**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)**

Факультет Инфокоммуникационных сетей и систем

Кафедра Защищенных систем связи

*Допустить к защите*

Зав. Кафедрой Красов А. В.

(Ф.И.О., подпись)

« » июня 2024 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Разработка и реализация программного комплекса для практических работ по дисциплине "основы стеганографии"

*(тема ВКР)*

Вид выпускной квалификационной работы Бакалаврская работа

*(бакалаврская работа, дипломная работа, дипломный проект, магистерская диссертация)*

Направление/специальность подготовки

11.03.02 Инфокоммуникационные технологии и системы связи

*(код и наименование направления/специальности)*

Квалификация Бакалавр

*(наименование квалификации в соответствии с ФГОС ВО)*

Студент:

Ерж Егор Витальевич, ИКТЗ-05

*(Ф.И.О., № группы) (подпись)*

Руководитель ВКР:

доцент каф. ЗСС, Ахрамеева К.А.

*(учёная степень, учёное звание, Ф.И.О.) (подпись)*

*Оборотная сторона титульного листа*

*работа написана мною самостоятельно*

*работа не содержит неправомерных заимствований*

*работа может быть размещена в электронно-библиотечной системе университета*

|  |  |  |
| --- | --- | --- |
| « » июня 2024 |  | Ерж Е.В. |
| *(дата)* | *(подпись)* | *(ФИО студента)* |

Текст ВКР размещен в электронно-библиотечной системе университета.

Руководитель отдела комплектования библиотеки

« » июня 2024

*(Ф.И.О.)*

*(дата) (подпись)*

Коэффициент оригинальности ВКР % .

Проверила: Ахрамеева Ксения Андреевна, к.т.н., доцент, доцент каф. ЗСС

*(Должность, Ф.И.О.)*

« » июня 2024

*(дата) (подпись)*

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА» (СПбГУТ)**

Факультет Инфокоммуникационных сетей и систем Кафедра Защищённых систем связи Направление (специальность) 11.03.02 Инфокоммуникационные технологии и системы связи

(код и наименование)

**Утверждаю:**

Зав. Кафедрой Красов А. В.

(Ф.И.О., подпись)

« » июня 2024 г.

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы (ВКР)**

1. Студент Ерж Егор Витальевич № группы ИКТЗ-05

(фамилия, имя, отчество)

1. Руководитель Ахрамеева Ксения Андреевна, к.т.н, доцент, доцент каф. ЗСС

(фамилия, имя, отчество, должность, уч. степень и звание)

1. Квалификация Бакалавр

(наименование в соответствии с ФГОС ВО)

1. Вид ВКР Бакалаврская работа

(бакалаврская работа, дипломная работа, дипломный проект, магистерская диссертация)

1. Тема ВКР Разработка и реализация программного комплекса для практических работ по дисциплине "основы стеганографии"
2. Исходные данные (технические требования): Материалы преддипломной практики, открытые источники информации в сети Интернет, источники научно-технической информации.
3. Содержание работы (анализ состояния проблемы, проведение

исследований, разработка, расчеты параметров, экономическое обоснование и др.)

Введение

Глава 1 Стеганография

Глава 2 Глубокое обучение в анализе сетевого трафика

Глава 3 Реализация нейросетевой модели для предотвращения DDoS-атак

Заключение

1. Вид отчетных материалов, представляемых в ГЭК (пояснительная записка, перечень, графического материала, отчет о НИР, технический проект, образцы и др.): пояснительная записка, графический материал в виде pptx- презентации
2. Консультанты по ВКР с указанием относящихся к ним разделов

|  |  |  |  |
| --- | --- | --- | --- |
| Раздел | Консультант | Подпись дата | |
| Задание выдал | Задание принял |
| 1. Общие положения по использованию глубокого обучения в кибербезопасности | Ахрамеева К.А. |  |  |
| 2. Глубокое обучение в анализе сетевого трафика | Ахрамеева К.А. |  |  |
| 3. Реализация нейросетевой модели для предотвращения DDoS-атак | Ахрамеева К.А. |  |  |

Дата выдачи задания « » \_\_\_\_\_\_ 2024 г.

Дата представления ВКР к защите « » \_\_\_\_ 2024 г.

*Руководитель ВКР* (Ахрамеева К.А.)

(подпись)

*Студент* (Ахрамеева К.А.)

(подпись)

КАЛЕНДАРНЫЙ ПЛАН

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование этапов выпускной квалификационной работы (ВКР) | Срок выполнения этапов ВКР | Примечание |
| 1. | Постановка цели выполнения ВКР и задач |  |  |
| 2. | Работа с теоретическим материалом |  |  |
| 3. | Сбор информации, необходимой |  |  |
|  | для написания работы |  |
| 4. | Систематизация и обработка материалов ВКР |  |
| 5. | Анализ полученных в работе |  |  |
|  | результатов, обобщение |  |
| 6. | Подготовка отчетных материалов, |  |  |
|  | представляемых в государственную |  |
|  | экзаменационную комиссию, |  |
|  | доклада к защите и презентации |  |
| 7. | Консультации с руководителем |  |  |
|  | ВКР |  |
| 8. | Представление выполненной ВКР |  |  |
|  | руководителю для подготовки |  |
|  | отзыва |  |
| 9. | Подготовка к защите ВКР, включая |  |  |
|  | подготовку к процедуре защиты и |  |
|  | процедуру защиты |  |

*Студент*  (Ерж Е.В.)

(подпись)

*Руководитель ВКР* (Ахрамеева К.А.)

(подпись)

**РЕФЕРАТ**

Тема выпускной квалификационной работы “ Разработка и реализация программного комплекса для практических работ по дисциплине "основы стеганографии"”

Пояснительная записка содержит:

1. 92 страниц
2. 38 рисунка
3. 9 таблиц
4. 10 использованных источников

Ключевые слова: Глубокое обучение, нейронные сети, сетевой трафик, DDoS-атаки, кибербезопасность, машинное обучение.

Целью данной работы является теоретическое исследование аномалий в сетевом трафике, а также разработка и анализ эффективности модели глубокого обучения для детектирования и предотвращения DDoS-атак.

Для достижения поставленной цели в данной работе будет проведён анализ существующих методов проведения DDoS-атак, их классификация, а также анализ основных угроз и уязвимостей, связанных с данными атаками. Будут рассмотрены теоретические аспекты глубокого обучения и искусственных нейронных сетей, их применение в анализе сетевого трафика, а также их способности к обнаружению аномалий и угроз в режиме реального времени. Также будет выбрана и реализована архитектура модели глубокого обучения, подходящей для задач детектирования DDoS-атак. Будут использованы различные метрики для оценки качества обучения и проведены эксперименты по сравнению разработанной модели с традиционными методами классификации сетевого трафика.

Результатом данной работы будет представлена модель глубокого обучения способная анализировать сетевой трафик на наличие в нем DDoS-атак.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 9](#_Toc166799004)

[ОСНОВНАЯ ЧАСТЬ 11](#_Toc166799005)

[Глава 1. Общие положения по использованию глубокого обучения в кибербезопасности 11](#_Toc166799006)

[1.1 Актуальность темы исследования 11](#_Toc166799007)

[1.2 Основные понятия и определения 12](#_Toc166799008)

[1.3 Принципы глубокого обучения и нейросетей. 15](#_Toc166799009)

[1.4 Обзор существующих методик обнаружения DDoS-атак. 20](#_Toc166799010)

[Глава 2. Глубокое обучение в анализе сетевого трафика 22](#_Toc166799011)

[2.1 Теоретические основы анализа сетевого трафика 22](#_Toc166799012)

[2.1.1 Методы анализа сетевого трафика 23](#_Toc166799013)

[2.1.2 Стек TCP/IP и модель OSI 26](#_Toc166799014)

[2.1.3 Инструменты и технологии для анализа сетевого трафика 39](#_Toc166799015)

[2.2 Проведение тестирования на проникновения ручным методом 43](#_Toc166799016)

[2.2.1 Традиционные методы классификации сетевого трафика 43](#_Toc166799017)

[2.2.2 Классификация DDoS-атак 48](#_Toc166799018)

[2.3 Принципы построения моделей глубокого обучения для анализа сетевого трафика 54](#_Toc166799019)

[2.3.1 Обзор и сравнение алгоритмов глубокого обучения 57](#_Toc166799020)

[2.4 Примеры существующих методов и систем глубокого обучения в обнаружении DDoS-атак 63](#_Toc166799021)

[Глава 3. Реализация нейросетевой модели для предотвращения DDoS-атак 67](#_Toc166799022)

[3.1 Выбор архитектуры и инструментов для модели 67](#_Toc166799023)

[3.2 Подготовка и анализ данных для обучения 67](#_Toc166799024)

[3.3 Обучение и валидация модели 75](#_Toc166799025)

[3.4 Тестирование и оценка эффективности 78](#_Toc166799026)

[3.5Выводы по разработке и оценке модели 88](#_Toc166799027)

[ЗАКЛЮЧЕНИЕ 90](#_Toc166799028)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 91](#_Toc166799029)

**ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ**

DDoS (Distributed Denial of Service) – распределённая атака типа "отказ в обслуживании", направленная на исчерпание ресурсов целевого сервера путём отправки большого количества запросов с множества заражённых устройств.

NTA (Network Traffic Analysis) – анализ сетевого трафика, процесс мониторинга и интерпретации сетевого трафика для выявления аномалий, угроз и других важных событий.

CNN (Convolutional Neural Network) – сверточная нейронная сеть, тип искусственной нейронной сети, используемой для обработки данных с топологической структурой, таких как изображения.

RNN (Recurrent Neural Network) – рекуррентная нейронная сеть, тип искусственной нейронной сети, в которой связи между узлами образуют направленный граф вдоль временной последовательности, что позволяет обрабатывать последовательные данные.

LSTM (Long Short-Term Memory) – долговременная кратковременная память, тип рекуррентной нейронной сети, способной запоминать и использовать информацию из длинных временных последовательностей.

LLM (Large Language Model) – большая языковая модель, тип модели машинного обучения, обученной на больших объемах текстовых данных для обработки и генерации человеческого языка.

ИИ (Искусственный Интеллект) – искусственный интеллект, область компьютерных наук, занимающаяся созданием систем, способных выполнять задачи, требующие человеческого интеллекта.

TCP (Transmission Control Protocol) – протокол управления передачей, один из основных протоколов интернета, обеспечивающий надежную, упорядоченную и проверенную доставку данных между приложениями.

UDP (User Datagram Protocol) – протокол пользовательских датаграмм, транспортный протокол интернета, обеспечивающий обмен датаграммами без установления соединения и гарантии доставки.

SGD (Stochastic Gradient Descent) – стохастический градиентный спуск, метод оптимизации, использующий случайные подвыборки данных для нахождения минимума функции потерь.

QDA (Quadratic Discriminant Analysis) – квадратичный дискриминантный анализ, статистический метод для классификации данных, основанный на квадратичной дискриминантной функции.

NLP (Natural Language Processing) – обработка естественного языка, область искусственного интеллекта, занимающаяся взаимодействием между компьютерами и человеческими языками.

XGBoost (Extreme Gradient Boosting) – экстремальное градиентное бустинг, мощный метод машинного обучения, использующий ансамбль слабых моделей для улучшения точности предсказаний.

# **ВВЕДЕНИЕ**

В эпоху стремительного развития технологий, учебные заведения сталкиваются с необходимостью внедрения современных технических решений для повышения доступности и эффективности образовательных ресурсов. В рамках моей выпускной квалификационной работы я занимаюсь разработкой программного комплекса для выполнения практических работ по дисциплине "Основы стеганографии". Цель проекта — создание дизайна, который будет соответствовать требованиям к удобству использования и обеспечит легкий доступ к необходимым сервисам для студентов и сотрудников, а также переработка практических работ с реализацией их на созданном ресурсе.

Работа включает в себя анализ текущих систем обучения, определение их сильных и слабых сторон, а также формулирование требований к новому интерфейсу; разработка новых практических работ. Основываясь на этом анализе, разработан учебный ресурс, который демонстрирует как применение теоретических знаний, так и учет практических потребностей пользователей.

Проект направлен на улучшение качества работы системы образования, что способствует более эффективной учебной и исследовательской деятельности в университете и повышает уровень удовлетворенности пользователей. Это исследование не только вносит вклад в академическую среду, но и обладает практической значимостью для университета, способствуя его дальнейшему развитию в области информационной безопасности.

**ОСНОВНАЯ ЧАСТЬ**

# **Глава 1. Общие положения по стеганографии**

## **1.1 Актуальность темы исследования**

## В современном мире, где объемы передаваемой и хранимой информации растут с невероятной скоростью, защита данных становится одной из ключевых задач. Актуальность стеганографии, как метода защиты информации, обусловлена:

1. Рост числа киберугроз: с каждым годом увеличивается количество кибератак, направленных на кражу, подделку и разрушение данных. Традиционные методы защиты, такие как криптография, часто оказываются недостаточными. Стеганография, скрывающая сам факт передачи данных, предоставляет дополнительный уровень безопасности, затрудняя работу злоумышленников.
2. Конфиденциальность и анонимность: в условиях всеобщего мониторинга и отслеживания активности в интернете, стеганография становится важным инструментом для обеспечения конфиденциальности и анонимности пользователей. Она позволяет скрывать личные данные и коммуникации, что особенно актуально в условиях жесткого государственного контроля и цензуры.
3. Защита интеллектуальной собственности: цифровая водяная маркировка, являющаяся одной из форм стеганографии, играет важную роль в защите авторских прав и интеллектуальной собственности. В условиях массового распространения цифрового контента, защита от нелегального копирования и распространения становится критически важной для правообладателей.
4. Развитие технологий: c развитием технологий увеличиваются и возможности стеганографии. Современные методы позволяют скрывать данные в различных медиаформатах с высокой степенью надежности и незаметности. Постоянное совершенствование алгоритмов и методов стеганографии способствует ее актуальности и востребованности.
5. Образование и научные исследования: в образовательной и научной сферах стеганография является перспективной областью для исследований и разработок. Она предоставляет широкие возможности для изучения и совершенствования методов защиты информации, что важно для подготовки специалистов в области информационной безопасности.

## **1.2 Основные понятия и определения**

Для разработки и реализации методов предотвращения кибератак, особенно важным становится глубокое понимание ключевых концепций и терминологии. Понимание этих терминов необходимо для того, чтобы полностью оценить потенциал и проблемные аспекты, связанные с применением глубокого обучения в борьбе с киберугрозами. Вот основные из них:

1. DDoS-атака (Distributed Denial of Service): это тип кибератаки, при которой множество компьютеров (часто зараженных вирусами и объединенных в сеть ботнет) отправляют большое количество запросов к одному серверу или сети с целью превысить их возможности по обработке данных и вызвать сбои в работе.
2. Глубокое обучение (Deep Learning): Отрасль машинного обучения, использующая модели искусственных нейронных сетей с множеством слоев (глубоких нейронных сетей), что позволяет системе обучаться на больших объемах данных и выявлять сложные закономерности и зависимости.
3. Искусственная нейронная сеть (Artificial Neural Network, ANN): Компьютерная модель, вдохновленная структурой и функционированием биологических нейронных сетей. Это основной инструмент глубокого обучения, используемый для моделирования сложных отношений между входами и выходами и нахождения закономерностей в данных.
4. Обучение с учителем (Supervised Learning): Метод машинного обучения, при котором модель обучается на основе предварительно размеченных данных, где каждому примеру входных данных соответствует ответ.
5. Классификация: Задача машинного обучения, в которой модель вычисляется для определения категории новых наблюдений на основе предыдущего опыта (обучающих данных).
6. Обнаружение аномалий (Anomaly Detection): Процесс определения необычных паттернов или отклонений в данных, которые не соответствуют ожидаемому поведению и могут указывать на важные события, такие как возможные атаки на информационную безопасность.
7. Метрики оценки модели: включают точность (accuracy), точность (precision), полноту (recall) и F1-меру. Эти показатели помогают оценить, насколько эффективно модель предсказывает данные и находит реальные DDoS-атаки.
8. Ботнет: Сеть частных компьютеров, зараженных вредоносным ПО, которые могут быть контролированы удаленно для проведения DDoS-атак или других мошеннических действий без ведома их владельцев.

Данные понятия и определения формируют фундамент для понимания технической стороны выпускной квалификационной работы, подчеркивая важность и сложность использования глубокого обучения для анализа и предотвращения DDoS-атак. Также необходимо понимать принцип построения DDoS-атак, который изображен на рисунке 1. DDoS-атаки, или распределённые атаки типа "отказ в обслуживании", организуются следующим образом:

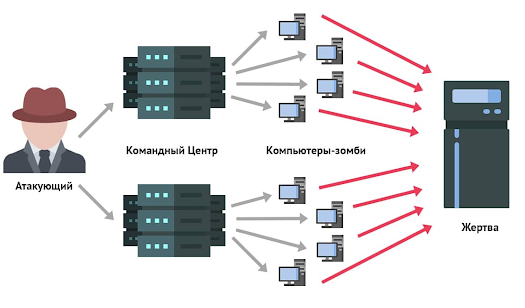


Рисунок 1 - Архитектура DDoS – Атак

Атакующий, обычно индивидуум или группа, которая планирует и инициирует DDoS-атаку. Обычно они используют компьютеры или серверы, на которых размещены инструменты для управления ботнетами. Командный центр часто представлен серверами или веб-платформами, которые атакующий использует для управления ботнетом. Отсюда они могут координировать действия множества компьютеров, которые будут выполнять атаку. Командный центр позволяет координировать множество компьютеров-зомби - сеть заражённых устройств, которые контролируются атакующим. Каждое устройство (бот) может быть любым подключённым к Интернету устройством, которое было заражено вредоносным ПО, позволяющим атакующему удалённо взять контроль. Запросы от ботов: Каждый компьютер-зомби начинает посылать непрерывный поток запросов к целевому серверу. Эти запросы могут быть различных типов (например, HTTP-запросы или TCP-подключения) и предназначены для истощения ресурсов сервера. Конечная цель атаки – это может быть сервер веб-сайта, онлайн-сервиса или инфраструктуры организации. Из-за массированной нагрузки со стороны ботнета сервер перегружается, что может привести к сбоям в работе, замедлению или полному отказу в обслуживании легитимных запросов пользователей. DDoS-атаки могут иметь различные вариации, включая, но не ограничиваясь, Volumetric Attacks, Protocol Attacks и Application Layer Attacks. Все они направлены на создание такого объёма запросов, который невозможно обработать целевой системе, что приводит к её деградации и отказу.

DDoS-атаки могут использоваться злоумышленниками не только для непосредственного вреда целевым системам, но и для достижения других целей. Они могут служить отвлекающим маневром для осуществления более серьёзных атак, таких как кража данных, в то время как основное внимание IT-специалистов приковано к борьбе с DDoS. Также злоумышленники могут получать финансовую выгоду, манипулируя финансовыми рынками или делая ставки на недоступность сервисов. DDoS-атаки относительно легко организовать через доступные онлайн-сервисы, что делает их доступными даже для неопытных пользователей. При этом анонимность и низкая стоимость таких атак делают их привлекательными для многих. Однако стоит помнить о серьёзных правовых последствиях для тех, кто их проводит, а также об этических вопросах, связанных с ущербом для невинных третьих лиц, чьи устройства могут быть заражены и использованы в составе ботнетов.

Таким образом, в контексте информационной безопасности, DDoS-атаки представляют собой значительную угрозу для организаций всех размеров, обладая потенциалом серьёзно нарушить их операции и подорвать доверие клиентов. Учитывая анонимность, доступность и разрушительную силу DDoS, требуются эффективные средства для их обнаружения и противодействия. Сложность и динамичность современных кибератак требуют использования передовых методов анализа, способных адаптироваться и эволюционировать в ответ на меняющиеся стратегии злоумышленников.

## **1.3 Принципы глубокого обучения и нейросетей.**

Принципы глубокого обучения и нейронных сетей являются фундаментальными для современного искусственного интеллекта и машинного обучения. Вот основные из них:

Искусственные нейронные сети (ИНС): ИНС вдохновлены структурой и функционированием мозга человека и состоят из слоёв нейронов, которые обрабатывают информацию, передавая сигналы от входного слоя к выходному. Они обозначены в виде кружков, которые отражены на рисунке 2. Нейроны связаны между собой.

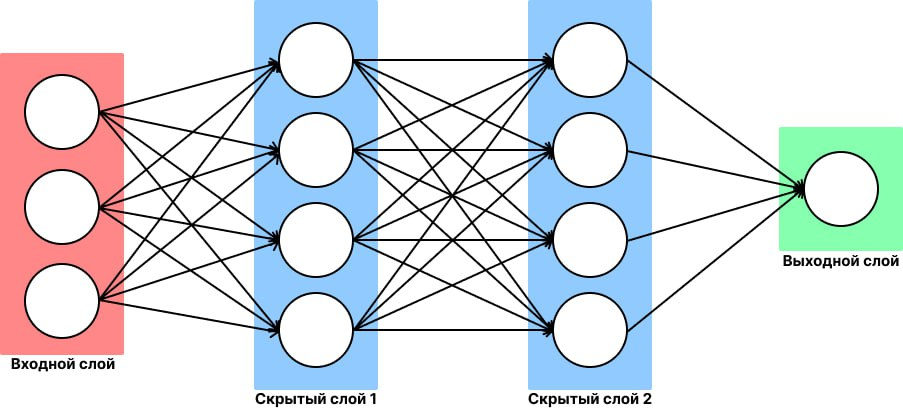


Рисунок 2 - Устройство нейросетей

Существует три вида слоев нейронов: входной слой, cкрытые слои, выходной слой. Для описания работы устройства нейросетей применим, для наглядности, авиа-терминологию, что отражено на рисунке 3. Входной слой получает входные данные. В нем 4 нейрона: начальный пункт полета, конечный пункт полета, дата вылета, авиакомпания. От этого слоя входные данные транслируются на первый скрытый слой.

В скрытых слоях выполняют математические операции с данными. Одна из трудностей при создании нейросетей – определить количество скрытых слоев и сколько нейронов в каждом слое. Глубокое обучение потому и называется глубоким, что количество скрытых слоев больше одного.

Выходной слой выдает, что получилось в итоге, то есть примерную оценку стоимости рейса.

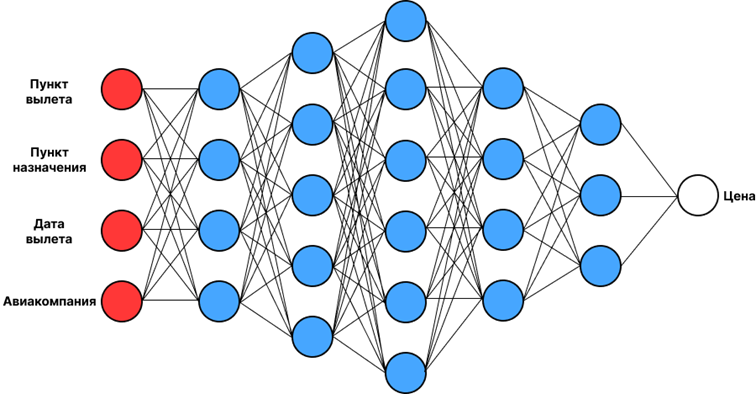
****

Рисунок 3 - Взаимодействие нейронов в слоях нейросети

Каждая связь между нейронами имеет вес, который определяет важность входного значения, что отражено на рисунке 4. Изначально веса произвольные. Для оценки стоимости авиарейса одним из важнейших параметров является дата вылета. Следовательно, связи с нейроном даты вылета будут обладать большим весом.

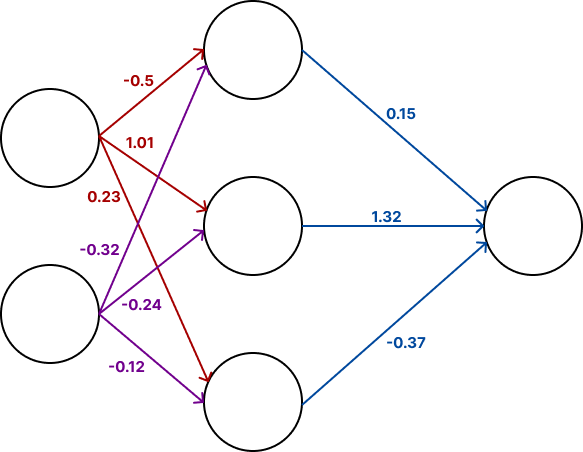


Рисунок 4 - Связи нейронов в нейросети

У каждого нейрона есть функция активации. Если простыми словами, одна из ее задач – «стандартизация» выходных данных. После того, как входные данные прошли через все слои нейросети, выходной слой выдает выходные данные.

Обучение ИИ – сложнейший этап глубокого обучения. Во-первых, набор данных должен быть огромным. Во-вторых, ресурсы для вычислений должны быть очень большими. В случае с сервисом по оценке стоимости авиарейсов, необходимы данные о стоимости билетов за прошлые периоды. Комбинаций аэропортов и дат вылета очень много. Поэтому придется работать с очень большим количеством вариантов цен на билеты. Для обучения машины потребуется задать входы из набора данных и сравнить полученные выходы с выходами из набора данных. Поначалу выходы будут неверными. После обработки, можно создать функцию стоимости, которая показывает, насколько неверными были выходы ИИ в сравнении с реальными выходами. В идеале, она должна иметь нулевое значение. Это произойдет, когда выходы ИИ совпадут с выходами из набора данных.

Глубина обучения, или глубокое обучение (Deep Learning), — это подмножество машинного обучения, которое включает в себя использование глубоких нейронных сетей. Термин "глубокий" относится к количеству слоёв в нейронной сети. В традиционных нейронных сетях обычно есть только один скрытый слой между входным и выходным слоями, в то время как глубокие нейронные сети имеют два или более скрытых слоя.

Основные элементы глубокого обучения:

1. Многослойность: глубокие нейронные сети содержат несколько скрытых слоёв между входным и выходным слоями. Каждый слой состоит из нейронов, которые обучаются на основе взвешенных входных данных и функции активации.
2. Иерархия признаков: нейроны в более глубоких слоях сети учатся на основе более высокоуровневых абстракций, представленных выходными данными предыдущих слоёв. Это означает, что нейросеть может распознавать сложные структуры в данных, постепенно переходя от простых паттернов к более сложным.
3. Обратное распространение (Backpropagation): это процесс, с помощью которого нейросеть обновляет свои веса в ответ на ошибку между предсказанными и истинными значениями. Ошибка распространяется обратно по сети, так что каждый нейрон может скорректировать свои веса для уменьшения этой ошибки.
4. Функции активации: функции активации, такие как ReLU или сигмоид, вводят нелинейность в процесс обучения, что позволяет сети учиться и представлять более сложные шаблоны.
5. Оптимизация: для настройки весов нейронной сети используются алгоритмы оптимизации, например, стохастический градиентный спуск (SGD) и его вариации. Эти методы минимизируют функцию потерь, которая измеряет разницу между предсказаниями сети и фактическими значениями.

Преимущества глубокого обучения:

* Способность обучения на необработанных данных: Глубокие нейронные сети могут обучаться на сырых данных, автоматически извлекая необходимые признаки, что сокращает необходимость вручную разрабатывать алгоритмы их выделения.
* Генерализация: после обучения на большом объеме данных глубокие нейронные сети могут обобщать знания и распознавать шаблоны в новых, невиданных ранее данных.
* Гибкость и адаптивность: Глубокие нейронные сети могут быть адаптированы к различным типам данных и задач, от распознавания изображений и речи до прогнозирования временных рядов и анализа текста.

Глубокое обучение сыграло ключевую роль в многих прорывах в области искусственного интеллекта за последнее десятилетие и продолжает быть важной областью исследований и разработок, включая предотвращение и анализ киберугроз, таких как DDoS-атаки.

## **1.4 Обзор существующих методик обнаружения DDoS-атак.**

Изобретение методик обнаружения DDoS-атак было вызвано необходимостью реагировать на эволюцию киберугроз и защищать онлайн-сервисы и инфраструктуру от вредоносных действий. Рост Интернета и цифровой экономики привёл к тому, что вопросы информационной безопасности стали особенно важными. Существует несколько методик обнаружения DDoS-атак, каждая из которых применяется для идентификации различных видов и характеристик атак. Вот основные из них:

Анализ сигнатур — это один из самых распространенных методов обнаружения DDoS-атак. Этот метод зависит от базы данных, содержащей определения известных атак, называемых сигнатурами. Как антивирусный сканер, он сравнивает входящий трафик с этими сигнатурами для обнаружения совпадений. Если трафик соответствует известной сигнатуре атаки, система может быстро идентифицировать и блокировать его. Этот метод эффективен для обнаружения уже известных атак, но его главный недостаток в том, что он не способен обнаруживать новые или модифицированные типы атак, которые ещё не были включены в базу данных.

Анализ аномалий, напротив, основан на выявлении отклонений от нормального поведения трафика. Системы, использующие этот подход, создают базовый профиль нормального трафика, чтобы любое значительное отклонение от этой нормы могло быть распознано как потенциальная атака. Это делает анализ аномалий мощным инструментом для обнаружения неизвестных атак. Однако трудность заключается в точном определении, что считать нормальным, и в возможности ложных срабатываний, когда легитимный, но необычный трафик ошибочно воспринимается как атака.

Объемно-основанный анализ сосредоточен на мониторинге объема трафика в сети. Если система обнаруживает необычно большой объем трафика, который не соответствует обычным моделям, это может указывать на DDoS-атаку. Такие системы эффективны против внезапных и массивных всплесков трафика, но более сложные атаки, такие как распределённые атаки на прикладном уровне, могут оставаться незамеченными.

Методы, основанные на анализе поведения, включают в себя мониторинг характерных действий пользователей или сетевых устройств. Это может включать такие вещи, как частота соединений или запросов, временные интервалы между действиями или даже стилевые особенности взаимодействия с приложениями. Например, если аккаунт отправляет слишком много запросов в короткий промежуток времени, это может сигнализировать о части ботнета. Такой подход требует тонкой настройки и непрерывного обучения системы для адаптации к изменяющимся условиям.

Использование машинного обучения является относительно новым, но перспективным подходом в обнаружении DDoS. Модели машинного обучения обучаются распознавать сложные паттерны в данных о трафике, которые могут указывать на атаку. Эти модели могут обучаться на исторических данных о DDoS-атаках и адаптироваться к новым угрозам по мере их появления. Главное преимущество здесь в способности машинного обучения постоянно совершенствоваться, но для эффективности таких систем требуются большие объемы разнообразных обучающих данных.

Наконец, гибридные системы сочетают в себе различные методы обнаружения, например, сигнатурный анализ с анализом аномалий или машинное обучение с объемно-основанным анализом, чтобы усилить их эффективность и уменьшить недостатки каждого из подходов. Эти системы стараются использовать сильные стороны каждого метода для создания более надежной защиты от DDoS-атак.

# **Глава 2. Глубокое обучение в анализе сетевого трафика**

# **2.1 Теоретические основы анализа сетевого трафика**

Анализ сетевого трафика (NTA - Network Traffic Analysis) можно описать как процесс изучения сетевого трафика для характеристики общих портов и протоколов, используемых в сети, мониторинга и реагирования на угрозы, а также обеспечения максимально полного представления о сети организации. Данный процесс (NTA) помогает специалистам по информационной безопасности выявлять аномалии, в том числе угрозы безопасности в сети, а также своевременно и эффективно реагировать на них. Анализ сетевого трафика также может облегчить процесс соблюдения правил информационной безопасности. Злоумышленники все чаще находят уязвимости, которые большинство компаний допускают в своих сетях, что усложняет обнаружение и, как следствие, реагирование на угрозы. В таких случаях анализ сетевого трафика снова может оказаться полезным. В таблице (таб.1) показаны и перечислены повседневные случаи использования анализа сетевого трафика (NTA):

Таблица 1 - Случаи использования анализа сетевого трафика (NTA)

|  |  |  |
| --- | --- | --- |
| **Использование NTA** | **Описание** | **Цели и Преимущества** |
| Сбор трафика в реальном времени (Collecting) | Мониторинг всех сетевых взаимодействий для наблюдения за потоками данных и быстрой реакции на аномалии. | Быстрое выявление потенциальных угроз, таких как DDoS-атаки или несанкционированный доступ. |
| Установление базового уровня (Setting) | Анализ типичных паттернов трафика, частоты передачи данных, используемых портов и протоколов для определения "нормы". | Облегчение точного определения отклонений, которые могут указывать на кибератаки или технические сбои. |
| Анализ подозрительных портов и хостов (Identifying) | Анализ активности через необычные порты и мониторинг | Идентификация попыток эксплуатации уязвимостей и несанкционированной деятельности. |
| Обнаружение вредоносного ПО (Detecting) | Выявление сигнатур вредоносного ПО и аномальных паттернов поведения, указывающих на наличие вирусов, троянов, программ-вымогателей. | Защита сети и данных от разрушительного программного обеспечения и улучшение сетевой безопасности. |

Также стоит отметить, что анализ сетевого трафика помогает не только обнаружить текущие угрозы, но и спрогнозировать потенциальные уязвимости. Например, если обнаружить множество SYN-пакетов на портах, которые никогда (или редко) использовались в сети, то можно сделать вывод что злоумышленник пытается определить, какие порты открыты на хостах. Подобные действия являются типичными маркерами «portscan». Проведение данного сетевого анализа трафика и приход к таким выводам требует определённого теоретического минимума.

### **2.1.1 Методы анализа сетевого трафика**

Мониторинг сети включает в себя постоянное наблюдение и анализ сетевой инфраструктуры, устройств и моделей трафика для обеспечения оптимальной производительности, доступности и безопасности. Отслеживая ключевые показатели, такие как задержка, потеря пакетов и состояние устройства, сетевые администраторы получают ценную информацию о поведении сети, что позволяет им оперативно выявлять и устранять проблемы. Методы анализа сетевого трафика могут быть различными и зависят от целей мониторинга, типов сетей и используемых инструментов. Рассмотрим их более подробно:

Активный анализ сетевого трафика – данный метод предполагает активную генерацию и отправку трафика в сеть для тестирования и измерения ее производительности. Данный метод обычно основан на синтетических транзакциях, таких как проверка связи с устройством или выполнение скриптовых тестов, для оценки производительности сети и обнаружения потенциальных проблем. Он предоставляет в режиме реального времени информацию о доступности сети, времени отклика и пропускной способности.

Пассивный анализ сетевого трафика - подразумевает мониторинг и запись сетевого трафика без влияния на его ход. Вместо этого он захватывает и анализирует существующий сетевой трафик, часто с помощью сетевых перехватов или зеркалирования портов. Пассивный мониторинг обеспечивает комплексное представление о фактическом поведении сети, позволяя проводить детальный анализ моделей трафика, угроз безопасности и производительности приложений с течением времени. Инструменты для пассивного анализа включают снифферы, такие как Wireshark, которые могут захватывать и анализировать копии пакетов, проходящих через сеть.

Анализ временных рядов в контексте сетевого трафика включает изучение трафика, который записывается и анализируется в течение определённого периода времени. Этот метод позволяет выявить закономерности, тренды и возможные аномалии в данных, которые проходят через сеть. Основная цель анализа временных рядов — обеспечение возможности проследить динамику изменения сетевого трафика, что критически важно для ряда задач. Например, с его помощью можно определить, когда именно происходят пики загрузки сети, что может указывать на необходимость увеличения сетевых ресурсов или наличие нестандартной или вредоносной активности, такой как DDoS-атаки. Анализ также помогает определить периоды сниженной активности, что может быть полезно для планирования технического обслуживания или апгрейдов систем. Для выполнения анализа временных рядов данные собираются в регулярные интервалы времени. Эти данные могут включать различные параметры, такие как объем переданных данных, скорость передачи, количество активных соединений и другие метрики. После сбора данных применяются статистические методы для выявления общих тенденций и отклонений от нормы.

Статический анализ сетевого трафика представляет собой метод исследования, при котором данные анализируются без учёта изменений во времени. Это означает, что анализируются отдельные моменты или выборки данных без привязки к их динамике. В отличие от анализа временных рядов, статический анализ не фокусируется на изменении данных со временем и не требует длительного наблюдения за трафиком для выявления тенденций или закономерностей. Также характерной чертой данного метода анализа сетевого трафика является использование математических моделей и алгоритмов для исследования собранных данных о сети. Статический анализ часто используется для проверки соответствия трафика определенным стандартам или правилам безопасности, для выявления вредоносного кода или других аномалий, которые могут быть встроены в сами данные. При статическом анализе большое внимание уделяется детальному изучению содержимого трафика, что позволяет глубоко понять его природу и потенциальные угрозы.

Диаграммы и визуализации – данные методы анализа сетевого трафика, наглядно представляют данные, которые помогают быстрее и эффективнее понимать сложные паттерны и аномалии в сетевой активности. Инструменты визуализации, такие как Splunk или Grafana, могут использоваться для создания комплексных дашбордов, отображающих ключевые метрики производительности и безопасности. Визуализация данных трафика позволяет отслеживать в реальном времени такие параметры, как объем трафика, скорость передачи, источники и маршруты трафика, а также выявлять необычные или подозрительные активности, которые могут указывать на нарушения безопасности. Она также полезна для оптимизации сетевых ресурсов, позволяя анализировать использование пропускной способности и эффективность сетевой инфраструктуры.

Каждый из рассмотренных методов играет важную роль в комплексном подходе к управлению сетевыми ресурсами и обеспечению кибербезопасности. Пассивный и активный анализы обеспечивают фундаментальное понимание текущего состояния сети и её поведения под нагрузкой, предоставляя важные данные для мониторинга и оптимизации. Анализ временных рядов раскрывает динамические изменения в трафике, помогая выявлять тенденции и предсказывать потенциальные проблемы. Статистический анализ применяется для глубокого понимания характеристик трафика и обнаружения аномалий, что критически важно для предотвращения атак и сбоев. Наконец, диаграммы и визуализации предоставляют интуитивно понятные и легко усваиваемые способы представления данных, что существенно упрощает анализ и принятие решений.

Комбинирование этих методов позволяет организациям не только реагировать на текущие события в сети, но и стратегически планировать улучшения безопасности и эффективности своих сетевых инфраструктур. Это подчеркивает необходимость интегрированного подхода к анализу сетевого трафика, где каждый метод дополняет другие, обеспечивая всеобъемлющий контроль и управление сетевыми операциями

### **2.1.2 Стек TCP/IP и модель OSI**

Стек TCP/IP и модель OSI являются двумя фундаментальными концепциями в области сетевых технологий, которые помогают в понимании организации и функционировании компьютерных сетей. Хотя обе модели разработаны для стандартизации коммуникаций в сети, они имеют разные подходы и структуру. Специалисты по сетевым технологиям используют OSI для моделирования или концептуализации того, как данные отправляются или принимаются по сети. Понимание этого является основополагающей частью большинства сертификаций ИТ-сетей, включая программы сертификации Cisco Certified Network Associate (CCNA) и CompTIA Network+. OSI модель предназначена для разделения стандартов, процессов и протоколов передачи данных на семь уровней, каждый из которых отвечает за выполнение конкретных задач, касающихся отправки и получения данных.

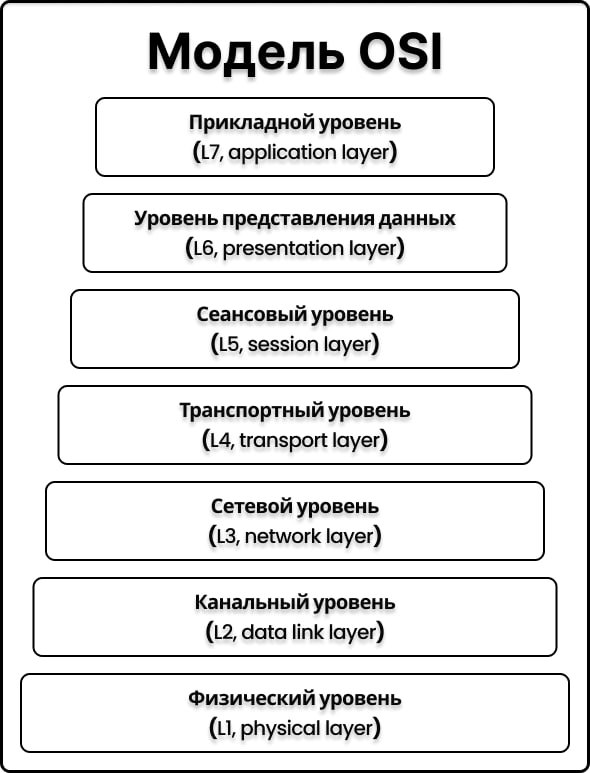


Рисунок 5 - Семь уровней модели OSI

Основная концепция OSI заключается в том, что процесс связи между двумя конечными точками в сети можно разделить на семь отдельных групп связанных функций или уровней, что отражено на рисунке 5. В этой архитектуре каждый уровень обслуживает уровень выше него и, в свою очередь, обслуживается слоем ниже него. Таким образом, в данном сообщении между пользователями будет поток данных вниз через уровни исходного компьютера, через сеть, а затем вверх через уровни принимающего компьютера. Только уровень приложения наверху стека не предоставляет услуги уровню более высокого уровня.

Уровень 1. Физический уровень (L1, physical layer) - передает данные с помощью электрических, механических или процедурных интерфейсов. Этот уровень отвечает за отправку компьютерных битов с одного устройства на другое по сети. Он определяет, как устанавливаются физические соединения с сетью и как биты преобразуются в предсказуемые сигналы. На физическом уровне наиболее широко известным протоколом является Ethernet. Этот протокол определяет методы кодирования и передачи сигналов по кабелям. Также существуют такие технологии, как Bluetooth, Wi-Fi и ИК-порт, каждая из которых предусматривает собственные правила для передачи данных.

Уровень 2. Канальный уровень (L2, data link layer) - Уровень канала передачи данных или уровень протокола в программе обрабатывает перемещение данных в физический канал сети и из него. Этот уровень решает проблемы, возникающие в результате ошибок передачи битов. Это гарантирует, что скорость потока данных не перегружает отправляющие и принимающие устройства. Данные, получаемые с физического уровня, разделяются на фреймы, или кадры. Как устроены фреймы можно подробно изучить на рисунке 6. Каждый из этих фреймов включает в себя служебные данные, такие как адрес отправителя и адрес получателя, а также сами данные.

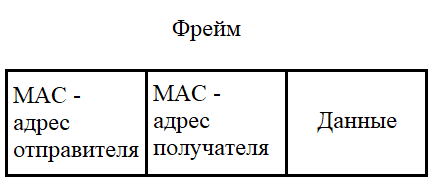


Рисунок 6 - Структура фрейма

Уровень канала передачи данных можно дополнительно разделить на два подуровня. Более высокий уровень, который называется управлением логическим каналом (LLC), отвечает за мультиплексирование, управление потоком, подтверждение и уведомление верхних уровней в случае возникновения ошибок передачи/приема (TX/RX). Подуровень управления доступом к среде передачи данных отвечает за отслеживание кадров данных с использованием MAC-адресов отправляющего и принимающего оборудования. Он также отвечает за организацию каждого кадра, маркировку начальных и конечных битов и организацию времени относительно того, когда каждый кадр может быть отправлен по среде физического уровня.

Уровень 3. Сетевой уровень (L3, network layer) - Основная функция данного уровня — перемещение данных в другие сети и через них. Протоколы сетевого уровня достигают этого путем упаковки данных с правильной информацией о сетевом адресе, выбора соответствующих сетевых маршрутов и пересылки упакованных данных вверх по стеку на транспортный уровень. С точки зрения TCP/IP, именно здесь IP-адреса применяются для целей маршрутизации.



Рисунок 7 - Маршрутизатор сети

Маршрутизатор, который изображен на рисунке7, обеспечивают взаимодействие различных сетей, используя MAC-адреса для создания маршрута между устройствами. На сетевом уровне информация представлена в форме пакетов. Их внутренняя структура отражена на рисунке 8. Эти пакеты аналогичны фреймам канального уровня, однако они используют другие типы адресов — IP-адреса отправителя и получателя.

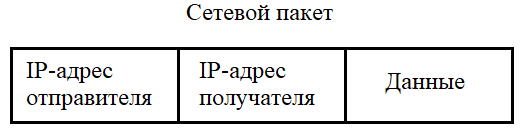


Рисунок 8 - Структура сетевого пакета

Для определения IP-адресов устройств (отправителя и получателя) применяется протокол ARP (protocol address resolution), который позволяет преобразовывать MAC-адреса в IP-адреса и обратно.

Уровень 4. Транспортный уровень (L4, transport layer) - отвечает за передачу данных по сети и обеспечивает механизмы проверки ошибок и управления потоками данных. Он определяет, какой объем данных следует отправить, куда они будут отправлены и с какой скоростью. На данном уровне существует два основных протокола – TCP и UDP. Именно они отвечают за то, как именно будут передаваться данные. TCP (Transmission Control Protocol) — это протокол, обеспечивающий точную и надежную доставку данных. Он строго контролирует передачу каждого бита информации, однако его скорость работы ниже, чем у UDP. В протоколе TCP информация разбивается на сегменты, каждый из которых представляет собой часть данных пакета. Сегментация позволяет адаптироваться к пропускной способности сети и повышает надежность передачи. Большие пакеты данных могут потеряться или быть ошибочно отправлены не тому адресату, тогда как меньшие сегменты минимизируют эти риски и позволяют контролировать количество передаваемых данных. Если доставка какого-либо сегмента не удаётся, TCP может запросить его повторную отправку, тем самым гарантируя достоверность передачи. В UDP данные организуются в датаграммы, что аналогично пакетам, однако каждая датаграмма является самостоятельной и содержит всю необходимую информацию для достижения конечного получателя. Благодаря этому датаграммы не зависят друг от друга, могут передаваться различными маршрутами и поступать в любом порядке.

Уровень 5. Сетевой уровень (L5, session layer) - устанавливает, координирует и завершает диалог между приложениями. Его основные функции – это аутентификация и повторное подключение после прерывания. Этот уровень определяет, как долго система будет ждать ответа другого приложения. На этом уровне данные представлены в привычном виде, например, MP3 или JPEG-файлов. Простым языком задача этого уровня представить данные в понятном для человека виде и сделать так, чтобы пользователь мог ею воспользоваться.

Уровень 6. Уровень представления данных (L6, presentation layer) - преобразует или форматирует данные для уровня приложения на основе семантики или синтаксиса, принимаемого приложением. Этот уровень также выполняет шифрование и дешифрование, наглядно как устроен этот процесс можно изучить на рисунке 9, необходимые для уровня приложения. К примеру, полученные данные могут быть преобразованы в файлы формата GIF или MP4.

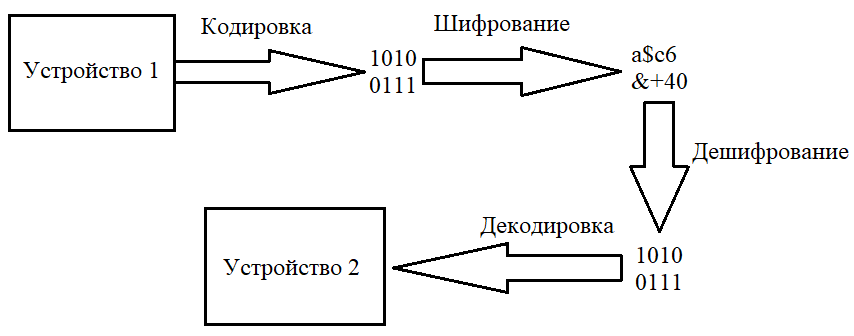


Рисунок 9 - Процесс отправки данных между устройствами

Уровень 7. Прикладной уровень (L7, application layer) - позволяет пользователю (человеку или программному обеспечению) взаимодействовать с приложением или сетью всякий раз, когда пользователь решает читать сообщения, передавать файлы или выполнять другие задачи, связанные с сетью. Веб-браузеры и другие приложения, подключенные к интернету, используют протоколы приложений уровня 7. Прикладной уровень можно сравнить с видом графического интерфейса для всей модели OSI — он позволяет пользователю взаимодействовать с другими уровнями модели, часто даже не осознавая этого. Этот интерфейс также известен как сетевой. Наиболее известные примеры сетевых интерфейсов включают HTTP, HTTPS, FTP и SMTP.

Межуровневые функции или услуги, которые могут влиять на более чем один уровень, включают следующее:

* функции управления, которые позволяют конфигурировать, создавать экземпляры, контролировать и прекращать связь двух или более объектов;
* Многопротокольная коммутация по меткам (MPLS), которая работает на уровне модели OSI, который находится между уровнем канала передачи данных уровня 2 и сетевым уровнем уровня 3. MPLS может передавать различный трафик, включая кадры Ethernet и IP-пакеты;
* Протокол разрешения адресов (ARP) преобразует адреса IPv4 (уровень OSI 3) в MAC-адреса Ethernet (уровень OSI 2)
* система доменных имен (DNS), которая представляет собой службу прикладного уровня, которая используется для поиска IP-адреса доменного имени.

В модели OSI можно выделить основные плюсы и минусы. Данная информация отображена в таблице (таб.2), представленной ниже.

Таблица 2 - Плюсы и минусы модели OSI

|  |  |
| --- | --- |
| **Преимущества модели OSI** | **Недостатки модели OSI** |
| Стандартизация сетевых протоколов, обеспечивающая взаимодействие и совместимость различных устройств. | Сложность и избыточность множества уровней для понимания и применения в некоторых сценариях. |
| Модульность, позволяющая специалистам сосредоточиться на отдельных аспектах сетевого взаимодействия. | Непрактичность некоторых аспектов модели в реальных сетевых технологиях. |
| Облегчение процесса отладки за счет четкой структуризации сетевых процессов. | Избыточность уровней, где некоторые могут быть объединены или оказаться менее значимыми. |
| Гибкость в разработке продуктов, ориентированных на конкретные уровни модели. | Трудности в полной реализации всех уровней модели, что может быть ресурсоемким и дорогим. |
| Улучшенное понимание и систематизация, способствующая обучению и освоению сетевых протоколов. | Отставание от быстрого развития технологий, что делает некоторые аспекты устаревшими. |

Модель OSI является одной из теоретических основ для понимания и разработки сетевых решений и проведения сетевого анализа, но её практическое применение может быть ограничено в зависимости от конкретных требований и условий.

Протокол TCP/IP, или Протокол управления передачей/Интернет-протокол, является фундаментом для интернета и значительной части современных сетевых взаимодействий. Эта технология представляет собой набор коммуникационных протоколов, используемых для соединения сетевых устройств в Интернете. TCP/IP позволяет различным сетям успешно обмениваться данными, независимо от их внутренней архитектуры или платформы.

TCP (Transmission Control Protocol) задействован в передаче данных, контролируя отправку и гарантируя их доставку получателю в неизменном виде.

IP (Internet Protocol) отвечает за адресацию и связывание устройств в сети, а также за разделение данных на пакеты для их эффективной отправки. Для быстрого нахождения маршрута между компьютерами были разработаны IP-адреса — уникальные идентификаторы, присваиваемые каждому устройству в сети.

Функциональность модели TCP IP разделена на четыре уровня, что наглядно отражено на рисунке 10, каждый из которых включает определенные протоколы. TCP/IP — это система многоуровневой серверной архитектуры, в которой каждый уровень определяется в соответствии с конкретной выполняемой функцией. Все эти четыре уровня TCP IP работают совместно, передавая данные с одного уровня на другой.

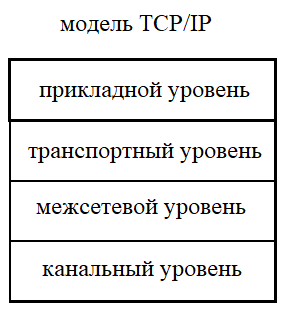


Рисунок 10 - Структура модели TCP/IP

Прикладной уровень — Это самый верхний уровень, который указывает приложения и программы, которые используют модель TCP/IP для связи с пользователем через приложения и различные задачи, выполняемые этим уровнем, включая представление данных для приложений, выполняемых пользователем, и пересылает их на транспортный уровень.

Уровень приложений поддерживает плавное соединение между приложением и пользователем для обмена данными и предлагает различные функции, такие как удаленное управление системой, услуги электронной почты и т. д.

Некоторые из протоколов, используемых на этом уровне:

* HTTP: протокол передачи гипертекста используется для доступа к информации, доступной в Интернете.
* SMTP: простой протокол передачи почты, предназначенный для обработки шагов и проблем, связанных с электронной почтой.
* FTP: это стандартный протокол, который контролирует передачу файлов по сетевому каналу.

Транспортный уровень - Этот уровень отвечает за установление соединения между отправителем и устройством-получателем, а также выполняет задачу разделения данных прикладного уровня на пакеты, которые затем используются для создания последовательностей. Он также выполняет задачу сохранения данных, т. е. их передачи без ошибок, и контролирует скорость потока данных по каналу связи для плавной передачи данных.

На этом уровне используются следующие протоколы:

* TCP: протокол управления передачей отвечает за правильную передачу сегментов по каналу связи. Он также устанавливает сетевое соединение между системой источника и назначения.
* UDP: протокол пользовательских дейтаграмм отвечает за выявление ошибок и другие задачи во время передачи информации. UDP поддерживает различные поля для передачи данных, такие как:
* Адрес исходного порта: этот порт отвечает за разработку приложения, составляющего передаваемое сообщение.
* Адрес порта назначения: этот порт принимает сообщение, отправленное со стороны отправителя.
* Total Length: общее количество байтов пользовательской дейтаграммы.
* Checksum: используется для обнаружения ошибок сообщения на стороне назначения.

Межсетевой уровень - выполняет задачу управления передачей данных в сетевых режимах и применяет протоколы, связанные с различными этапами, связанными с передачей данных по каналу, который находится в форме пакетов, отправленных предыдущим уровнем. Простыми словами данный уровень строит маршруты между устройствами в сети интернет что в свою очередь называется процессом маршрутизации. Для того чтобы установить местоположение получателя и проложить маршрут к нему, IP использует систему DNS, которая содержит информацию об IP-адресах всех устройств в интернете. После получения адреса передаваемый файл делится на маленькие части, известные как пакеты. Эти пакеты включают в себя части данных и служебную информацию, такую как IP-адреса отправителя и получателя. Затем начинается процесс передачи пакетов через маршрутизаторы и коммутаторы. Однако за процесс отправки отвечает уже транспортный уровень.

Некоторые из протоколов, применяемых на этом уровне:

* IP: этот протокол присваивает вашему устройству уникальный адрес; IP-адрес также отвечает за маршрутизацию данных по каналу связи.
* ARP: Этот протокол относится к протоколу разрешения адресов, который отвечает за поиск физического адреса с использованием IP-адреса.

Канальный уровень – данный уровень устанавливает физическое соединение между устройствами в локальной сети с помощью радиоволн и проводов. Здесь используются всеми известные протоколы: Ethernet, Wi-Fi, Bluetooth. На этом уровне информация разделяется на маленькие сегменты, называемые фреймами, которые затем передаются между устройствами. Каждый фрейм включает часть передаваемых данных и служебные сведения.

Для определения маршрута фреймов применяется адресация канального уровня, использующая MAC-адреса. Эти уникальные физические адреса устройств позволяют протоколам канального уровня идентифицировать отправителей и получателей. Важной функцией канального уровня является также обеспечение безошибочной передачи данных, для чего используются различные средства проверки.

Модель TCP/IP является стандартной сетевой моделью, которая легла в основу интернета и сыграла ключевую роль в развитии современных сетевых технологий. Она имеет свои плюсы и минусы, которые отражены в таблице (таб.3) ниже.

Таблица 3 - Плюсы и минусы модели TCP/IP

|  |  |
| --- | --- |
| **Преимущества модели TCP/IP** | **Недостатки модели TCP/IP** |
| Масштабируемость. Модель TCP/IP хорошо масштабируется и может работать как в небольших, так и в крупных сетях. | Сложность: модель довольно сложна и требует определенного уровня знаний для настройки и обслуживания. |
| Надежность: модель прочная и надежная, что делает ее подходящей для критически важных приложений. | Уязвимость: из-за своей сложности он уязвим для атак. |
| Гибкость: он очень гибок и обеспечивает совместимость между различными типами сетей. | Производительность. Производительность может снизиться из-за перегрузки сети и задержек. |
| Экономическая эффективность: TCP/IP относительно недорог в реализации и обслуживании. |  |

Модель TCP/IP была разработана в 1960-х годах для поддержки и объяснения передачи данных, тогда как модель OSI представляет собой сетевую концепцию, специально предназначенную для объяснения взаимодействия и работы данных и протоколов во время передачи информации. Ниже представлена таблица (таб.4), в которой наглядно показаны отличия модели OSI от модели TCP/IP.

Таблица 4 - Различия моделей OSI и TPC/IP

|  |  |  |
| --- | --- | --- |
| **Критерий** | Модель OSI | Модель TCP/IP |
| **Количество уровней** | 7 уровней: физический, канальный, сетевой, транспортный, сеансовый, представительный, прикладной. | 4 уровня: сетевой интерфейс, интернет, транспортный, прикладной. |
| **Стандартизация** | Разработана ISO для универсальной стандартизации сетевых взаимодействий. | Разработана как стандарт для интернета и конкретных сетевых протоколов. |
| **Принципы разработки** | Строго разделенные функциональные уровни. | Более гибкая, сосредоточена на достижении конечной функциональности. |
| **Интероперабельность** | Подчеркивает универсальную взаимозаменяемость и взаимодействие. | Сфокусирована на взаимодействии в рамках интернет-протоколов. |
| Применение | Теоретическая модель, используется для обучения и концептуализации. | Практическая модель, широко используемая в реальных сетевых технологиях. |
| Гибкость и адаптация | Менее гибкая из-за жесткой структуры. | Большая гибкость в адаптации к новым технологиям и требованиям. |
| Безопасность | Рассматривает безопасность на нескольких уровнях. | Интегрирует безопасность в основном на транспортном и прикладном уровнях. |

Стек TCP/IP и модель OSI представляют собой фундаментальные концепции в разработке и анализе сетевых технологий, имея при этом свои уникальные особенности и применения. Модель OSI, с её чётким разделением сетевых функций на семь уровней, обеспечивает комплексное и детальное понимание сетевого взаимодействия, что делает её незаменимым инструментом для обучения и стандартизации. Она позволяет лучше понять, как разные аспекты сетевых операций взаимодействуют между собой, что критически важно для диагностики проблем и разработки сетевых решений.

Стек TCP/IP, с другой стороны, является более прагматичным и ориентированным на реализацию подходом, что делает его основой для интернета и большинства коммерческих сетей. Он упрощает сложность многоуровневых интеракций, сведя их к четырем уровням, что облегчает понимание и управление сетевыми процессами в реальных условиях.

Обе модели имеют решающее значение для анализа сетевого трафика, так как предоставляют рамки для интерпретации сетевых данных и взаимодействий. Понимание этих моделей позволяет аналитикам точно определять, на каком уровне возникают проблемы, и какие инструменты или методы наиболее подходят для их решения. Таким образом, глубокое знание стека TCP/IP и модели OSI обеспечивает сильную основу для эффективного мониторинга, управления и защиты сетевых ресурсов.

### **2.1.3 Инструменты и технологии для анализа сетевого трафика**

Для анализа сетевого трафика существуют специальные инструменты, которые помогают специалистам ИБ с задачей отслеживания аномалий и своевременного реагирование на угрозы. Инструменты анализа сетевого трафика позволяют отслеживать входящие и исходящие сетевые пакеты с данными, чтобы получить ключевые сведения, такие как производительность сети, безопасность и пропускная способность и т.д. Однако не все решения одинаковы. Не существует универсального подхода, подходящего для всех сетей и конфигураций. В первую очередь, стоит узнать как можно больше информации о сети и устройствах, находящихся в ней, чтобы выбрать подходящий инструмент для анализа сетевого трафика. Наиболее популярные и эффективные инструменты, а также их описание приведены ниже:

Wireshark — это один из самых известных и широко используемых анализаторов пакетов. Он позволяет захватывать и взаимодействовать с реальным трафиком в сети, предоставляя детальную информацию о каждом пакете. Wireshark поддерживает сотни протоколов и позволяет глубоко анализировать сетевую активность. Данный инструмент можно использовать для изучения деталей трафика на различных уровнях: от информации уровня соединения до битов, составляющих один пакет. Также можно отметить, что возможности Wireshark выходят за рамки простого мониторинга и позволяют решать другие задачи сетевого администрирования: устранение неполадок сети, анализ безопасности, анализ производительности, анализ протокола.

Tshark — это консольная версия Wireshark, предоставляющая мощные возможности для захвата и анализа пакетов в режиме командной строки. Это делает его идеальным инструментом для использования в автоматизированных скриптах и на серверах без графического интерфейса.

Tcpdump — компьютерная программа для анализа пакетов сети передачи данных, работающая под интерфейсом командной строки. Он позволяет пользователю отображать TCP/IP и другие пакеты, передаваемые или получаемые по сети, к которой подключен компьютер. Tcpdump работает в большинстве Unix-подобных операционных систем: Linux, Solaris, FreeBSD, DragonFly BSD, NetBSD, OpenBSD, Openwork, macOS, HP-UX 11i и AIX. В этих системах tcpdump использует библиотеку libpcap для захвата пакетов. Порт tcpdump для Windows называется WinDump; он использует WinPcap, версию libpcap для Windows.

Tcpick — это инструмент для отслеживания, перехвата и логирования TCP потоков, который может сохранять данные сессий TCP. Этот инструмент полезен для восстановления данных и отладки сетевых приложений. Tcpick также Tcpdump является утилитой командной строки, что делает ее гибкой для использования в различных сценариях, включая удаленные и автоматизированные среды, где графический интерфейс пользователя недоступен.

NGrep, или network grep, — это инструмент сопоставления с образцом, созданный для выполнения тех же функций, что и grep для дистрибутивов Linux. Большая разница в том, что он работает с пакетами сетевого трафика. NGrep понимает, как читать текущий трафик или трафик из файла PCAP, а также использовать выражения регулярных выражений и синтаксис BPF. Этот инструмент лучше всего работает при использовании для отладки трафика по таким протоколам, как HTTP и FTP.

Network Taps - Тапы (Gigamon, Niagra-taps) — устройства, способные снимать копии сетевого трафика и отправлять их в другое место для анализа. Они могут активно захватывать и анализировать трафик напрямую или пассивно, возвращая исходный пакет обратно в сеть, как будто ничего не изменилось. Данные инструменты широко используются для безопасного мониторинга, диагностики и анализа сетевого трафика.

Networking Span Ports Span Ports — это способ копирования кадров с сетевых устройств второго или третьего уровня во время исходящей или входящей обработки и отправки их в точку сбора. Часто порт зеркалируется для отправки этих копий на сервер журналов. Это особенно полезно для мониторинга и отладки сетевых проблем без вмешательства в производительность сети.

Elastic Stack (ранее известный как ELK Stack) состоит из Elasticsearch, Logstash и Kibana. Этот набор инструментов широко используется для централизованного сбора, обработки и визуализации логов и данных сетевого трафика. Это позволяет анализировать большие объемы данных и искать тренды и аномалии в реальном времени.

Системы Security Information and Event Management (SIEM) предназначены для мониторинга и анализа событий безопасности в реальном времени. Они агрегируют и анализируют логи и данные из различных источников, помогая обнаруживать, оценивать и реагировать на потенциальные угрозы.

SolarWinds Network Performance Monitor (NPM) — это мощное решение для мониторинга сети, которое предоставляет детализированные сведения о производительности сети. Он автоматически обнаруживает сетевые устройства и визуализирует их производительность в реальном времени, что помогает выявлять проблемы до того, как они скажутся на пользователях.

PRTG Network Monitor от Paessler — это универсальный инструмент для мониторинга сети, который проверяет доступность устройств, меряет трафик и помогает пользователям оптимизировать свои сети за счёт комплексной визуализации всех сетевых компонент.

NetFlow Analyzer от ManageEngine — это инструмент, основанный на технологии NetFlow. Он предназначен для анализа пропускной способности и использования сети. NetFlow Analyzer может помочь идентифицировать кто, когда и как использует сетевые ресурсы.

Инструменты для анализа сетевого трафика, такие как tcpdump, tcpick, Wireshark, Tshark, а также технологии, включая Networking Span Ports, Network Taps, Elastic Stack, и SIEM системы, представляют собой критически важные средства для глубокого понимания и эффективного управления сетевой инфраструктурой. Эти инструменты и технологии обеспечивают широкие возможности для мониторинга, диагностики и безопасности сетей, позволяя специалистам захватывать, анализировать и фильтровать трафик в реальном времени, выявлять аномалии, оптимизировать производительность и реагировать на угрозы. Каждый из этих инструментов и методов обладает уникальными особенностями и преимуществами, предлагая различные уровни сложности и детализации для анализа, что делает их подходящими для разнообразных сценариев использования — от простых задач мониторинга до сложных операций по обеспечению безопасности и отладки сетей. Вместе они формируют мощный арсенал, который позволяет компаниям и организациям поддерживать высокий уровень производительности сети и надежности, гарантируя безопасность ценных данных и сетевых ресурсов.

## **Проведение тестирования на проникновения ручным методом**

Классификация сетевого трафика и DDoS-атак играет критически важную роль в современном управлении сетевой безопасностью. В условиях неуклонно растущего объема данных и всё более сложных сетевых взаимодействий, понимание и разграничение различных типов трафика становится необходимым для обеспечения стабильности, производительности и безопасности сетевых систем. Это включает в себя не только идентификацию и анализ легитимного трафика, но и быстрое выявление и нейтрализацию потенциально вредоносных действий, таких как DDoS-атаки, которые могут нанести значительный ущерб сетевой инфраструктуре. Через различение трафика по типам, источникам, назначению, а также по способам его передачи, специалисты могут оптимизировать управление сетью, обеспечивая необходимый уровень сервиса для критически важных операций и ограничивая ресурсы для менее значимых процессов. С другой стороны, детальное понимание методов и векторов DDoS-атак позволяет разрабатывать более эффективные стратегии защиты, которые могут предотвратить или минимизировать ущерб от таких атак, обеспечивая непрерывность бизнес-процессов и защиту данных.

### **2.2.1 Традиционные методы классификации сетевого трафика**

Традиционная классификация сетевого трафика представляет собой метод классификации и идентификации трафика путем анализа свойств пакетов сетевого трафика. В этой области было проведено множество исследований, как итог данных работ, были сформированы основные методы классификаций.

1. Метод классификации на основе протокола. Это один из самых ранних методов классификации сетевого трафика, который классифицирует трафик в зависимости от используемого сетевого протокола, что позволяет точно определять типы данных, передаваемых через сеть, и соответствующим образом реагировать на них. Принцип работы данного метода начинается с анализа пакетов данных, которые передаются в сети. Каждый пакет содержит заголовки, которые указывают, какой протокол используется для передачи данных. Наиболее часто используемые протоколы включают TCP (Transmission Control Protocol), UDP (User Datagram Protocol), ICMP (Internet Control Message Protocol), а также протоколы более высокого уровня, такие как HTTP, HTTPS, FTP, и SMTP.

Применение классификации сетевого трафика на основе протокола охватывает множество аспектов управления сетью и безопасности. Этот метод позволяет глубже понять, как трафик взаимодействует с сетевой инфраструктурой, и обеспечивает основу для множества операционных и стратегических решений. Классифицировать трафик по протоколам особенно важно для обеспечения безопасности сети. Зная, какие протоколы должны использоваться в нормальных условиях, специалисты могут быстро выявлять аномалии, которые могут указывать на вредоносные действия или технические нарушения. Например, необычно высокое количество трафика по определённому протоколу может сигнализировать о попытке атаки или наличии вредоносного ПО. Также классификация трафика помогает в оптимизации работы сети. Понимание того, какие протоколы потребляют больше всего ресурсов, позволяет администраторам сети настроить приоритеты и обеспечить достаточный уровень производительности для критически важных приложений, в то время как менее важные задачи могут быть ограничены в ресурсах. Это также способствует более эффективному распределению пропускной способности и может предотвратить перегрузки сети.

1. Метод классификации на основе портов. Данный метод основан на анализе номеров портов, используемых в сетевых соединениях. Эти порты являются частью транспортного слоя в стеке TCP/IP, и каждый порт обычно ассоциируется с определённым типом службы или протокола. Классификация по портам позволяет определить тип передаваемых данных, оценить их приоритет и определить соответствующие меры безопасности. Для наглядного представления информации о методе классификации сетевого трафика на основе портов, разделим ключевые плюсы и минусы этого метода и отразим их в таблице 5:

Таблица 5 - Ключевые преимущества и недостатки метода классификации сетевого трафика на основе портов

|  |  |
| --- | --- |
| **Преимущества** | Недостатки |
| Контроль доступа и фильтрация. Позволяет настроить файрволы и другие механизмы фильтрации для разрешения или блокирования трафика на основе номеров портов. | Динамическое портовое присвоение. Современные приложения могут использовать динамические порты, что усложняет идентификацию и фильтрацию постоянного порта. |
| Приоритизация трафика. Классификация по портам облегчает настройку качества обслуживания (QoS) для критически важных приложений. | Шифрование трафика. Шифрованный трафик, такой как HTTPS, скрывает детали портов, делая классификацию менее эффективной. |
| Мониторинг и аналитика. Помогает в мониторинге и анализе сетевого трафика для выявления аномалий и нарушений безопасности. | Обход сетевых фильтров. Приложения могут маскировать свою деятельность под стандартные порты, чтобы обойти сетевые фильтры. |
| Соответствие нормам безопасности. Управление портами способствует соблюдению требований к безопасности и политик доступа. | Требование дополнительных методов. Часто требует комбинирования с другими методами классификации для увеличения точности и эффективности. |

Данная таблица подчеркивает, как метод классификации сетевого трафика на основе портов может быть полезен для различных аспектов управления сетью, но также подчеркивает ограничения, с которыми могут столкнуться организации при его использовании. Эффективное применение этого метода часто требует его интеграции с другими технологиями и подходами к анализу сетевого трафика.

1. Классификация сетевого трафика по содержанию является одним из методов анализа, позволяющим глубоко понимать и управлять передаваемыми через сеть данными. Этот метод основан на изучении конкретных данных внутри пакетов трафика, что позволяет не только определить тип трафика, но и выявить потенциальные угрозы безопасности, такие как вредоносные программы или неавторизованный доступ к конфиденциальной информации. Классификация по содержанию требует анализа данных на уровне приложений, включая текст, изображения, аудио и видео. Это может включать инспекцию глубоких пакетов (Deep Packet Inspection, DPI), которая позволяет просматривать и управлять данными, идентифицируя конкретные признаки или сигнатуры в трафике. DPI является мощным инструментом для обеспечения безопасности, соблюдения политик и оптимизации сети.
2. Традиционный метод машинного обучения в контексте классификации и идентификации сетевого трафика включает использование алгоритмов, таких как машины опорных векторов (SVM) и нейронные сети. Эти методы обеспечивают возможность анализировать большие объемы данных и автоматически выявлять сложные шаблоны и аномалии в трафике, что особенно полезно для обеспечения безопасности сетей и управления трафиком. На рисунке 11 отображены два основных подхода в традиционных методах машинного обучения (SVM и нейронные сети), их основные функции.

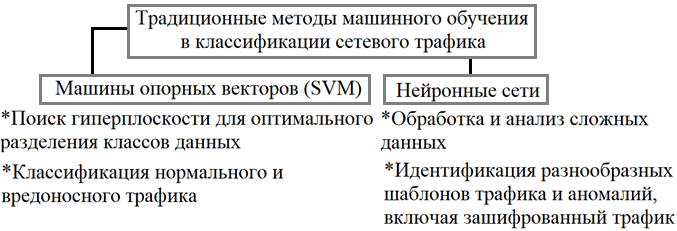


Рисунок 11 - Основные подходы в традиционных методах машинного обучения и их функции

Использование традиционных методов машинного обучения для классификации сетевого трафика позволяет повысить точность идентификации различных видов трафика и аномалий. Эти методы способны обрабатывать и анализировать данные в режиме реального времени, предоставляя сетевым администраторам мощные инструменты для мониторинга, предотвращения и реагирования на сетевые угрозы.

1. Метод классификации на основе глубокого обучения. Этот метод использует алгоритмы глубокого обучения, такие как CNN и рекуррентные нейронные сети (RNN), для классификации и идентификации трафика. Глубокое обучение позволяет машинам учиться на больших объемах данных и делать предсказания или принимать решения без вмешательства человека.

Применение данного метода:

* Обнаружение аномалий и угроз: Глубокое обучение может эффективно распознавать необычные паттерны трафика, которые могут указывать на DDoS-атаки, вторжения или другие сетевые угрозы. Модели могут обучаться на исторических данных и адаптироваться к новым угрозам по мере их появления.
* Классификация и приоритизация трафика: Глубокое обучение помогает классифицировать трафик по различным категориям, таким как видео, голос, веб-трафик и т.д., что позволяет оптимизировать управление сетевыми ресурсами и обеспечивать качество обслуживания (QoS).
* Оптимизация сетевых операций: Прогнозирование будущих паттернов трафика на основе текущих данных помогает в планировании пропускной способности и управлении ресурсами сети, улучшая общую эффективность сети.

Использование метода классификации сетевого трафика на основе глубокого обучения представляет ряд вызовов, которые необходимо учитывать при интеграции этого подхода в сетевую инфраструктуру. Одной из ключевых проблем является необходимость в больших объемах размеченных обучающих данных. В сетевом контексте, где трафик постоянно эволюционирует и часто содержит чувствительную информацию, сбор и поддержка актуальности таких данных может быть как трудоемким, так и потенциально небезопасным.

Кроме того, глубокое обучение требует значительных вычислительных ресурсов, особенно во время тренировки моделей. Это может стать серьезным барьером для организаций с ограниченными техническими ресурсами или теми, кто нуждается в быстрой обработке данных в реальном времени. Реализация эффективных и масштабируемых вычислительных систем для поддержки глубокого обучения может потребовать значительных инвестиций в инфраструктуру.

Дополнительной сложностью является интерпретация результатов. Модели глубокого обучения, как правило, действуют как "черные ящики", что затрудняет понимание того, как были получены конкретные выводы. Это может усложнить диагностику проблем и разработку стратегий ответа на инциденты, так как администраторам сетей может быть сложно точно определить, почему модель классифицировала трафик определенным образом.

### **2.2.2 Классификация DDoS-атак**

Классификация DDoS-атак (распределённых атак типа "отказ в обслуживании") имеет решающее значение для разработки эффективных стратегий защиты сетей. DDoS-атаки представляют собой угрозы, при которых злоумышленники стремятся нарушить доступность веб-ресурсов, серверов или инфраструктур, насыщая их избыточным трафиком из множества источников. Эти атаки могут быть классифицированы по различным критериям в зависимости от их механизмов, целей и поведения. Понимание этих классификаций помогает в разработке более целенаправленных и эффективных мер защиты. Рассмотрим ключевые методы классификации DDoS-атак:

1. Протокольные DDoS-атаки. Такие атаки в основном нацелены на следующие большие группы сетевых протоколов: UDP, TCP, другие (ICMP, GRE, IPIP, ESP, AH, SCTP, OSPF, SWIPE, TLSP, Compaq\_PEE). Злоумышленники могут потреблять сетевые ресурсы или серверные ресурсы, отправляя запросы, которые требуют значительного обработки времени сервером. Вот несколько примеров наиболее распространенных протокольных DDoS-атак:

Атака IP Null - Согласно техническим стандартам глобальных сетей, заголовок IP-пакета должен содержать информацию об используемом протоколе транспортного уровня в специальном поле – Протокол. Хакеры установили значение этого поля равным нулю. Этот метод позволяет отправлять большое количество пакетов, и никакие брандмауэры или маршрутизаторы не могут этому помешать. Системные ресурсы жертвы начинают постоянно анализировать входящий трафик, и в конечном итоге сервер выходит из строя.

SYN-флуд - Этот тип атаки основан на алгоритме трехэтапного установления связи TCP. Хакер быстро отправляет запросы на подключение к серверу, содержащие поддельный IP-адрес источника. SYN-флуд постепенно занимает всю память таблицы соединений.

UDP-флуд - Во время этой атаки сервер-жертва получает множество UDP-пакетов с разных IP-адресов. Поддельные пакеты UDP перегружают сетевой интерфейс, занимая всю полосу пропускания. Флуд — это огромное количество искаженных сообщений, используемых для создания мощного потока запросов, который перегружает всю выделенную полосу пропускания атакуемого ресурса.

Ping of Death - Эта атака дестабилизирует или приводит к сбою целевого компьютера. Хакер постоянно генерирует пакеты неправильного формата или слишком большого размера, используя простую команду ping. Если атакуемая система основана на стандартном протоколе IPv4, общий размер пакета не может превышать 65 535 байт, иначе система выйдет из строя. В настоящее время некоторые системы работают на обновленной версии протокола IPv6, и хакеры также нашли способы саботировать его. Они отправляют фрагменты искаженных пакетов. Система-жертва пытается их собрать заново, но в результате получается слишком большой размер пакета, что приводит к перегрузке памяти и сбоям в работе.

Сеансовая атака (SlowLoris) - Хакер устанавливает TCP-сеанс между сервером-жертвой и ботом. Когда сеанс успешно установлен, бот злоумышленника не отвечает пакетом ACK, чтобы поддерживать сеанс открытым до тех пор, пока не произойдет тайм-аут сеанса. Пустые сессии потребляют системные ресурсы, в результате чего сервер-жертва выделяет доступные ресурсы для поддержания открытых TCP-сессий с ботами, а затем становится недоступным.

1. Классификация DDoS-атак, основанных на OSI. Данные атаки могут быть направлены на разные уровни, от физического до прикладного, каждый из которых предполагает свои уникальные методы защиты.

Физический уровень: Канал передает необработанные двоичные данные между машинами. Он использует Bluetooth, USB, IrDA, а также концентраторы, розетки и патч-панели.

Пример атаки: Данный уровень может пострадать в результате физического разрушения или любого другого нарушения работы сети. Техногенные сбои приводят к полной непригодности оборудования. На этом уровне невозможны DoS- или DDoS-атаки.

Уровень канала передачи данных: Канальный уровень отвечает за обмен данными между узлами локальной сети. Данные группируются в кадры и передаются на физический уровень. Еще одна функция этого канального уровня — установка сетевым адаптерам уникальных идентификаторов — MAC-адресов.

Пример атаки: наиболее распространенной является лавинная рассылка MAC-адресов. Сетевые коммутаторы перегружены пакетами данных, что приводит к отключению всех портов подключения.

Сетевой уровень: на этом уровне начинают взаимодействовать маршрутизаторы и коммутаторы разных сетей. Маршрутизация основана на преобразовании MAC-адресов в сетевые адреса. Основная цель этого уровня — создать лучший способ передачи данных между устройствами.

Пример атаки: ICMP-флуд, который перегружает целевую сеть сообщениями ICMP. Он направлен на уменьшение пропускной способности и ограничение количества запросов, которые могут быть обработаны по протоколу ICMP.

Транспортный уровень: Он использует протоколы UDP и TCP, а также обрабатывает и транспортирует пакеты данных между узлами связи. Данный уровень контролирует поток информации и обнаруживает ошибки. Если они обнаружены, он повторно отправляет данные.

Пример атаки: превышение пороговых значений ширины канала и количества доступных соединений. Наиболее распространенными типами DDoS-атак являются Smurf и SYN-флуд.

Сеансовый уровень: Данный уровень отвечает за взаимодействие между приложениями, а также за установление и завершение соединений и синхронизацию задач ОС.

Пример атаки: злоумышленник использует уязвимости программного обеспечения через Telnet, в результате чего администратор теряет доступ к серверу.

Уровень представления: Уровень кодирует и декодирует данные и адаптирует их для людей или машин понятным образом. Сюда входят видео, аудио, изображения и текстовые данные. Между уровнями 6 и 7 существует протокол SSL. Он обеспечивает клиенту безопасное соединение с сервером и взаимную аутентификацию.

Пример атаки: мусорный флуд SSL. Хакеры генерируют неправильные SSL-запросы для атаки на сервер жертвы. Это замедляет работу ресурсов, поскольку проверка зашифрованных пакетов SSL занимает много времени.

Прикладной уровень: Прикладной уровень полностью работает для пользователя и представляет ему данные в понятном виде.

1. Классификация DDoS-атак по механизму действия. В данной классификации можно выделить 3 группы DDoS-атак с разными механизмами действия: использующие механизм флуда, эксплуатация уязвимости в стеке сетевых протоколов, атаки на уровне приложения. В таблице 6 представлены данные группы и их описание.

Таблица 6 - Классификации DDoS-атак по механизму действия, разделённая на три основные группы

|  |  |  |
| --- | --- | --- |
| Группа | Тип атаки | Описание |
| Атаки, использующие механизм флуда | Атака с усилением DNS | Запросы к общедоступному DNS-серверу с последующей переадресацией ответов на сервер-жертву, перегружая его длинными ответами. |
| Атаки, использующие механизм флуда | DNS-флуд | Массовые DNS-запросы на сервер, вызывая перегрузку ответами на легитимные и поддельные запросы. |
| Атаки, использующие механизм флуда | Фрагментированный UDP-флуд | Использование больших фрагментированных пакетов для заполнения полосы пропускания, требуя от сервера ресурсы на восстановление данных. |
| Атаки, использующие механизм флуда | Неподдельный UDP-флуд | Подобно UDP-флуду, но с использованием реальных IP-адресов для генерации пакетов, забивающих системные ресурсы. |

Продолжение таблицы 6

|  |  |  |
| --- | --- | --- |
| Группа | Тип атаки | Описание |
| Атаки, использующие механизм флуда | ICMP Fragmentation Flood | Отправка фрагментированных ICMP-пакетов для засорения пропускной способности. |
| Атаки, использующие механизм флуда | Атака с усилением NTP | Отправка запросов monlist на NTP-сервер с подменённым IP-адресом, перегружая сервер ответами. |
| Атаки, использующие механизм флуда | Ping-флуд | Использование ICMP эхо-запросов для вызова сбоев сервера. |
| Эксплуатация уязвимости протоколов | TOS-флуд | Использование полей TOS и ECN в IP-пакетах для создания иллюзии перегруженности сети и ограничения пропускной способности. |
| Эксплуатация уязвимости протоколов | ACK-флуд | Отправка поддельных пакетов ACK, которые не соответствуют ни одному сеансу, вызывая сбои. |
| Эксплуатация уязвимости протоколов | RST и FIN-флуд | Поддельные пакеты RST или FIN вызывают сбои, так как сервер не может идентифицировать их как легитимные. |
| Атаки на уровне приложения | HTTP-флуд | Генерация множества запросов, направленных на истощение ресурсов сервера. |

Продолжение таблицы 6

|  |  |  |
| --- | --- | --- |
| Группа | Тип атаки | Описание |
| Атаки на уровне приложения | DoS-атака на приложение | Использование уязвимостей веб-сайта или приложения, например, через SQL-инъекции, для вызова сбоев. |
| Атаки на уровне приложения | HTTP-флуд с одним запросом | Отправка одного большого пакета с множеством запросов, перегружая сервер. |
| Атаки на уровне приложения | Фрагментированная HTTP-флуд | Медленная отправка фрагментированных HTTP-пакетов, что позволяет обходить механизмы безопасности. |

Классификация DDoS-атак по механизму действия обеспечивает глубокое понимание тактик, используемых злоумышленниками, что критически важно для разработки эффективных стратегий защиты. Включает атаки, нацеленные на сетевые службы, такие как DNS и NTP, которые используют усиление для умножения трафика. Атаки также могут эксплуатировать уязвимости в сетевых протоколах или приложениях, вызывая исчерпание ресурсов и отказ в обслуживании. Каждый тип атаки требует специфического подхода к митигации и защите, что делает анализ механизмов атаки ключевым элементом в сетевой безопасности.

## **Принципы построения моделей глубокого обучения для анализа сетевого трафика**

Применение глубокого обучения для анализа сетевого трафика представляет собой последовательность определённых действий, начиная с сбора и обработки данных до внедрения и мониторинга моделей. Эти методы позволяют выявлять сложные шаблоны и аномалии, которые трудно обнаружить с помощью традиционных подходов.

Основные принципы построения моделей глубокого обучения для анализа трафика:

1. Сбор данных для анализа сетевого трафика с использованием моделей глубокого обучения начинается с определения типов данных, которые будут собираться, например, пакеты данных из HTTP-запросов или потокового видео. Используются специализированные инструменты, такие как Wireshark или tcpdump, для захвата этих данных. После сбора, данные необходимо преобразовать в формат, подходящий для обработки, что включает извлечение признаков, необходимых для обучения модели. Это также включает шаги по очистке и нормализации данных для удаления шума и обеспечения консистентности данных, что критически важно для повышения точности предсказаний модели.
2. Этап предобработки данных в анализе сетевого трафика с помощью глубокого обучения включает тщательную очистку и стандартизацию данных для обеспечения их качества и пригодности для обучения моделей. На этом этапе важно удалить из данных любые аномалии или шумы, которые могут ввести модель в заблуждение. Процесс нормализации также играет ключевую роль, поскольку он помогает унифицировать различные масштабы и диапазоны значений данных, что критически важно для эффективного обучения нейронных сетей. Эти процедуры помогают повысить точность и улучшить производительность моделей глубокого обучения.
3. Выбор архитектуры модели. Данный принципе построения модели глубокого обучения для анализа сетевого трафика требует определения структуры и типа нейронной сети, которая наилучшим образом соответствует специфике задачи. Например, сверточные нейронные сети (CNN) эффективно работают с паттернами в пакетах данных благодаря своей способности выявлять пространственные иерархии признаков. В то время как рекуррентные нейронные сети (RNN) предпочтительнее для анализа временных последовательностей данных, таких как потоковый трафик, поскольку они могут обрабатывать информацию с учетом ее временной структуры. Эти выборы определяются необходимостью адаптации к особенностям данных и целям анализа.
4. Принцип обучения модели в анализе сетевого трафика с использованием глубокого обучения включает несколько ключевых процессов. Во-первых, необходимо настроить параметры обучения, включая скорость обучения и количество эпох, что позволяет модели адаптироваться и оптимизировать свои веса на основе предоставленных данных. Важным аспектом является использование алгоритмов оптимизации, таких как SGD (стохастический градиентный спуск) или Adam, которые помогают минимизировать функцию потерь, улучшая тем самым производительность модели. Для борьбы с переобучением, когда модель слишком точно адаптируется к обучающему набору данных и плохо работает на новых данных, применяются различные техники. К ним относятся регуляризация, такая как L1 или L2, и методы, такие как dropout, который случайным образом исключает нейроны во время тренировки, помогая тем самым уменьшить зависимость от конкретных признаков. Эти методы помогают улучшить способность модели к обобщению, что критически важно для эффективного прогнозирования на новых данных.
5. Этап тестирования и валидации в процессе построения моделей глубокого обучения для анализа сетевого трафика критически важен для оценки эффективности модели. На этом этапе модель проверяется на отдельном тестовом наборе данных, который не использовался во время обучения. Это позволяет оценить, насколько хорошо модель способна обобщать свои знания на новых данных, а не только на тех, на которых она была обучена. Тестирование помогает выявить проблемы с переобучением и подтвердить, что модель действительно улавливает общие закономерности в данных, а не просто запоминает обучающий набор. Этот процесс может включать различные метрики оценки, такие как точность, прецизионность, полнота и F1-мера, чтобы дать всестороннюю оценку производительности модели.
6. Процесс внедрения и мониторинга модели глубокого обучения для анализа сетевого трафика включает развертывание обученной модели в реальной сетевой среде. Данный принцип требует интеграции модели с существующей инфраструктурой для наблюдения и анализа трафика в реальном времени. Критически важным аспектом является постоянный мониторинг её производительности и точности, чтобы обеспечить адекватное реагирование на новые угрозы и изменения в паттернах трафика. Это также может потребовать регулярной адаптации и переобучения модели, чтобы поддерживать её актуальность и эффективность в динамично изменяющейся сетевой среде.

Данные принципы составляют основу для создания мощных систем глубокого обучения, способных анализировать сетевой трафик. Они помогают не только в обнаружении и предотвращении угроз, но и в оптимизации работы сети, обеспечивая более высокую производительность и безопасность.

### **2.3.1 Обзор и сравнение алгоритмов глубокого обучения**

Алгоритмы глубокого обучения играют ключевую роль в современной обработке данных, обеспечивая возможность распознавать сложные шаблоны и делать предсказания на основе больших объемов данных. Рассматриваемые методы включают сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN), в том числе LSTM, каждый из которых имеет свои уникальные преимущества и ограничения в различных сценариях анализа трафика.

Сверточные нейронные сети (CNN) представляют собой особый тип нейронных сетей, которые специализируются на обработке данных в виде сетки. Примерами этого типа данных являются временные ряды и изображения, которые можно рассматривать как одномерную и двумерную сетку пикселей соответственно. Сверточные сети широко использовались в различных реальных задачах, таких как обработка естественного языка (NLP), компьютерное зрение, распознавание речи и т. д. Термин «сверточный» в сверточных нейронных сетях поддерживает идею о том, что CNN используют математическую операцию, называемую сверткой. В своей наиболее распространенной форме оператор свертки представляет собой особый тип линейной операции, которая выполняет интеграл произведения двух функций/сигналов. Другими словами, CNN — это нейронные сети, которые используют операторы свертки вместо общего умножения матриц хотя бы на одном из своих сетевых уровней. CNN применяют три ключевых принципа, которые можно применять для повышения производительности системы машинного обучения за счет сокращения пространства параметров модели: совместное использование параметров или весов, разреженное взаимодействие и эквивариантные представления.

Большая размерность — очевидный недостаток архитектуры нейронных сетей, особенно когда входные данные слишком велики и сложны, например, изображения. Для решения этой проблемы в качестве альтернативы полной связности в архитектуре нейронных сетей был введен оператор свертки (или уровень свертки). Графическое описание глубокой архитектуры CNN представлено на рисунке 12. CNN принимает многоканальные изображения (например, автомобилей и кораблей) в качестве входных данных для целей обучения. CNN использует преимущества нескольких слоев свертки с нелинейными функциями активации для облегчения сложности входных данных (т. е. изображений) и получения выходных данных, т. е. вероятности принадлежности каждого изображения к классу (или категории). В CNN каждая входная зона подключена к выходному нейрону, то есть локальной связности. Каждый уровень использует различные фильтры для распознавания абстрактных понятий, например границ транспортного средства. CNN может изучать функции более высокого уровня, такие как различные детали автомобиля, на более глубоких уровнях. Фильтры в CNN заранее не определены; вместо этого он автоматически изучает значение каждого фильтра на этапе обучения. Более того, CNN использует уровень объединения как метод понижающей выборки. На выходном слое применяется классификатор для использования функций высокого уровня для задачи классификации.

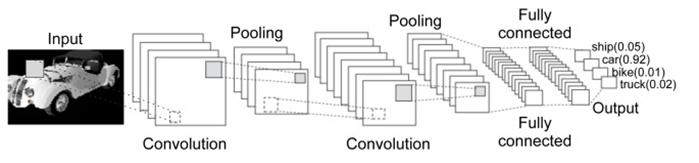


Рисунок 12 - Архитектура сверточных нейронных сетей CNN

Рекуррентные нейронные сети (RNN) представляют собой категорию искусственных нейронных сетей, подходящих для анализа последовательных данных. В отличие от CNN, которые предназначены для работы с данными топологии в виде сетки, например изображениями, RNN представляют собой нейронные сети, которые имеют специализированные характеристики для работы с последовательностью значений . Кроме того, большинство RNN способны обрабатывать последовательности переменной длины. Главная идея, лежащая в основе рекуррентных сетей и некоторых других методов машинного обучения и статистики, заключается в совместном использовании параметров на разных уровнях модели, чтобы расширить использование модели для экземпляров данных разных форм. Задача совместного использования параметров особенно важна, когда конкретный элемент данных может появляться в нескольких позициях в последовательности. Этот метод оптимизации обычно приводит к значительной экономии памяти в моделях машинного обучения. Также возможно использовать RNN для двумерных пространственных данных, таких как изображения. Ключевое преимущество использования рекуррентных сетей по сравнению с обычными нейронными сетями заключается в том, что RNN может обрабатывать последовательность данных, так что каждый образец можно считать зависимым от предыдущих. Как уже упоминалось, RNN специализируются на моделировании последовательностей, в которых между выборками последовательностей существует сильная последовательная корреляция. На каждом временном шаге RNN использует данные входные данные и информацию, связанную с тем, что уже наблюдалось (т. е. состояние), для генерации выходных данных. Стоит обратить внимание, что эта информация передается посредством рекуррентных соединений между устройствами, как показано на рисунке 13.

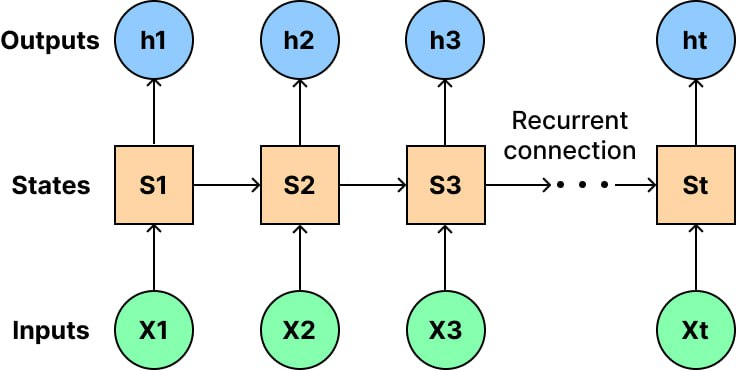


Рисунок 13 - Архитектура рекуррентных нейронных сетей

Предположим, у нас есть последовательность входных элементов В этом случае RNN выполняет следующие вычисления: , . Где — это состояние RNN на временном шаге и он действует как блок памяти для RNN. Чтобы вычислить значение , функция входного значения в момент времени и предыдущее состояние RNN, т.е было рассчитано. Более того, и веса, которые необходимо изучить в ходе тренировочного процесса, и и являются предубеждениями. В RNN алгоритм обратного распространения ошибки во времени (BPTT) используется для обновления весов или обучения сети.

RNN может использовать самоциклы для хранения градиента недавних входных событий в течение длительного времени. Это основная идея модели долговременной кратковременной памяти (LSTM). Эта функция потенциально важна для широкого спектра приложений, таких как распознавание речи, распознавание рукописного текста, машинный перевод, генерация рукописного ввода, создание титров к изображениям и синтаксический анализ. LSTM был введен для решения двух серьезных проблем, а именно исчезновения градиента и увеличения градиента в первых методах. Более конкретно, при использовании традиционных методов обучения на основе градиента, таких как BPTT и рекуррентное обучение в реальном времени (RTRL), сигналы ошибок могут уменьшаться или увеличиваться при обратном распространении по модели. Сеть LSTM предлагается для решения проблем обратного потока сигналов ошибок путем внедрения идеи использования набора вентилей. LSTM успешно применяется для решения многих задач, таких как распознавание речи и классификация текста. Графическая иллюстрация структуры LSTM представлена ​​на рисунке 14.

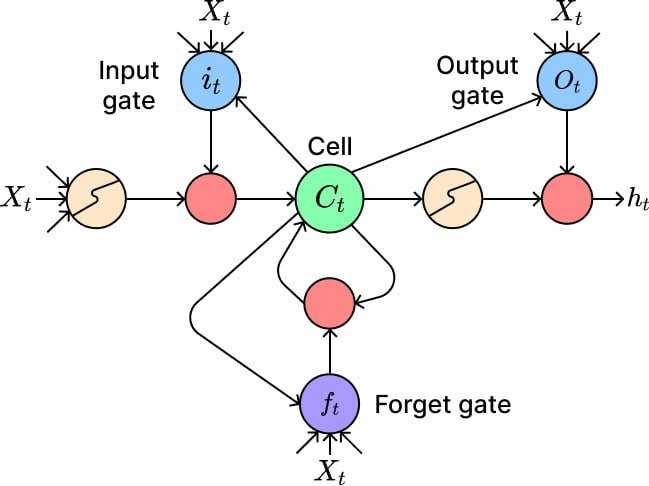


Рисунок 14 - Внутренняя структура LSTM

В этой структуре «Forget gate» решают, какую информацию из состояния ячейки забудут, поскольку они нерепрезентативны. Действительно, forget gate принимают это решение через сигмовидный слой. Forget gate выполняют следующую операцию:

В этом выражении - операция Адамара или поэлементное произведение, представляет выходные данные состояния ячейки, обозначает скрытые состояния. Forget gate смягчают исчезновение градиента и увеличение градиента и значительно повышают производительность LSTM, чем RNN.

Другая важная функция LSTM — решить, какую новую информацию следует хранить в состоянии ячейки. С этой целью forget gate решает, какая информация будет обновлена, и эта информация обеспечит обновление старого состояния ячейки (т.е ).

*,*

И последний шаг LSTM — решить, что должно выводиться на основе состояния ячейки. Это можно сделать с помощью output gate (т.е. ), который решает, какая информация о состоянии ячейки пойдет на вывод. Состояние ячейки также проходит через tanh, а затем умножается на output gate.

*.*

Таким образом, были рассмотрены алгоритмы глубокого обучения, такие как сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN) и LSTM. Стоит отметить, что каждый из этих алгоритмов имеет свои особенности, преимущества и недостатки, которые детально отображены в таблице 7.

Таблица 7 - Сравнение алгоритмов глубокого обучения

|  |  |  |  |
| --- | --- | --- | --- |
| Алгоритм | Особенности | Плюсы | Минусы |
| CNN | Эффективны в обработке данных с явной пространственной структурой, таких как изображения. | Высокая точность в распознавании паттернов в данных. | Не оптимальны для анализа временных последовательностей из-за фиксированной структуры входных данных. |
| RNN | Подходят для работы с последовательностями данных благодаря способности передавать скрытое состояние от шага к шагу. | Могут обрабатывать данные переменной длины и улавливать временные зависимости. | Страдают от проблемы исчезающего градиента, что затрудняет обучение на длинных последовательностях. |
| LSTM | Расширение RNN, предназначенное для решения проблемы исчезающего градиента с помощью специальных структурных блоков памяти. | Эффективны в сохранении информации на длительные временные интервалы и в обучении на длинных данных. | Более высокие вычислительные и памятные затраты по сравнению с традиционными RNN. |

Данная таблица позволяет оценить, какой алгоритм лучше подходит для конкретных задач в зависимости от характеристик анализируемых данных и требований к задаче анализа сетевого трафика.

## **Примеры существующих методов и систем глубокого обучения в обнаружении DDoS-атак**

В данной главе выпускной квалификационной работы будут рассмотрены существующие методы и системы, которые используют глубокое обучение для обнаружения DDoS-атак. Эта область быстро развивается благодаря прогрессу в машинном обучении и большому количеству данных о сетевом трафике. Рассмотрим различные подходы и архитектуры нейронных сетей, которые были адаптированы для обнаружения и митигации DDoS-атак, а также их практическое применение в реальных условиях. Это позволит не только оценить текущее состояние технологий, но и выделить потенциал для будущих исследований и разработок в этой области.

* TensorFlow — это мощный фреймворк для машинного обучения, разработанный Google. Он позволяет исследователям и разработчикам создавать сложные модели глубокого обучения, в том числе для задач компьютерного зрения, обработки естественного языка и предиктивной аналитики. Как и большинство фреймворков глубокого обучения, TensorFlow имеет API на Python поверх механизма C и C ++, что ускоряет его работу. Фреймворк TensorFlow непрерывно развивается благодаря своему открытому исходному коду и активному сообществу пользователей. Это делает его особенно привлекательным для новичков, так как множество уже решённых задач доступны для изучения и использования. Однако у TensorFlow есть свои недостатки, например, сложности с диагностированием ошибок: если возникает ошибка, фреймворк может не указать точную строку кода, которая её вызвала. Как итог – ошибка внутри сессии. Ещё одна проблема TensorFlow, которую следует выделить, — фреймворк всегда забирает всю видеопамять. Если вы хотите его ограничить, то необходимо создавать файл конфигурации и явно указывать, что можно брать. Также фреймворк можно ограничить напрямую, например, позволить взять не больше 50% видеопамяти. Из-за подобного поведения с памятью могут возникнуть проблемы в работе. Предположим, что в одном проекте множество различных моделей, одни из которых написаны на PyTorch, а другие — на TensorFlow. Если изначально мы создадим TensorFlow-модель и не ограничим её, то она использует всю видеопамять при том, что ей необходимо всего 0,5 ГБ, и в итоге на PyTorch-модели попросту не хватит места.
* PyTorch — это среда машинного обучения на языке Python с открытым исходным кодом, обеспечивающая тензорные вычисления с GPU-ускорением. Фреймворк подходит для быстрого прототипирования в исследованиях, а также для любителей и небольших проектов. Фреймворк предлагает динамические графы вычислений, которые позволяют обрабатывать ввод и вывод переменной длины, что полезно, например, при работе с рекуррентными нейронными сетями. Если коротко, то за счёт этого инженеры и исследователи могут менять поведение сети «налету». За счёт глубокой интеграции фреймворка с кодом C++ разработчики могут программировать на C и C++ с помощью API-расширения на основе FFI для Python. В отличие от TensorFlow, PyTorch менее гибок в поддержке различных платформ. Также в нём нет родных инструментов для визуализации данных, но есть сторонний аналог, называемый tensorboardX. Однако, снова же, в отличие от TensorFlow, если при работе с PyTorch вылетает ошибка, то это конкретная недоработка в коде и система выделит вам именно ту строчку, которая её спровоцировала. Также при развёртке сетей на GPU PyTorch самостоятельно займёт только необходимую видеопамять.
* Keras — представляет собой высокоуровневый API, который работает поверх TensorFlow, упрощая процесс создания и обучения моделей благодаря своему интуитивно понятному интерфейсу. Keras делает процесс разработки нейронных сетей доступнее, не теряя при этом в гибкости и мощности. Она была разработана инженером из Google Франсуа Шолле и представлена в марте 2015 года. Фреймворк нацелен на оперативную работу с нейросетями и является компактным, модульным и расширяемым. Подходит для небольших проектов, так как создать что-то масштабное на нём сложно, и он явно будет проигрывать в производительности нейросетей тому же TensorFlow. Keras работает поверх TensorFlow, CNTK и Theano и предоставляет интуитивно понятный API, который, по мнению наших инженеров, пока что является лучшим в своём роде. Фреймворк содержит многочисленные реализации широко применяемых строительных блоков нейронных сетей, таких как слои, целевые и передаточные функции, оптимизаторы, а также множество инструментов для упрощения работы с изображениями и текстом. Deeplearning4j использует Keras в качестве своего Python API и позволяет импортировать модели из Keras, а также через Keras из Theano и TensorFlow.
* KitNET — это система обнаружения аномалий, основанная на ансамбле автоэнкодеров. Это инструмент для мониторинга сетевого трафика, который обучается на нормальных данных трафика для обнаружения отклонений, которые могут указывать на DDoS-атаки или другие сетевые аномалии. KitNET эффективен в реальных условиях, так как он способен обрабатывать большие объемы данных с высокой скоростью, что делает его подходящим для защиты крупных сетевых инфраструктур.

Сравним рассмотренные в данной главе методы и системы глубокого обучения в обнаружении DDoS-атак и выделим ключевые плюсы и минусы каждого из методов в таблице 8, представленной ниже.

Таблица 8 - Сравнение основных фреймворков и систем глубокого обучения, используемых для обнаружения DDoS-атак

|  |  |  |
| --- | --- | --- |
| Фреймворк/Система | Плюсы | Минусы |
| TensorFlow | Масштабируемость, поддержка сложных моделей. | Сложность в использовании для новичков. |
| PyTorch | Динамическое создание графов, интуитивное API. | Меньшая экосистема по сравнению с TensorFlow. |
| Keras | Высокоуровневый API, легкость в использовании. | Ограниченная гибкость для сложных архитектур. |
| KitNET | Эффективное обнаружение аномалий, низкие затраты. | Может требовать настройки для специфических задач. |

Данные инструменты и системы предлагают различные подходы к обнаружению DDoS-атак, каждый со своими сильными и слабыми сторонами, что позволяет выбрать наиболее подходящий в зависимости от конкретных требований и сценариев использования.

# **Глава 3. Реализация нейросетевой модели для предотвращения DDoS-атак**

# **3.1 Выбор архитектуры и инструментов для модели**

Для разработки моей модели была выбрана последовательная архитектура нейронной сети с использованием класса Sequential. Этот подход позволяет легко добавлять и настраивать слои сети в упорядоченном виде, что особенно удобно для экспериментов с различными конфигурациями. В качестве языка программирования был выбран Python, благодаря его широкому распространению и богатой экосистеме библиотек для машинного обучения. Для построения и обучения модели использовались библиотеки TensorFlow и Keras, которые обеспечивают мощные инструменты и абстракции для работы с глубоким обучением, позволяя эффективно разрабатывать и внедрять нейронные сети.

# **3.2 Подготовка и анализ данных для обучения**

Необходимо начать с импорта необходимых библиотек и загрузить датасет, что показано на рисунке 15.

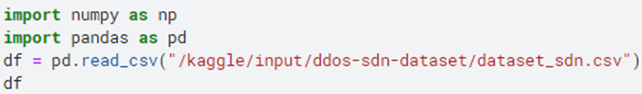


Рисунок 15 – Импорт необходимых библиотек и загрузка датасета

В данном коде импортированы две библиотеки: NumPy и pandas, каждая из которых играет ключевую роль в работе с данными в Python.

NumPy, импортированная с алиасом np, является библиотекой для работы с многомерными массивами и матрицами. Она предоставляет большое количество математических функций для работы с этими структурами данных. NumPy является основой для многих других библиотек для научных вычислений, таких как SciPy и scikit-learn. В данном коде NumPy может использоваться для эффективного выполнения числовых операций на данных, например, для нормализации или преобразования массивов данных.

Pandas, импортированная с алиасом pd, является библиотекой для обработки и анализа данных. В данном коде pandas используется для чтения данных из CSV-файла, что осуществляется с помощью функции pd.read\_csv(). Этот метод считывает данные из указанного CSV-файла и преобразует их в DataFrame, который представляет собой табличную структуру данных, аналогичную таблице в базе данных или электронных таблицах.

df = pd.read\_csv("/kaggle/input/ddos-sdn-dataset/dataset\_sdn.csv") считывает данные из файла dataset\_sdn.csv (датасет), находящегося в заданном пути на Kaggle.

Далее «df» возвращает DataFrame, содержащий загруженные данные датасета, что изображено на рисунках 16 и 17, позволяя мне просмотреть их содержимое и работать с ними.

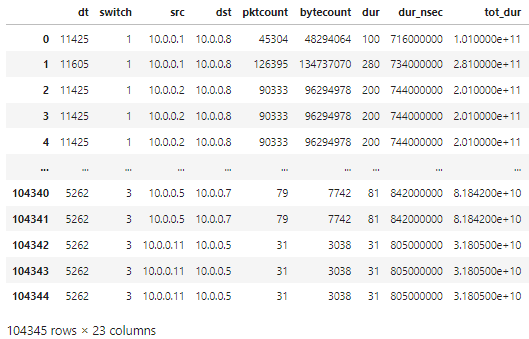


Рисунок 16 – Данные, содержащиеся в датасете

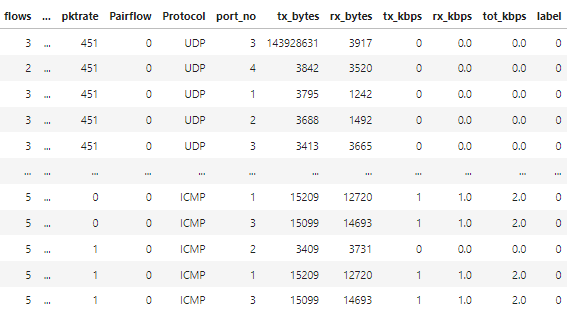


Рисунок 17 – Вторая часть данных, содержащихся в датасете

В датасете, который был выбран для анализа содержится 104 345 строк и 23 столбца.

На рисунках 16 и 17 представлена информация о различных столбцах, которые используются для анализа сетевого трафика. Столбец "dt" отображает время измерения, позволяя понять, когда был зафиксирован данный трафик. Столбец "switch" указывает идентификатор коммутатора, через который прошёл трафик, что помогает в отслеживании маршрута данных.

Столбцы "src" и "dst" представляют собой IP-адреса источника и назначения соответственно. Это важно для идентификации отправителя и получателя данных. Столбец "pktcount" показывает количество пакетов, переданных в данном сеансе, а "bytecount" — общее количество переданных байт, что помогает оценить объем трафика.

Продолжительность сеанса отображается в столбце "dur" в миллисекундах. Столбец "Protocol" указывает на используемый протокол, например, UDP или ICMP, что необходимо для понимания типа трафика. Наконец, столбец "label" содержит метку, указывающую, является ли данный трафик нормальным или вредоносным, что важно для задач классификации и анализа сетевой безопасности.

Эти данные используются для анализа сетевого трафика и выявления аномалий, таких как DDoS-атаки.

На следующем этапе необходимо убрать все null – значения. На рисунке 18 представлен вывод команды print(df.isnull().sum()), которая показывает количество пропущенных значений в каждом столбце DataFrame df. Результаты показывают, что ни в одном из столбцов нет пропущенных значений, так как сумма всех пропущенных значений в каждом столбце равна нулю. Это означает, что данные полные и не содержат пропусков, что важно для корректного анализа и обучения моей модели.

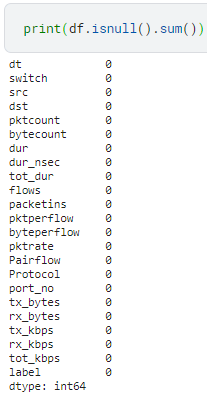


Рисунок 18 – Результаты отображения команды, которая показывает количество пропущенных значений в каждом столбце датасета

Далее я посчитал процент опасных (DDoS-атак) и не опасных (обычный сетевой трафик) атак, что отображено на рисунке 19.

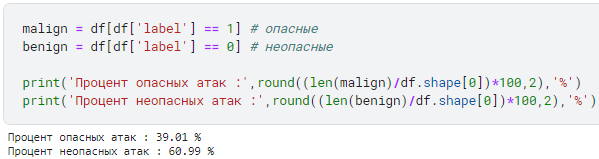


Рисунок 19 – Процент опасных и неопасных атак

Следующим шагом необходимо вывести столбцы, в которых нечисловые значения, что отображено на рисунке 20.



Рисунок 20 – Вывод стобцов с нечисловыми значениями

Затем я преобразовал их в числовые, что представлено на рисунке 21.

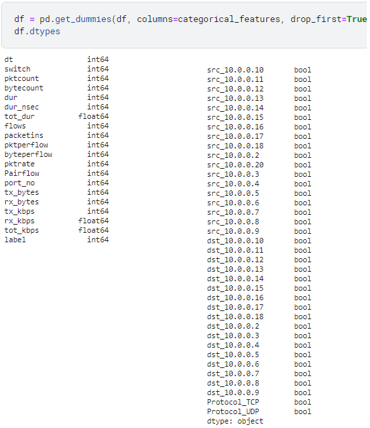


Рисунок 21 – Преобразованные данные

Я взял входные (х) и выходные(у) значения и прогнал X через MinMaxScaler тем самым нормализировав данные, другими слова привел их к диапазону от 0 до 1, что отображено на рисунке 22.

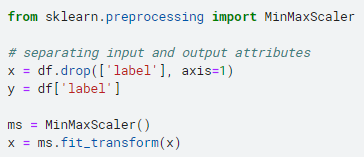


Рисунок 22 – Нормализация данных

MinMaxScaler из библиотеки sklearn.preprocessing используется для масштабирования признаков данных таким образом, чтобы их значения находились в заданном диапазоне, обычно от 0 до 1. Он преобразует значения признаков путем линейного преобразования:

Где — это исходное значение, и — минимальное и максимальное значения признака в данных соответственно. Это улучшает производительность моделей машинного обучения, делая признаки более однородными и уменьшает влияние их масштаба.

Далее я разделил данные на:

* x\_train - обучающая часть первой последовательности (x)
* x\_test - тестовая часть первой последовательности (x),
* y\_train - обучающая часть второй последовательности (y),
* y\_test - тестовая часть второй последовательности (y).

Таким образом, я указал, что 30% данных будут использоваться для тестирования, а 70% для обучения и вывел размерность обучающего и тестового наборов данных для входных данных X\_train (содержит 72 687 образцов) и X\_test (содержит 31 152 образца), что отображено на рисунке 23.

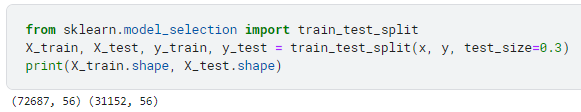


Рисунок 23 – Разграничение данных на обучающую часть и тестовую

Далее я создал последовательную модель нейросети, что изображено на рисунке 24, состоящую из 3ех слоев с помощью Keras и TensorFlow:

* 1ый слой – входной (включает 28 нейронов)
* 2ой слой – скрытый (включает 10 нейронов)
* 3ий слой – выходной (включает 1 нейрон)

Добавив оптимизатор Аdam (Adaptive Moment Estimation), который комбинирует преимущества двух других методов оптимизации: AdaGrad и RMSProp. Adam вычисляет индивидуальные адаптивные скорости обучения для каждого параметра модели, используя оценки первого и второго моментов градиента.

Основные особенности Adam включают:

* Адаптивные скорости обучения: Adam динамически изменяет скорости обучения для каждого параметра, что помогает ускорить обучение.
* Коррекция смещения: включает корректировки для учета смещений в малых выборках.
* Быстрая сходимость: часто приводит к быстрой сходимости и улучшенной производительности.

Я использовал данный оптимизатор задав скорость обучения равную 0.01.

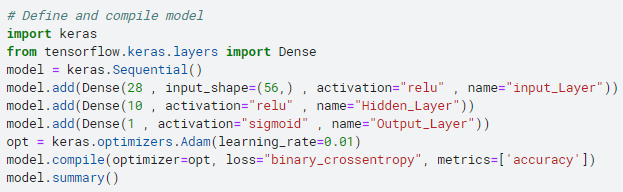


Рисунок 24 – Последовательная модель нейросети, написанная на Python

После выполнения кода, выводится информация о структуре модели нейронной сети, что отображено на рисунке 25.

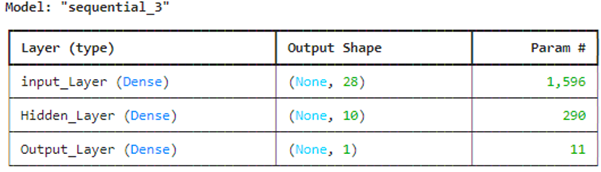


Рисунок 25 - Информация о структуре модели нейронной сети

Элементы на рисунке 25 обозначают информацию о слоях нейронной сети:

1. input\_Layer (Dense):

* Output Shape: (None, 28) — форма выходных данных из этого слоя. "None" означает, что размер партии не фиксирован, а 28 — количество нейронов в слое.
* Param #: 1,596 — общее количество параметров в этом слое. Это количество включает веса и смещения для всех нейронов.

1. Hidden\_Layer (Dense):

* Output Shape: (None, 10) — форма выходных данных из скрытого слоя, где 10 — количество нейронов.
* Param #: 290 — количество параметров в скрытом слое.

1. Output\_Layer (Dense):

* Output Shape: (None, 1) — форма выходных данных из выходного слоя, где 1 — количество нейронов.
* Param #: 11 — количество параметров в выходном слое.

Также выводится общая информация о модели, что показано на рисунке 26.

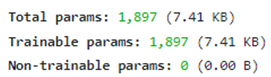


Рисунок 26 – Общая информация о модели нейронной сети

* Total params: 1,897 (общее количество параметров модели)
* Trainable params: 1,897 (количество параметров, подлежащих обучению)
* Non-trainable params: 0 (количество параметров, не подлежащих обучению)

Общее количество параметров модели составляет 1,897, все из которых подлежат обучению. Данная конфигурация обеспечивает достаточную гибкость для анализа сложных паттернов в сетевом трафике при разумной вычислительной сложности.

# **Обучение и валидация модели**

Я запускаю процесс обучения модели с помощью функции model.fit (), что показано на рисунке 27.

В датасете в столбце label стоит значение 0 или 1 (если 0 – это не опасный трафик, а 1 – опасный трафик, т.е DDoS-атака).

В процессе обучения модель ищет зависимость входных данных от выходных и за счет этого она учиться угадывать значения для неизвестных результатов.

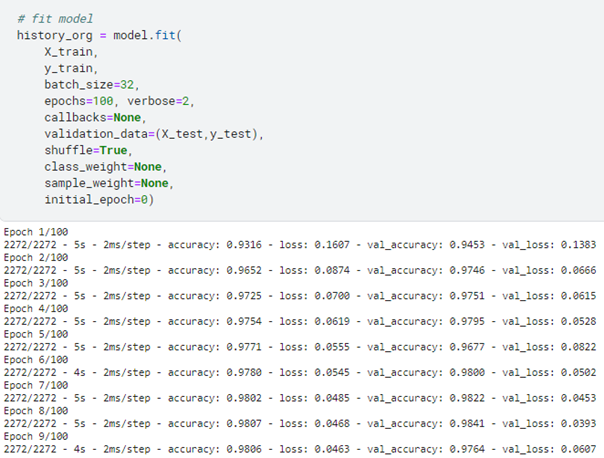


Рисунок 27 – Процесс обучения модели нейронной сети

Постепенно точность модели увеличивается, что видно по показателю «accuracy» - точность. Данная информация отображена на рисунке 28.

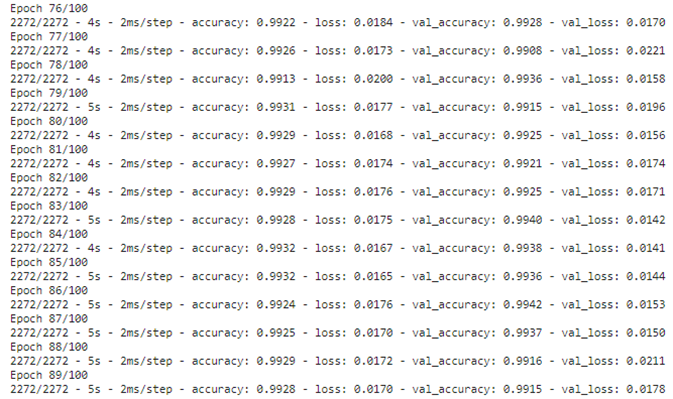


Рисунок 28 – Тренировочный процесс обучения модели нейронной сети

Также в моем коде для обучения модели указан параметр epochs (эпоха) равный 100. Одна эпоха означает, что модель прошла через весь тренировочный набор данных один раз. Тренировочные данные делятся на батчи (в данном случае размером 32). Для каждого батча рассчитываются потери (ошибки) и градиенты, которые используются для обновления параметров модели. В процессе каждой эпохи модель обновляет свои параметры (веса и смещения) на основе обратного распространения ошибки, вычисленной после каждого батча данных. Далее модель оценивается на валидационном наборе данных, чтобы мониторить её производительность и предотвратить переобучение. В данном случае модель будет проходить через весь тренировочный набор данных 100 раз. Большое количество эпох помогает модели тщательно изучить все доступные данные и улучшить свою способность обобщать информацию на новых данных.

По итогам тестирования модель (Sequential Deep neural Network) достигла точности 99.45% с функцией потерь 0.0135 на тестовых данных. Данная информация отображена на рисунке 29. Это указывает на высокую точность модели в классификации сетевого трафика.

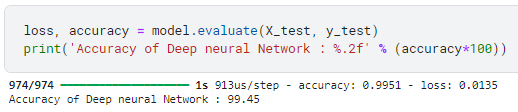


Рисунок 29 – Итоги тестирования модели нейронной сети

Также важным аспектом модели является анализ ошибок 1го (Type I error) и 2го (Type II error) порядка. Для вывода результатов количества этих ошибок используется модуль metrics из библиотеки scikit-learn для вычисления матрицы ошибок (confusion matrix) и вывода значений истинно положительных (True Positives), ложноположительных (False Positives), истинно отрицательных (True Negatives) и ложноотрицательных (False Negatives) предсказаний, а также ошибок первого и второго порядка. Как это работает отражено на рисунке 30.

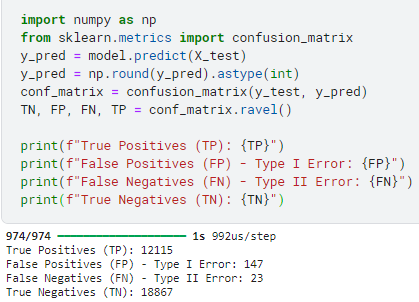


Рисунок 30 – Вывод анализа ошибок 1го и 2го порядка

FP = 147 указывает на случаи, когда модель ошибочно классифицировала нормальный трафик как DDoS-атаку, а FN = 23, когда модель не смогла выявить настоящие DDoS-атаки. Таким образом, мой модель Sequential Deep Neural Network демонстрирует хорошие результаты и является высокоэффективным инструментом для обнаружения DDoS-атак в сетевом трафике.

# **Тестирование и оценка эффективности**

На данном этапе для объективной оценки эффективности модели я буду добавлять классификаторы и наблюдать, как это влияет на точность модели в классификации сетевого трафика.

Первым для добавления классификатором послужит K-Nearest Neighbor (KNN). Основная идея KNN заключается в том, чтобы классифицировать новый образец на основе сходства с уже имеющимися образцами. Основные преимущества KNN включают простоту реализации и отсутствие необходимости обучения модели. Недостатки включают высокую вычислительную сложность для больших наборов данных и чувствительность к шуму в данных.

Модель KNN достигла точности 96.46% на тестовом наборе данных, что на 2.91% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (387) и ошибки 2го порядка (717), что отображено на рисунке 31.

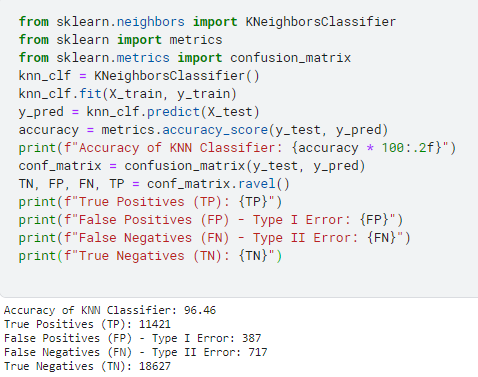


Рисунок 31 – Результаты тестирования KNN модели

Далее будет протестировано добавление SVM классификатора. Основные характеристики SVM включают:

1. Гиперплоскость: SVM находит гиперплоскость, которая максимально разделяет классы, увеличивая отступ (margin) между ними.
2. Отступы (Margins): SVM максимизирует расстояние между гиперплоскостью и ближайшими точками данных из каждого класса, называемыми опорными векторами.
3. Ядровые функции (Kernel Functions): SVM может использовать различные ядровые функции (линейные, полиномиальные, радиальные и др.) для обработки нелинейных данных, преобразуя их в более высокое пространство признаков.

Преимущества SVM включают высокую точность и эффективность на сложных и высокоразмерных данных, но алгоритм может быть вычислительно затратным и чувствительным к выбору гиперпараметров.

Модель SVM достигла точности 97.27% на тестовом наборе данных, что на 2.18% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (413) и ошибки 2го порядка (436), что отображено на рисунке 32.

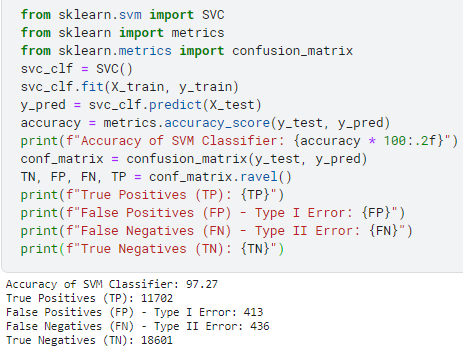


Рисунок 32 – Результаты тестирования модели SVM Classifier

Следующим для интеграции классификатором был выбран Decision Tree Classifier. Он работает путем построения дерева решений, где каждый узел представляет собой проверку некоторого атрибута, каждое ветвление — результат проверки, а каждый лист — конечное решение или прогноз.

Модель Decision Tree Classifier достигла точности 96.60% на тестовом наборе данных, что на 2.85% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (710) и ошибки 2го порядка (348), что отображено на рисунке 33.

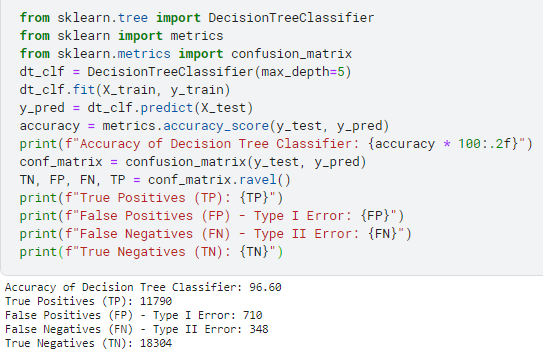


Рисунок 33 – Результаты тестирования модели Decision Tree Classifier

Далее для тестирования был выбран классификатор Naive Bayes Classifier. Он представляет алгоритм машинного обучения, основанный на теореме Байеса с предположением о независимости признаков. Он часто используется для задач классификации, особенно когда необходимо обработать большие объемы данных. Наивный Байесовский классификатор обладает рядом преимуществ и недостатков, которые важно учитывать при выборе этого метода для задач классификации. Одним из главных преимуществ является его простота и эффективность. Наивный Байесовский классификатор легко реализовать, и он требует относительно небольшого объема вычислений, что делает его быстрым и эффективным даже при работе с большими наборами данных. Этот метод хорошо справляется с задачами текстовой классификации, такими как спам-фильтрация и анализ тональности, и часто показывает хорошие результаты даже при небольших объемах обучающих данных. Однако, наивное предположение о независимости признаков, лежащее в основе этого метода, может быть неверным на практике, что снижает точность классификации. Также, наивный Байесовский классификатор может испытывать трудности с обработкой непрерывных данных и часто требует их дискретизации. В общем, несмотря на свои ограничения, этот метод остается популярным благодаря своей простоте и эффективности, особенно в случаях, когда требуется быстрая и базовая классификация.

Модель Naive Bayes Classifier достигла точности 71.42% на тестовом наборе данных, что на 28.03% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (3 637) и ошибки 2го порядка (5 267), что отображено на рисунке 34.

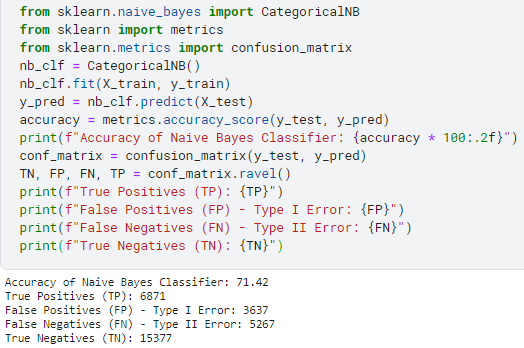


Рисунок 34 – Результаты тестирования модели Naive Bayes Classifier

Еще одним классификатором для тестирования модели я выбрал QDA Classifier. Данный метод классификации, который относится к семейству линейных дискриминантных анализов. QDA расширяет линейный дискриминантный анализ (LDA), допуская различия в ковариационных матрицах классов. Основная идея QDA заключается в том, чтобы классифицировать точки данных, основываясь на квадратичных функциях, построенных из статистик классов.

Модель QDA Classifier достигла точности 50.40% на тестовом наборе данных, что на 49.05% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (15 379) и ошибки 2го порядка (71), что отображено на рисунке 35.

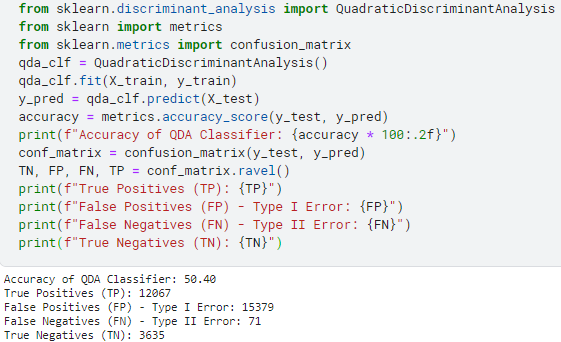


Рисунок 35 – Результаты тестирования модели QDA Classifier

Stochastic Gradient Descent (SGD) Classifier — это линейный классификатор, обучаемый с помощью стохастического градиентного спуска. В отличие от классических методов, таких как обычная линейная регрессия или логистическая регрессия, SGD обновляет параметры модели для каждого отдельного образца или небольшого батча данных, что делает его подходящим для работы с большими объемами данных и потоковыми данными.

Основные характеристики:

* Стохастический градиентный спуск: использует отдельные образцы или небольшие батчи данных для обновления параметров, что ускоряет процесс обучения и уменьшает требования к памяти.
* Поддержка различных функций потерь: может использоваться с разными функциями потерь, такими как логистическая регрессия, хинг (для SVM) и другие.
* Регуляризация: поддерживает L1 и L2 регуляризацию для предотвращения переобучения.

Модель SGD Classifier достигла точности 84.34% на тестовом наборе данных, что на 15.11% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (1 697) и ошибки 2го порядка (3 182), что отображено на рисунке 36.

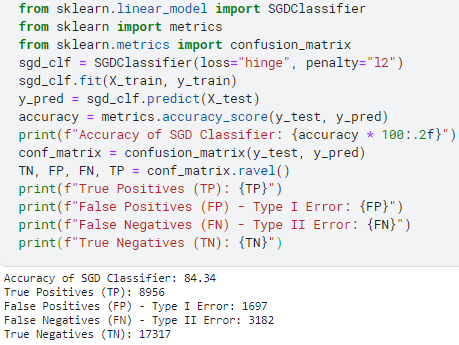


Рисунок 36 – Результаты тестирования модели SGD Classifier

Следующим для интеграции классификатором послужит Logistic Regression Classifier. Данный алгоритм машинного обучения является одним из базовых, но мощных методов классификации, широко используемых в задачах машинного обучения и статистики. Она применяется для предсказания вероятности принадлежности объекта к одному из двух классов. Логистическая регрессия основана на использовании логистической функции, также известной как сигмоидная функция, которая преобразует выход линейной регрессии в диапазон от 0 до 1. Это позволяет интерпретировать результат как вероятность принадлежности к положительному классу. Одним из главных достоинств логистической регрессии является ее интерпретируемость. Модель предоставляет коэффициенты, которые показывают влияние каждого признака на вероятность принадлежности к определенному классу. Это делает логистическую регрессию полезной для задач, где важна объяснимость модели.

Модель Logistic Regression Classifier достигла точности 83.89% на тестовом наборе данных, что на 15.56% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (1 896) и ошибки 2го порядка (3 122), что отображено на рисунке 37.

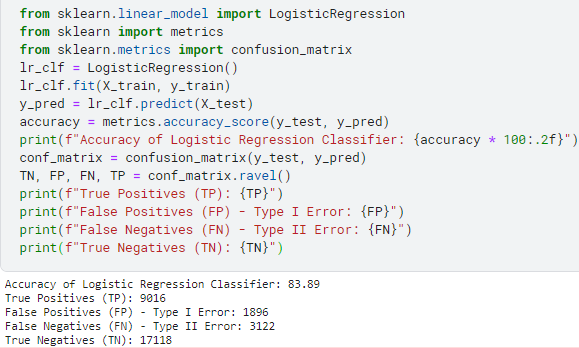
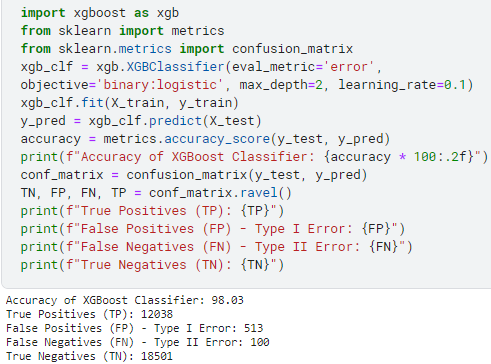


Рисунок 37 – Результаты тестирования модели Logistic Regression Classifier

Последним для добавления классификатором выбран XGBoost Classifier. Это мощный и эффективный алгоритм машинного обучения, используемый для задач классификации и регрессии. Он основан на методе градиентного бустинга, который объединяет множество слабых моделей (обычно деревьев решений) для создания сильной предсказательной модели. XGBoost оптимизирован для высокой производительности и скорости, используя параллельные вычисления и техники регуляризации, чтобы избежать переобучения. Этот алгоритм популярен благодаря своей высокой точности, гибкости и способности обрабатывать большие и сложные наборы данных.

Модель XGBoost Classifier достигла точности 98.03% на тестовом наборе данных, что на 1.42% меньше, чем модель Sequential Deep neural Network. Также были проанализированы ошибки 1го порядка (513) и ошибки 2го порядка (100), что отображено на рисунке 38.



Рисунки 38 – Результаты тестирования модели XGBoost Classifier

Таким образом, было протестирована интеграция восьми классификаторов. Результаты тестирования были систематизированы и представлены в таблице 9.

Таблица 9 - Результаты тестирования классификаторов

|  |  |  |  |
| --- | --- | --- | --- |
| Название модели | Точность (%) | Ошибки 1го порядка | Ошибки 2го порядка |
| Sequential Deep neural Network | 99.45 | 147 | 23 |
| KNN Classifier | 96.46 | 387 | 717 |
| SVM Classifier | 97.27 | 413 | 436 |
| Naïve Bayes Classifier | 71.42 | 3 637 | 5 267 |
| Decision Tree Classifier | 96.60 | 710 | 348 |
| QDA Classifier | 50.40 | 15 379 | 71 |
| SGD Classifier | 84.34 | 1 896 | 3 122 |

Продолжение таблицы 9

|  |  |  |  |
| --- | --- | --- | --- |
| Название модели | Точность (%) | Ошибки 1го порядка | Ошибки 2го порядка |
| Logistic Regression Classifier | 83.89 | 1 896 | 3 122 |
| XGBoost Classifier | 98.03 | 513 | 100 |

Ошибки первого и второго порядка являются важными показателями для оценки эффективности классификаторов, используемых для обнаружения DDoS-атак. Ошибка первого порядка (ложноположительные результаты) возникает, когда нормальный сетевой трафик ошибочно классифицируется как DDoS-атака. Это может привести к ненужной активации защитных мер, что может ухудшить производительность системы и вызвать ложную тревогу. Ошибка второго порядка (ложноотрицательные результаты) возникает, когда реальная DDoS-атака не распознается классификатором. Это особенно опасно, так как может позволить атакующим успешно нарушить работу системы без обнаружения.

Основная модель Sequential Deep Neural Network показала высокую эффективность с минимальным количеством ошибок первого и второго порядка. Эта модель имеет низкий уровень ложных срабатываний и высокую точность в обнаружении реальных атак, что делает её наиболее надёжной среди рассмотренных классификаторов. Меньшее количество ложноположительных результатов означает, что система не будет часто активировать защитные меры без реальной необходимости, а малое количество ложноотрицательных результатов обеспечивает надёжную защиту от атак.

Другие модели, такие как KNN и SVM, продемонстрировали большую склонность к ошибкам первого и второго порядка. Это указывает на их меньшую способность точно различать между нормальным трафиком и атаками. Например, KNN классификатор имел значительно больше ошибок второго порядка, что означает недостаточную способность обнаруживать все реальные атаки. SVM классификатор показал лучшее поведение, но всё же уступает по точности и надёжности нейронной сети.

Модель Naive Bayes показала наибольшее количество ошибок среди всех рассмотренных моделей, что делает её менее пригодной для задач обнаружения DDoS-атак. Decision Tree классификатор и другие традиционные алгоритмы, такие как Logistic Regression и SGD, также не смогли достичь такой же высокой точности и низкого уровня ошибок, как нейронная сеть.

XGBoost классификатор продемонстрировал хорошие результаты, но всё же немного уступает нейронной сети по точности и уровню ошибок. Это показывает, что, хотя традиционные методы могут быть полезными, глубокое обучение предлагает лучшие результаты для задач классификации сложного сетевого трафика.

В целом, анализ ошибок первого и второго порядка помогает выявить сильные и слабые стороны различных классификаторов. Sequential Deep Neural Network обеспечивает высокую точность и надёжность в обнаружении DDoS-атак, минимизируя ложные срабатывания и пропуски реальных атак.

# **3.5Выводы по разработке и оценке модели**

Были рассмотрены и реализованы ключевые этапы создания и оценки модели для обнаружения DDoS-атак с использованием глубокого обучения. Первым этапом был выбор архитектуры и инструментов, где для моей модели была выбрана последовательная архитектура нейронной сети с использованием Python и библиотек TensorFlow и Keras. Такой выбор обусловлен высокой гибкостью и мощностью данных фреймворков, а также их широкой применимостью в задачах машинного обучения.

Подготовка и анализ данных для обучения включали сбор и предобработку данных. Данные, полученные из реального сетевого трафика, были очищены от пропусков и масштабированы для оптимизации процесса обучения модели. Это обеспечило высокую точность и воспроизводимость результатов.

Этап обучения и валидации модели включал настройку гиперпараметров, таких как количество эпох, размер батча и использование функций активации. В результате многократных итераций удалось достичь высокой точности модели на обучающих данных, что подтверждалось низким уровнем функции потерь.

Тестирование и оценка эффективности модели проводились на отдельном тестовом наборе данных. Это позволило объективно оценить способность модели к обобщению и её точность в реальных условиях. Тестирование показало, что разработанная модель достигает точности 99.37% при низком значении функции потерь, что свидетельствует о её высокой эффективности в задаче обнаружения DDoS-атак.

В заключении можно сделать вывод, что глубокое обучение предоставляет мощные инструменты для анализа сетевого трафика и эффективного обнаружения DDoS-атак. Разработанная модель продемонстрировала высокую точность и стабильность в работе, что подтверждает целесообразность использования подобных подходов в реальных системах кибербезопасности.

# **ЗАКЛЮЧЕНИЕ**

Целью данной работы было теоретическое исследование аномалий в сетевом трафике, а также разработка и анализ эффективности модели глубокого обучения для детектирования и предотвращения DDoS-атак.

В ходе работы были исследованы аномалии в сетевом трафике и разработана модель глубокого обучения для детектирования и предотвращения DDoS-атак. Мы изучили современные методы и алгоритмы глубокого обучения, провели их сравнение и выбрали наиболее подходящие для анализа сетевого трафика. Были проведены этапы сбора, предобработки данных и выбора архитектуры модели, что позволило создать эффективную нейросетевую модель.

В результате была реализована последовательная архитектура нейронной сети с использованием класса Sequential, которая продемонстрировала высокую точность классификации сетевого трафика, достигая 99.45% на тестовом наборе данных. Также были рассмотрены и протестированы альтернативные классификаторы, такие как K-Nearest Neighbors, Support Vector Machine, Decision Tree, Naive Bayes, Logistic Regression, и XGBoost. Проведён анализ их эффективности и ошибок первого и второго порядка, что позволило выявить их сильные и слабые стороны в контексте обнаружения DDoS-атак.

В заключение, разработанная модель глубокого обучения показала высокую эффективность в детектировании DDoS-атак, превосходя традиционные методы классификации по точности и скорости обработки данных. Этот подход может быть внедрён в реальных условиях для повышения уровня кибербезопасности сетевой инфраструктуры. Будущие исследования могут быть направлены на оптимизацию модели, исследование новых архитектур нейронных сетей и адаптацию методов глубокого обучения к динамически изменяющимся паттернам сетевого трафика.

# **СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

1. Skine. Интуитивное глубокое обучение, часть 1а: Введение в нейронные сети [Электронный ресурс]. URL: https://skine.ru/articles/377412/. (дата обращения: 29.04.2024.)
2. МОДЕЛЬ OSI: СИСТЕМА, УРОВНИ, ПРОТОКОЛЫ Мисбахова А.Р. В сборнике: XXVII Всероссийский аспирантско-магистерский научный семинар, посвященный дню энергетика и 55-летию КГЭУ. Материалы докладов. Казань, 2023. С. 90-92. [Электронный ресурс]. URL: https://elibrary.ru/item.asp?id=64246772. (дата обращения: 30.04.2024.)
3. IMPLEMENTATION OF CONSENSUS WITH TCP/IP Chitrakar P., Ranjan P. В сборнике: Proceedings - 2021 IEEE 10th International Conference on Communication Systems and Network Technologies, CSNT 2021. 10. 2021. С. 105-112. [Электронный ресурс]. URL: https://ieeexplore.ieee.org/document/9509664. (дата обращения: 03.05.2024.)
4. Network Traffic Analysis (NTA) Tools [Электронный ресурс]. URL: https://www.trustradius.com/network-traffic-analysis-nta (дата обращения: 03.05.2024.)
5. ИССЛЕДОВАНИЕ DDOS-АТАК И МЕТОДОВ ИХ ОБНАРУЖЕНИЯ Лапина М.А., Вечканов А.В., Сысоев К.В., Швецов В.А. В сборнике: Студенческая наука для развития информационного общества. Материалы ХУ Всероссийской научно-технической конференции с приглашением зарубежных ученых. Ставрополь, 2024. С. 324-329. [Электронный ресурс]. URL: https://elibrary.ru/item.asp?id=59932081. (дата обращения: 03.05.2024.)
6. DDoS-guard: Классификация DDoS: полное руководство по типам атак [Электронный ресурс]. URL: https://ddos-guard.net/ru/blog/classification-of-ddos-attacks. (дата обращения: 10.05.2024.)
7. Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey. Mahmoud Abbasi., Amir Taherkordi. [Электронный ресурс]. URL: https://www.sciencedirect.com/science/article/pii/S0140366421000426. (дата обращения: 11.05.2024.)
8. HackTheBox. What is network traffic analysis. (2024 blue teamer guide) [Электронный ресурс]. URL: https://www.hackthebox.com/blog/network-traffic-analysis. (дата обращения: 11.05.2024.)
9. IBM. What is deep learning. [Электронный ресурс]. URL: https://www.ibm.com/topics/deep-learning (дата обращения: 11.05.2024.)
10. РАЗРАБОТКА СТЕНДА РЕАЛИЗАЦИИ ЭКСПЕРИМЕНТА ПО ЗАЩИТЕ ВИРТУАЛЬНЫХ МАШИН ОТ DDOS-АТАК Бугрова Е.С., Лоншаков М.А., Пестов И.Е., Савельева А.А. В сборнике: Подготовка профессиональных кадров в магистратуре для цифровой экономики (ПКМ-2023). Всероссийская научно-техническая и научно-методическая конференция магистрантов и их руководителей. Сборник лучших докладов: в 2-х томах. Санкт-Петербург, 2024. С. 245-249. [Электронный ресурс]. URL: https://elibrary.ru/item.asp?id=65698590. (дата обращения: 10.05.2024.)