа сетевого трафика – это простой инструмент для создания некоторых начальных сегментов для категоризации данных. «Нормальный» сегмент остается в наборе, на данный момент в качестве эталона. Процесс создания таблицы представлена на рис. 8 [10].

```
# Таблица атак по протоколам
attack_vs_protocol = pd.crosstab(df.attack, df.protocol_type)
print(attack_vs_protocol)
```

Рис. 8. Создание таблицы представления
Fig. 8. Create a view table

Сам датафрейм изображен на рис. 9. Он показывает, какой трафик проходит через протоколы, и что необходимо знать для определения уязвимости протоколов, и для каких протоколов нужно использовать защиту. Таблица даёт понимание того, какие DDoS–атаки

приходятся на различные протоколы. Для разработчика программ это важно, чтобы определить типы DDoS–атак, и по каким протоколам они происходят.

protocol_type	icmp	tcp	udp
attack			
back	0	956	0
buffer_overflow	0	30	0
ftp_write	0	8	0
guess_passwd	0	53	0
imap	0	11	0
ipsweep	3117	482	0
land	0	18	0
loadmodule	0	9	0
multihop	0	7	0
neptune	0	41214	0
nmap	981	265	247
normal	1309	53599	12434
perl	0	3	0
phf	0	4	0
pod	201	0	0
portsweep	5	2926	0
rootkit	0	7	3
satan	32	2184	1417
smurf	2646	0	0
spy	0	2	0
teardrop	0	0	892
warezclient	0	890	0
warezmaster	0	20	0

Рис. 9. Датафрейм DDoS–атак
Fig. 9. Dataframe of DDoS attacks

Датафрейм помогает увидеть, что большинство DDoS–атак нацелены на определенный протокол. Однако существует ряд DDoS–атак(satan, nmap, ipsweep), представляющих собой кросс-протокольные атаки. Можно обратить внимание, что данные icmp реже встречаются в обычном трафике. Эти данные представлены на рис. 13 в виде диаграммы для более удобного восприятия, программная реализация самой диаграммы на рис. 10, результат работы можно увидеть на рис. 11.

```
def bake_pies(data_list, labels):
    list_length = len(data_list)

    color_list = sns.color_palette()
    color_cycle = itertools.cycle(color_list)
    cdict = {}

    fig, axs = plt.subplots(1, list_length, figsize=(18, 10), tight_layout=False)
    plt.subplots_adjust(wspace=1 / list_length)

    for count, data_set in enumerate(data_list):

        for num, value in enumerate(np.unique(data_set.index)):
            if value not in cdict:
                cdict[value] = next(color_cycle)

        wedges, texts = axs[count].pie(data_set,
                                       colors=[cdict[v] for v in data_set.index])

        axs[count].legend(wedges, data_set.index,
                          title="Flags",
                          loc="center left",
                          bbox_to_anchor=(1, 0, 0.5, 1))

        axs[count].set_title(labels[count])

    return axs

icmp_attacks = attack_vs_protocol.icmp
tcp_attacks = attack_vs_protocol.tcp
udp_attacks = attack_vs_protocol.udp
```

Рис. 10. Функция диаграммы
Fig. 10. Chart function

Графики позволяют наглядно посмотреть на данные [11].

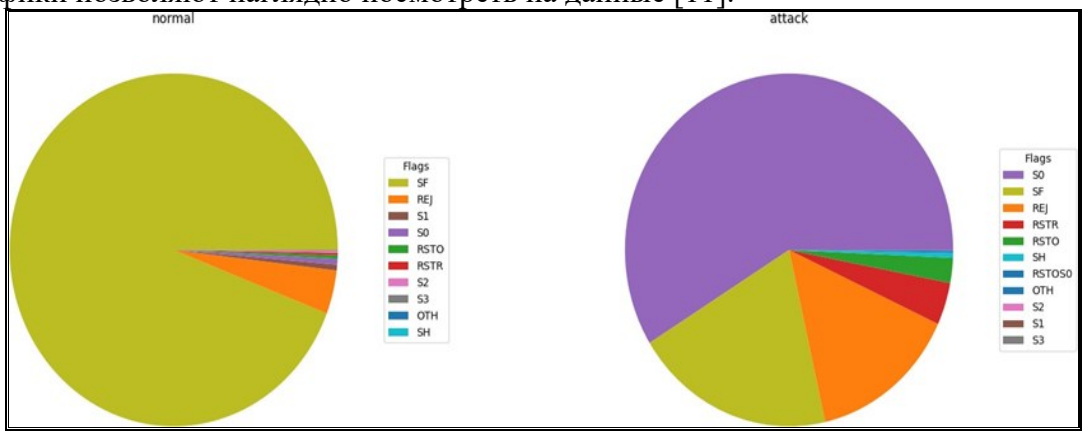

Рис. 11. Диаграмма трафика
Fig. 11. Traffic diagram

На диаграмме видно, сколько сервисов находятся в наборах DDoS–атаки. Большое количество трафика приходится на Http, в отличии от атакующего трафика, который направлен во всех направлениях. Это означает, что вредоносный трафик использует множество различных способов попасть в систему.

Следует отметить разность протоколов – тот факт, что каждый протокол может быть полезен для определенного типа трафика. Если подумать об этом с точки зрения сетевого администратора, то комбинация протокола, флага и службы показывает характер трафика. Связывание их с продолжительностью соединения и объемом данных в этом соединении является хорошей отправной точкой для дальнейшего анализа.

Разработка функций. Для создания функций используются следующие данные: protocol_type, service и flag. Между ними достаточно различий для получения базового уровня идентификации. Можно добавить пункты: durations, src_bytes, dst_bytes. Все они легко доступны на современном сетевом оборудовании и могут многое рассказать о том, что происходит в сети. Код разработки показан на рис. 12.

```
#Разработка функций
features_to_encode = ['protocol_type', 'service', 'flag']
encoded = pd.get_dummies(df[features_to_encode])
test_encoded_base = pd.get_dummies(test_df[features_to_encode])


test_index = np.arange(len(test_df.index))
column_diffs = list(set(encoded.columns.values)-set(test_encoded_base.columns.values))

diff_df = pd.DataFrame(0, index=test_index, columns=column_diffs)


column_order = encoded.columns.to_list()


test_encoded_temp = test_encoded_base.join(diff_df)

test_final = test_encoded_temp[column_order].fillna(0)


numeric_features = ['duration', 'src_bytes', 'dst_bytes']
```

Рис. 12. Получение базового уровня идентификации
Fig. 12. Obtaining a Basic Level of Identification

pd.get_dummies – это метод, который позволяет быстро выполнять кодирование данных для столбцов. Этот метод берет каждое значение, которое он находит в одном столбце, и создает для него отдельный столбец, где 0 или 1 указывает, является ли этот столбец «горячим». Каждое значение находится в тестовых данных. Так, создаются различные формы фрейма данных [12].Все данные, которые подаются на вход сети, могут изменяться. Чем больше параметров, относящихся непосредственно к запросам, будет подано, тем лучше будет обучен классификатор, но при этом упадет скорость обработки всего датасета.

Необходимо определить цель для дальнейшей классификации, создать два тренировочных набора: бинарный и мультиклассификационный, как показано на рис.13.

```
#Создание 2 целей для классификации
binary_y = df['attack_flag']
multi_y = df['attack_map']

test_binary_y = test_df['attack_flag']
test_multi_y = test_df['attack_map']

# Построение тренировочного набора
binary_train_X, binary_val_X, binary_train_y, binary_val_y = train_test_split(to_fit, binary_y, test_size=0.6)
multi_train_X, multi_val_X, multi_train_y, multi_val_y = train_test_split(to_fit, multi_y, test_size=0.6)
```

Рис. 13. Построение тренировочного набора
Fig. 13. Building a training set

Построение модели. Основываясь на характере данных, RandomForrest является оптимальной моделью искусственной нейронной сети для классификатора, а также для построения прогностических моделей дерева решений.

Построение модели ИНС показано на рис. 14. Данная модель наибольшим образом подходит для классификации DDoS–атак и показывает хороший результат.

```
base_rf_score = accuracy_score(binary_predictions,binary_val_y)
print(base_rf_score)

models = [
    RandomForestClassifier(),
    LogisticRegression(max_iter=250),
    KNeighborsClassifier(),
]
```

Рис. 14. Создание модели RandomForest
Fig. 14. Creating a RandomForest model

Точность модели в процентах показана на рис. 15.

```
0.9929614733276884
```

Рис. 15. Точность обучения модели
Fig. 15. Model training accuracy

Анализ прогнозов. Создание вспомогательной функции для извлечения важных показателей показано на рис. 16. Нужно добавить прогнозируемые и фактические результаты к данным [13].

```
#Вспомогательная функция
def add_predictions(data_set, predictions, y):
    prediction_series = pd.Series(predictions, index=y.index)

    # добавление прогнозируемых и фактических результатов к данным
    predicted_vs_actual = data_set.assign(predicted=prediction_series)
    original_data = predicted_vs_actual.assign(actual=y).dropna()
    conf_matrix = confusion_matrix(original_data['actual'],
                                   original_data['predicted'])

#Захват строк с неудачным прогнозом
    base_errors = original_data[original_data['actual'] != original_data['predicted']]

    # сброс колонок с 0 значением
    non_zeros = base_errors.loc[:, (base_errors != 0).any(axis=0)]

    false_positives = non_zeros.loc[non_zeros.actual == 0]
    false_negatives = non_zeros.loc[non_zeros.actual == 1]


    prediction_data = {'data': original_data,
                       'confusion_matrix': conf_matrix,
                       'errors': base_errors,
                       'non_zeros': non_zeros,
                       'false_positives': false_positives,
                       'false_negatives': false_negatives}

    return prediction_data
```

Рис. 16. Добавление прогнозируемых и фактических результатов
Fig. 16. Adding predicted and actual results

После этого можно построить матрицу с результатами, в которой искусственная нейронная сеть сопоставит предсказанную классификацию с фактической классификацией. матрица показана на рис. 17.

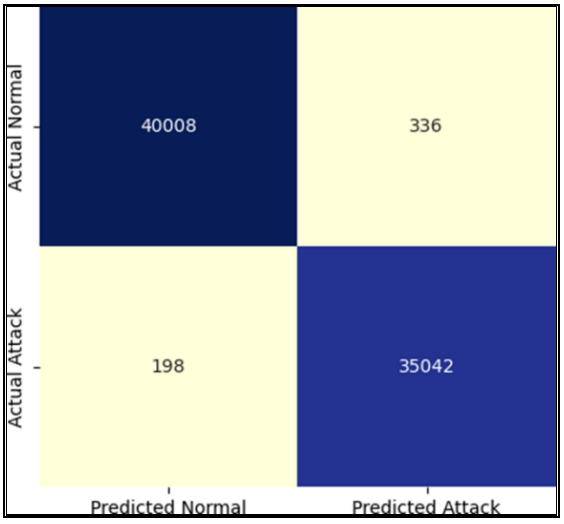

Рис. 17. Матрица результатов обучения
Fig. 17. Learning Outcomes Matrix

На рис. 18 показана работа искусственной нейронной сети на тестовых данных, и классификацияDDoS–атак по основным направлениям. Матрица изменяется в соответствии с данными, которые находились в датасете.

В этом случае в датасете было около 30 различных типов трафика, разбитых по основным типам DDoS–атак.

При переобучении на другом датасете, либо при увеличении текущего датасета нужно изменить группы DDoS–атак и пересоздать данную матрицу. Матрица, по вертикали, показывает предсказанный опасный и нормальный трафик, а по горизонтали показывает действительный опасный и безопасный трафик [14].

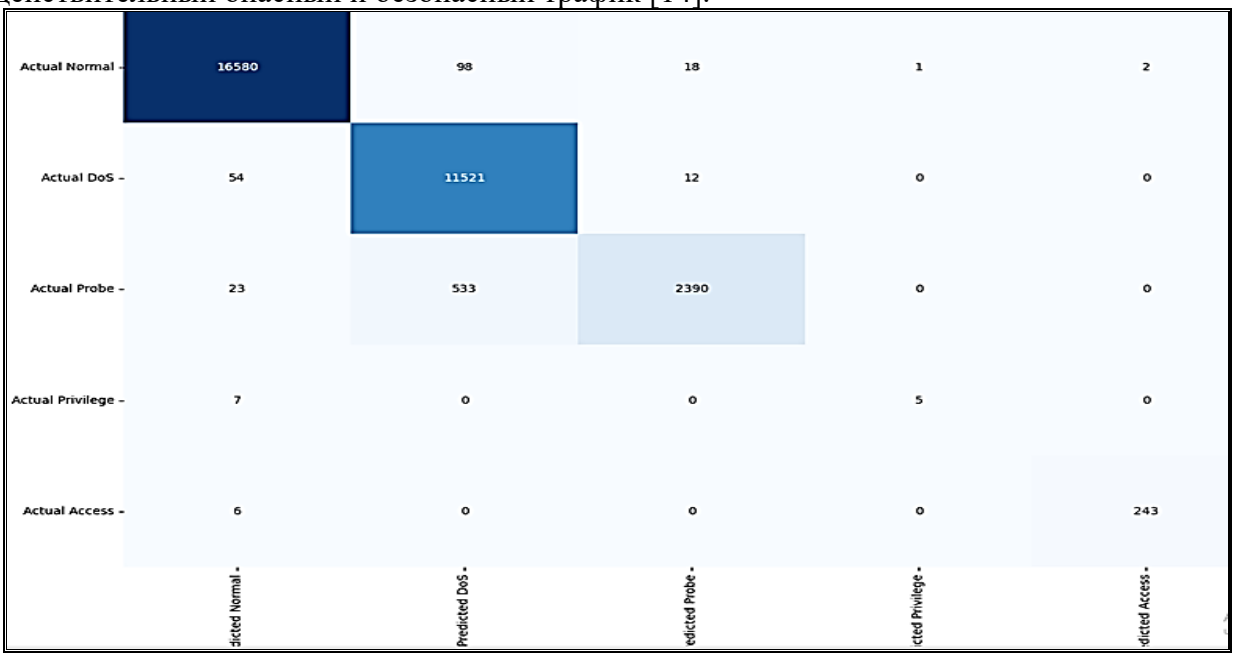

Рис. 18. Матрица обучения классификации
Fig. 18. Classification learning matrix

Механизм блокировки. На рис. 19 показан основной механизм блокировки. Он заключается в том, что вредоносные IP помещаются в файл .htaccess, который блокирует какие-либо действия с данного IP–адреса.

```
<?php
require "data.php";
$ip=$_SERVER['REMOTE_ADDR'];
$zapros = $_SERVER['REQUEST_METHOD'];
$nowtime = date("Y-m-d H:i:s");
$oldtime = date("Y-m-d H:i:s", strtotime(date("Y-m-d H:i:s"))-$second );
$url = $_SERVER['REQUEST_URI'];
$file = $_SERVER["DOCUMENT_ROOT"] . "/.htaccess";
$f_LOG = $_SERVER["DOCUMENT_ROOT"] . "/logs/logs.txt";

$result = mysqli_query($link, "INSERT INTO `antiddos` (`ip`,`time`) VALUES ('$ip','$nowtime')");
$result = mysqli_query($link, "DELETE FROM `antiddos` WHERE `time` BETWEEN '2020-01-01 00:00:00' and '{$oldtime}'");
$result = mysqli_query($link, "SELECT COUNT(*) FROM `antiddos` WHERE `ip`='$ip' and `time` BETWEEN '{$oldtime}' and
if ($row=mysqli_fetch_row($result)){
    $count = $row[0];
    if ((int)$count > $limit){
        $f = fopen($file, "a");
        fwrite($f, "Deny from $ip\n" );
        fclose($f);
        $result = mysqli_query($link, "DELETE FROM `antiddos` WHERE `ip`='$ip'");
    }
}

if ($zapros == "GET") { $array = json_encode($_GET); }
elseif ($zapros == "POST") { $array = json_encode($_POST); }
if (empty($array)) { $array = ""; }
$f = fopen($f_LOG, "a");
fwrite($f, "$nowtime | $zapros $ip $url $array\n" );
fclose($f);

?>
```

Рис. 19. Механизм блокировки
Fig. 19. Locking mechanism

На рис. 20 показано основное содержание файла .htaccses, а также заблокированные IP–адреса во время работы системы защиты от DDoS–атак [15].

```
php_value auto_prepend_file /home/f0668516/domains/f0668516.xsph.ru/public_html/antiddos.php

Deny from 103.117.192.14
Deny from 178.124.189.174
Deny from 31.171.154.199
Deny from 70.34.253.43
Deny from 23.129.64.142
Deny from 194.233.69.38
Deny from 188.0.147.102
Deny from 23.129.64.139
Deny from 23.129.64.146
Deny from 138.0.89.154
Deny from 18.189.228.38
Deny from 212.64.72.199
Deny from 42.194.130.58
Deny from 66.196.238.178
Deny from 162.150.62.90
Deny from 162.150.62.94
Deny from 162.150.62.89
Deny from 176.57.188.32
Deny from 163.172.64.42
Deny from 199.249.230.87
Deny from 180.125.216.109
Deny from 23.129.64.148
Deny from 23.184.48.238
Deny from 83.137.158.8
Deny from 23.129.64.133
Deny from 45.153.160.134
Deny from 185.254.75.32
```

Рис. 20. Содержание файла .htaccses
Fig. 20. Contents of the .htaccses file

Вывод. DDoS-атаки сегодня стали распространенными и опасным явлением. Они угрожают безопасности практически любого онлайн-бизнеса и могут существенно повлиять на доходы его владельцев. Бороться с этой угрозой самостоятельно крайне сложно — это требует серьезной технической подготовки, к тому же отнимает много времени и средств. Самым разумным шагом для любого Web–сайта и Web–приложения является размещение на специально подготовленной для этого хостинг-площадке, имеющей эффективную систему защиты от DDoS–атак. Выбирая хостинг–провайдер, следует обратить пристальное внимание на то, как именно он защищает своих клиентов от атак типа «отказ в обслуживании», а также понять, являются лиэти меры достаточными для конкретного случая пользователя [16].

В данном исследовании авторами проведён анализ существующих искусственных нейронных сетей для определения их оптимальной структуры;изучены существующие методы детектирования сетевых DDoS– атак; собраны и доработаны датасеты для улучшения качества обучения; создана структура искусственной нейронной сети классификатора и проведено её обучение, создано демонстрационное программное средство, иллюстрирующее процесс классификации и блокировки и нейтрализации DDoS–атак.

Библиографический список:

1. Машинное обучение // [Электр. ресурс] URL: https://trends.rbc.ru/trends//60c85c599a7947f5776ad409 (дата обращения 26.04.2022).
2. Способы обучения нейронной сети [Электр. ресурс] URL: https://neurohive.io/ru/osnovy-data-science (дата обращения 27.04.2022).
3. RandomForest [Электр. ресурс] URL: https://scikit-learn.org/stable/modules (дата обращения 03.05.2022).
4. Машинное обучение [Электр. ресурс] URL: https://trends.rbc.ru/trends/industry/ (дата обращения 02.05.2022).
5. DDOS-атаки [Электронный ресурс] URL: https://aws.amazon.com/ru/shield/ddos-attack (дата обр. 28.04.2022).
6. Различные типы DDoS-атак [Электр. ресурс] URL: https://stormwall.pro/blog-edu (дата обр.: 29.04.2022).
7. Виды атак [Электр. ресурс] URL: https://itelon.ru/blog/zashchita-servera-ot-ddos-atak / (дата обр.: 30.04.2022).
8. ДокументацияPython[Электронный ресурс]. URL: https://www.python.org/about/ (дата обращения 10.05.2022).
9. Документация Sklearn [Электр. ресурс] URL: https://scikit- learn.org/stable/index.html (дата обр. 15.05.2022).
10. Документация NumPy [Электронный ресурс] URL: https://numpy.org/ (Дата обращения 16.05.2022).
11. Документация Pandas [Электронный ресурс] URL: https://pandas.pydata.org/ (дата обращения 14.05.2022).
12. Документация Mathplotlib [Электронный ресурс] URL: https://matplotlib.org/ (Дата обращения 20.05.2022).
13. Документация Seaborn [Электронный ресурс] URL: https://seaborn.pydata.org/ (Дата обращения 22.05.2022).
14. Модификация классического квантового протокола bb84, повышающая его характеристики Ляшенко К.А., Поркшеян В.М., Черкесова Л.В., Ревякина Е.А., Енгибарян И.А., Бурякова О.С.,

Решетникова О.А. Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2023. № 2. С. 100-115.

15. Cherckesova L. V. et al. Development of Quantum Protocol Modification CSLOE–2022, Increasing the Cryptographic Strength of Classical Quantum Protocol BB84 //Electronics. – 2022. – Т. 11. – №. 23. – С. 3954.
16. Korochentsev, D.A., Cherckesova, L.V., Revyakina, E.A., Goncharov, R.A. Investigation of the application of software generator of the speech-like interference to protect acoustic information from leakage through an acoustic channel Journal of Physics: Conference Seriesthis link is disabled, 2021, 2131(2), DOI 10.1088/1742-6596/2131/2/022091

References:

1. Machine learning // [Electr. resource] URL: https://trends.rbc.ru/trends//60c85c599a7947f5776ad409 (date accessed 04/26/2022).
2. Methods for training a neural network [Electr. resource] URL: https://neurohive.io/ru/osnovy-data-science (access date 04/27/2022).
3. RandomForest [Electr. resource] URL: https://scikit-learn.org/stable/modules (accessed 05/03/2022).
4. Machine learning [Electr. resource] URL: https://trends.rbc.ru/trends/industry/ (access date 05/02/2022).
5. DDOS attacks [Electronic resource] URL: https://aws.amazon.com/ru/shield/ddos-attack (accessed 04/28/2022).
6. Different types of DDoS attacks [Electr. resource] URL: https://stormwall.pro/blog-edu (access date: 04/29/2022).
7. Types of attacks [Electr. resource] URL: https://itelon.ru/blog/zashchita-servera-ot-ddos-atak / (access date: 04/30/2022).
8. Python documentation [Electronic resource]. URL: https://www.python.org/about/ (accessed 05/10/2022).
9. Sklearn Documentation [Electr. resource] URL: https://scikit-learn.org/stable/index.html (accessed 05/15/2022).
10. NumPy Documentation [Electronic resource] URL: https://numpy.org/ (Date of access: 05/16/2022).
11. Pandas documentation [Electronic resource] URL: https://pandas.pydata.org/ (accessed May 14, 2022).
12. Mathplotlib documentation [Electronic resource] URL: https://matplotlib.org/ (Date of access: 05/20/2022).
13. Seaborn documentation [Electronic resource] URL: https://seaborn.pydata.org/ (Accessed 05/22/2022).
14. Modification of the classical quantum protocol bb84, increasing its characteristics Lyashenko K.A., Porksheyan V.M., Cherkesova L.V., Revyakina E.A., Engibaryan I.A., Buryakova O.S., Reshetnikova O.A. *Modern science: current problems of theory and practice. Series: Natural and technical sciences*. 2023; 2:. 100-115. (In Russ)
15. Cherckesova L. V. et al. Development of Quantum Protocol Modification CSLOE–2022, Increasing the Cryptographic Strength of Classical Quantum Protocol BB84. *Electronics.* 2022; 11(23):3954. (In Russ)
16. Korochentsev, D.A., Cherckesova, L.V., Revyakina, E.A., Goncharov, R.A. Investigation of the application of software generator of the speech-like interference to protect acoustic information from leakage through an acoustic channel Journal of Physics: Conference Seriesthis link is disabled, 2021, 2131(2), DOI 10.1088/1742-6596/2131 /2/022091

Сведения об авторах:

Разумов Павел Владимирович, аспирант 3 курса, кафедра «Кибербезопасность информационных систем»; razumov1996@inbox.ru ORCID: 0000-0003-2454-3600

Черкесова Лариса Владимировна, доктор физико-математических наук, кафедра профессор, «Кибербезопасность информационных систем»; chia2002@inbox.ru ORCID: 0000-0002-9392-3140

Ревякина Елена Александровна, кандидат технических наук, доцент, кафедра «Кибербезопасность информационных систем»; revyelena@yandex.ru, ORCID: 0000-0003-1577-2671

Information about the authors:

Pavel V. Razumov, 3rd year Graduate Student, Department of Cybersecurity of Information Systems; razumov1996@inbox.ru, ORCID: 0000-0003-2454-3600

Larisa V. Cherkesova, Dr. Sci., (Physical and Mathem.), Department of Professor, Department of Cybersecurity of Information Systems; chia2002@inbox.ru, ORCID: 0000-0002-9392-3140

Elena A. Revyakina, Cand. Sci. (Eng.), Assoc. Prof., Department of Cybersecurity of Information Systems; revyelena@yandex.ru, ORCID: 0000-0003-1577-2671

Конфликт интересов/Conflict of interest.

Авторы заявляют об отсутствии конфликта интересов/The authors declare no conflict of interest.

Поступила в редакцию/ Received 10.02.2024.

Одобрена после рецензирования/ Reviced 02.03.2024.

Принята в печать/ Accepted for publication 02.03.2024.