МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования   
«Южный федеральный университет»

Институт математики, механики   
и компьютерных наук им. И. И. Воровича

Кафедра информатики и вычислительного эксперимента

Бейдина Дарья Алексеевна

МЕТОДЫ АНАЛИЗА ДАННЫХ В ЗАДАЧАХ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

КУРСОВАЯ РАБОТА   
по направлению подготовки  
**02.03.02**— Фундаментальная информатика и информационные технологии

Ростов-на-Дону – 2024

Оглавление

[1 Введение 3](#_Toc135910023)

2 Постановка задачи 4

3 Анализ характеристик сети и трафика 5

4 Анализ показателей, характерных для определенных видов атак 9

5 Тестирование методов обнаружения нарушений 11

5.1 Сбор и подготовка данных 11

5.2 Модели машинного обучения 15

5.3 Применение нейросетей 16

6 Результаты 18

6.1 Анализ эффективности моделей 18

[6.2 Анализ результатов работы нейросети 2](#_Toc135910040)0

[Заключение 25](#_Toc135910042)

[Список использованных ресурсов 2](#_Toc135910043)6

**1 Введение**

Информация в наше время обладает самой большой коммерческой ценностью для любой компании. В это понятие включается всё: технологии производства, наработанный опыт, состояние дел предприятия, данные о сотрудниках. Сложно переоценить важность сохранения в секрете данных коммерческих тайн от конкурентов. Именно поэтому, большинство компаний уделяют большое внимание информационной безопасности своих данных.

Однако даже в такой области большую опасность представляет пресловутый человеческий фактор – небрежность системных администраторов или целенаправленные вредоносные действия сотрудников могут сделать бесполезными все старания обезопасить свои данные от постороннего вмешательства.

Верным помощником компании будет система, анализирующая трафик корпоративной сети. Её задача распознать вредоносную активность и выявить предполагаемый источник угрозы. Ядром такой системы будет интеллектуальный анализ и классификация данных трафика.

**2 Постановка задачи**

**Цель:** реализовать и сравнить методы анализа трафика для обнаружения вредоносной активности и атак на сеть.

Для достижения этой цели необходимо выполнить следующие задачи:

1. Выявить ключевые характеристики сети, необходимые для анализа вредоносной активности.

2. Определить особенности и принципы определенных видов атак.

3. Выбрать подходящие наборы данных, содержащие информацию о различных видах вредоносной активности, и подготовить к использованию.

4. Произвести классификацию данных при помощи разных методов.

5. Проанализировать полученные результаты, проверить качество классификации и сравнить результаты работы различных методов.

**3 Анализ характеристик сети и трафика**

Классификация сетевого трафика — это процесс идентификации конкретных приложений или действий путем сопоставления их с сетевым трафиком. Управление этим процессом крайне необходимо для отслеживания состояния сети и обеспечения безопасности данных.

Большинство уже существующих методов анализируют либо работу портов, либо содержимое пакетов. Первый вариант в настоящее время не является надежным из-за широкого распространения динамичности и «запутывания» портов. Кроме того, многие приложения могут использовать одни и те же порты. Метод глубокого анализа пакетов обладает гораздо большей точностью, но здесь тоже есть сложности, связанные с зашифрованным трафиком. Однако развитие технологий позволяет проводить анализ поведения трафика, сопоставление сигнатур и использовать методы глубокого обучения для классификации в том числе и зашифрованного сетевого трафика.

Алгоритмы машинного обучения позволяют распознавать подозрительный сетевой трафик с помощью обучающих моделей по известным свойствам. Одними из самых распространенных алгоритмов, используемых для этой цели, являются деревья решений, метод опорных векторов и наивный байесовские классификатор (применение этих методов будет описано в главе 5). Они используют для классификации такие характеристики пакетов, как IP-адреса источника и назначения, длина пакета и временные метки. Точность действия таких алгоритмов существенно повышает предварительный отбор наиболее важных признаков.

Метод анализа поведения классифицирует трафик путем сопоставления его с поведенческими моделями. Это позволяет обнаружить аномальности и идентифицировать сетевые атаки по необычным показаниям. Например, можно заметить вредоносную активность по частым попыткам подключения или необычные объемам передачи данных с определенных IP-адресов. Методы анализа поведения используют некоторые шаблоны для сопоставления трафика. Сравнивая его с сигнатурой известной сетевой атаки можно определить, относится ли трафик к определенному типу атаки. Этот метод обычно используется в системах обнаружения вторжений и системах предотвращения вторжений.

Классификация трафика с глубоким обучением использует модели глубоких нейронных сетей для автоматической классификации сетевого трафика. Основанный на глубоком обучении метод классификации сетевого трафика на основе пакетов был предложен на Международной конференции по искусственному интеллекту в информации и коммуникации в 2019 году. В нем использовалась сверточная нейронная сеть для извлечения признаков из пакетов и многослойный персептрон для классификации трафика. Метод был применен к реальному набору данных и показал свою эффективность по сравнению с другими алгоритмами машинного обучения.

Метод анализа поведения может идентифицировать определенный неизвестный или зашифрованный трафик приложения, опираясь только на предварительные знания. Поэтому с его помощью сложно обеспечить точный результат. Последний метод не требует ручного извлечения характеристик трафика и может автоматически создавать сопоставление с определенными значениями, устраняя, таким образом, зависимость от предварительных знаний.

В последние годы существенно выросло количество трафика, использующего шифрование. Это происходит из-за возрастания ценности конфиденциальности пользовательских данных. Шифрование преобразует контент в случайный зашифрованный текст, затрудняя его расшифровку за короткий промежуток времени. В результате методы, основанные на анализе поведения, неэффективны для точного сопоставления и обнаружения.

Зашифрованный трафик улучшает безопасность сети, но он также создает возможности для вредоносных действий. Это может привести к увеличению числа ложных срабатываний или ложноотрицательных результатов при существующих методах проверки, что создает значительную проблему для системы в плане точной идентификации зашифрованного трафика и управления им по причине затрудненности проверки содержимого. Традиционные методы обнаружения вредоносного трафика на основе контента не смогут эффективно идентифицировать часть обычного зашифрованного трафика, что приведет к ложным срабатываниям. Например, облачные хранилища (такие как Google Drive, Dropbox и другие), позволяющие пользователям загружать, хранить файлы и делиться ими, во многом используют шифрование для обеспечения конфиденциальности и безопасности пользовательских данных. Однако такое поведение трафика может быть похоже на некоторые вредоносные действия (например, массовая передача файлов или файлы нестандартных типов), которые могут привести к ошибочным результатам классификации.

Кроме того, такие сервисы зачастую используют общие IP-адреса или доменные имена, которые могут быть связаны с другими вредоносными действиями. Когда системы кибербезопасности обнаруживают вредоносный трафик на основе IP-адресов или доменных имен, они могут неправильно классифицировать трафик этих сервисов как вредоносный.

Из проблем, с которыми сталкиваются сетевые приложения, видно, что все еще существуют некоторые сложные проблемы:

1. Неточная классификация трафика: из-за разнообразия прикладного программного обеспечения трафик может отличаться на разных устройствах и версиях одного и того же приложения. Кроме того, некоторые из них могут использовать шифрование для сокрытия своего трафика, что затрудняет его классификацию.

2. Неправильная классификация: обычно классификация происходит по таким признакам, как номера портов, IP-адреса и домены. Поэтому, результат может быть ошибочным. Например, сходство трафика приложений с разной функциональность легко может привести к неверной оценке.

3. Сложный алгоритм шифрования: некоторые алгоритмы могут также использовать случайные числа, хэш-функции и другие методы, что увеличивает сложность классификации.

4. Проблема с шумом: в зашифрованном трафике может быть много шума, что может повлиять на точность классификации.

5. Атаки и спуфинг: некоторые злоумышленники могут использовать шифрование для сокрытия своего атакуемого трафика. В то же время они могут подделывать другие типы трафика, чтобы обмануть систему классификации трафика.

6. Загруженность сети: пользователь может использовать несколько приложений одновременно, и их трафик будет смешиваться, что затруднит точную классификацию и запись трафика каждого приложения.

Для решения данных критических проблем и повышения точности можно рассмотреть следующие три условия:

1. Обновление правил обнаружения вредоносного трафика: правила системы сетевой безопасности обновляются, основываясь на моделях поведения. Это может помочь избежать ложных срабатываний, когда обычный трафик ошибочно идентифицируется как вредоносный.

2. Классификация на основе признаков: вредоносный трафик можно идентифицировать, рассматривая множество признаков, таких как поведение трафика, схемы передачи данных, доменные имена и IP-адреса. Рассматривая несколько признаков вместе, становится возможным более точно различать обычный законный трафик и опасный для системы трафик.

3. Использование машинного обучения или методов глубокого обучения: изучение особенностей и закономерностей обычного трафика для обнаружения вредоносного, как аномального.

**4 Анализ показателей, характерных для определенных видов атак**

Анализ ключевых показателей для наиболее распространенных видов атак является важным этапом при решении задач информационной безопасности. Для этого необходимо рассмотреть различные признаки вторжений в сеть и их особенности, чтобы выявить соответствующие показатели трафика.

Один из наиболее распространенных видов атак – это DDoS-атака (Distributed Denial of Service), направленная на перегрузку сервера запросами до тех пор, пока он не станет недоступным для пользователей. Типичными признаками наличия такой атаки являются большое количество пакетов с одинаковым IP-адресом отправителя, высокая скорость передачи данных и отсутствие ответа от получателя.

Другой распространенный вид атаки – это сканирование портов (Port Scanning). Это процесс проверки доступности портов на удаленном компьютере для определения уязвимостей системы. При такой атаке в показателях трафика наиболее часто встречаются большое количество запросов к различным портам, использование специальных программ для сканирования портов и попытки получения доступа к защищенным ресурсам.

Также стоит упомянуть о фишинге (Fishing), который представляет собой попытку получить конфиденциальную информацию путем обмана пользователей. Характеристики трафика при такой атаке могут включать отправку электронных писем со ссылками на поддельные сайты или запросы на предоставление личной информации.

SQL-инъекции (SQL Injection), в основном, критичны для сайтов корпоративной сети. Это атака, направленная на базы данных, когда злоумышленник пытается внедрить вредоносный код через веб-формы или URL. SQL позволяет управлять базами данных. Поэтому этот тип атак является крайне опасным, так как он работает напрямую с информацией, хранящейся на сервере. Соответственно, типичные признаки такой атаки это странные запросы к базе данных и попытки получить доступ к защищенной информации.

Перехват сессии (Session Hijacking) – это атака, при которой злоумышленник получает доступ к активной сессии пользователя и использует ее для выполнения действий от его имени. Вследствие этого, часто меняется характер трафика от данного устройства: использование других приложений, запросы к нетипичным сайтам, скорость работы. Часто при такой атаке может встречаться использование украденного cookie-файла или токенов аутентификации.

Одной из самых легко распознаваемых атак является попытка взлома паролей (Password Cracking), когда злоумышленник пытается подобрать пароль пользователя для получения доступа к его аккаунту. Обнаружить её можно по частым запросом от одного и того же устройства к одному и тому же URL-адресу, то есть многократные попытки входа с использованием различных комбинаций символов.

Часто злоумышленники не атакуют сеть напрямую, а используют вредоносное ПО (Malware), которое может нанести вред компьютеру или сети, устанавливая его на одно или несколько устройств сети. В таком случае среди трафика можно обнаружить установку скрытых процессов, часто из подозрительных источников, и передачу конфиденциальной информации на удаленный сервер.

Все эти примеры показывают, что анализ показателей характеристик трафика является ключевым элементом при решении задач информационной безопасности. Он позволяет выявлять различные виды атак и принимать меры по защите от них.

**5 Тестирование методов обнаружения нарушений**

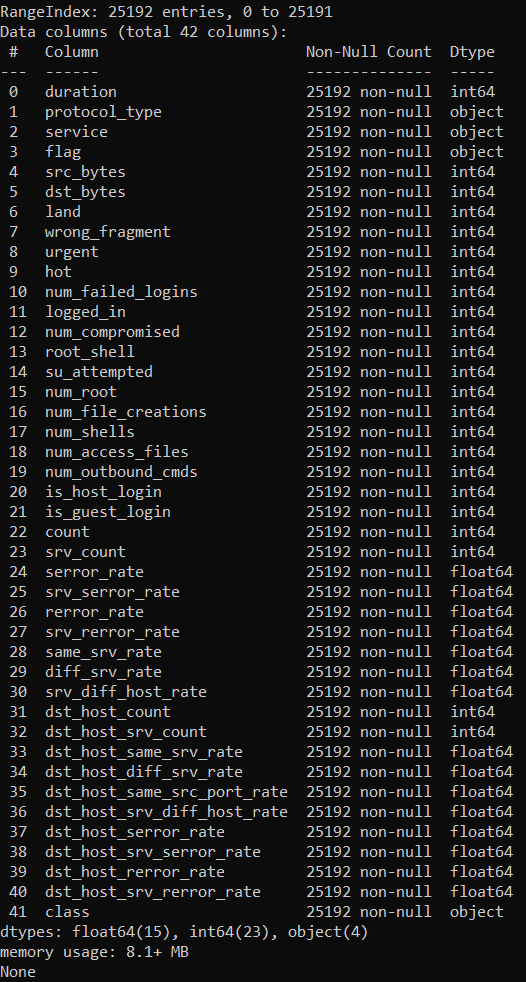
Описав анализируемые характеристики трафика и наиболее типичные признаки различных видов атак, переходим к самому интересному – реализации этих знаний на практике.

**5.1 Сбор и подготовка данных**

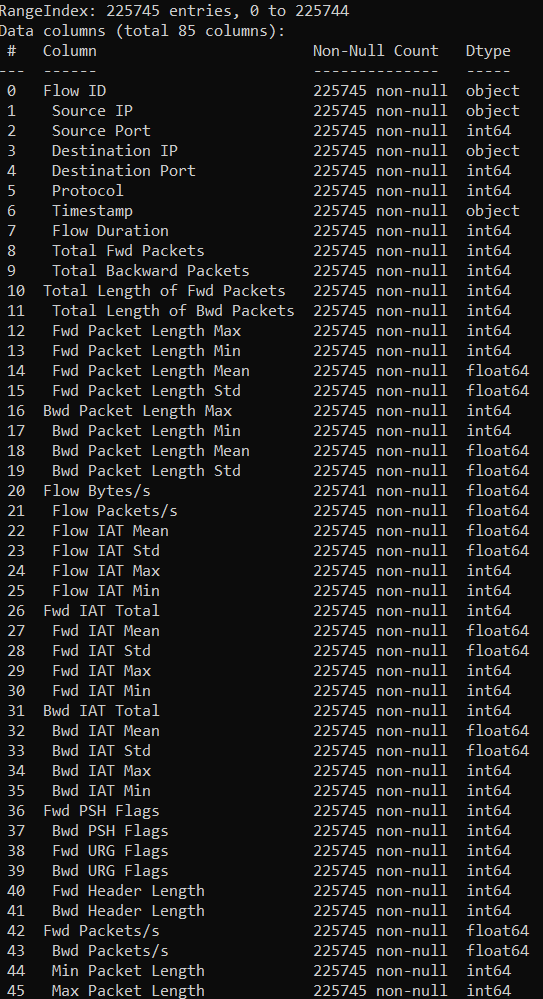
Для обучения моделей и нейросети необходимо большое количество различных, но при этом хорошо сбалансированных данных. Так как их малый объем может привести к переобучению – ситуации, когда модель хорошо работает на обучаемых данных, но не умеет грамотно обобщать результаты на новых, раннее ей не встречавшихся данных. Также датасет должен быть хорошо сбалансирован, так как при наличии большого количества выбросов (резко отличающихся по значениям показателей, нетипичным случаям) модель может больше соориентироваться на них при обучении, вследствие чего снизится точность классификации реальных данных.

Для исследования будут использованы два датасета. Первый из них Network Intrusion Detection был получен в смоделированной сети, имитировавшей типичную локальную сеть ВВС США. Каждая запись содержит 41 характеристику некого соединения, по которому отправляется цепочка TCP-пакетов, включая длительность подключения, а также метку нормального или аномального соединения. Далее используем датасет, собранный Канадским институтом кибербезопасности с использованием CICFlowMeter-V3. Это часть сгенерированных ими данных, пакеты поступают с нескольких устройств со смесью DDoS-атак и безопасных данных, всего более 225 000 строк. Среди заголовков столбцов содержатся похожие характеристики данных такие, как тип используемого протокола, размеры пакетов и метки класса трафика.

Для программирования будем использовать язык Python и библиотеки pandas, numpy, scikit-learn, keras для работы с данными и matplotlib для отображения результатов работы. Перед использованием нужно убедиться, что датасеты готовы к работе – нужно проверить отсутствие нулевых значений и преобразовать категориальные характеристики. Для этого выводим информацию об этих датасетах.



Выписка о датасете Network Intrusion Detection



Выписка о датасете DDoS атак

В первом датасете нет пропущенных значений. Поэтому мы лишь заменяем категориальные характеристики. В датасете о DDoS-атаках есть пропущенные значения в одном столбце и слишком большие значения для типа float64 в некоторых пакетах, а также лишние столбцы (Flow ID, Timestamp, Source IP, Destination IP), которые мы удаляем. Также убираем записи с чрезмерно высокими и нулевыми значениями.

**5.2 Модели машинного обучения**

Первым методом, на котором мы проверим свои гипотезы, будет машинное обучение. Мы протестируем 5 разных моделей, постепенно перемещаясь от простых до сложных. Всё это мы сделаем на основе библиотеки scikit-learn языка программирования python.

Начнем с наиболее простой для понимания – классификатора по методу ближайших соседей. Так как в этом методе нельзя задать количество кластеров в результате, то следует установить минимальное количество соседей по кластеру. Этому параметру задано значение 9000, так как 43% от исходного количества записей принадлежат одному кластеру, остальные – другому. А 9000 соответствуют примерно 35-40% от общего количества в обоих датасетах.

Второй моделью будет классификатор по дереву решений. По сути, эта модель выводит ряд правил принятия решений о принадлежности каждого случая, оформляя каждое правило в узел, а их последовательность в дерево. Здесь, как и далее значение параметра random\_state устанавливается для более детерминированного результата.

Далее мы используем метод опорных векторов, так как эта модель очень эффективна для больших датасетов с большим количеством параметров. Этот метод позволяет сэкономить память благодаря работе с подмножеством опорных векторов – случаев, наиболее близких к записям противоположного кластера. То есть обучение сводится к проведению некой линии в пространстве всех записей, которая разделяет их на 2 группы. Из параметров установим ядру значение “rbf”, так как в наших датасетах достаточно много параметров, имеющих достаточно сложную структуру зависимостей друг от друга, вдобавок ядро радиально-базисных функций хорошо подойдет по причине отсутствия настроек параметров данных, а также зададим проведение перекрестной проверки для увеличения точности.

И последней моделью будет лес случайных деревьев. Это также коллективный метод, состоящий из множества деревьев решений. Смысл в том, что к каждое из них строится не на всем наборе параметров, а на случайно подобранном наборе из них. Интересно, что такой подход во многом устраняет опасность переобучения благодаря устойчивости к шуму и выбросам в данных и принципу голосования по большинству.

Такой подбор моделей обоснован их распространенностью и достаточно высокой точностью. Выбор леса случайных деревьев в качестве ансамблевого метода обусловлен хорошим соотношением между временем обучения и точностью, чего не хватает, например, классификатору голосования, который на данных наборах обучался бы намного дольше леса, но по точности был бы сопоставим с ним.

**5.3 Применение нейросетей**

Также к тем же двум датасетам мы применим две разных нейросети: простую и более сложную. Здесь будет использована библиотека tensorflow.

Начнем с более простой нейросети. В ней будет 3 слоя: 2 полносвязных и между ними dropout-слой, уменьшающий обучающий датасет, для предотвращения переобучения. Первый полносвязный слой состоит из 64 нейронов, в качестве функции активации выбрана relu, так как с ней обучение проходит достаточно быстро, вследствие хороших показателей сходимости, и является хорошим аппроксиматором для таких больших наборов данных. Также relu может дезактивировать часть нейронов, если в них нет необходимости. В результирующем слое имеется 1 нейрон, а функцией активации выбрана сигмоида. Она имеет существенный недостаток – при чрезвычайно малом значении градиента (приближении активационной функции к кривой предсказания, условно разделяющей классы объектов) нейросеть “отказывается обучаться”, из-за чего увеличивается время работы. По этой причине сигмоида не используется для первого слоя.

Вторая нейросеть является рекуррентной. Первым и третьим слоем являются LSTM (long short-term memory) c 64 нейронами каждый, однако между ними есть некоторая разница: первый возвращает только последний вывод из выходной последовательности, а второй – полностью весь выведенный результат. Вторым и четвертым являются dropout-слои. А последний является полносвязным с аналогичными последнему слою описанной выше нейросети параметрами: 1 нейроном и сигмоидной функцией активатором. Обычно рекуррентные сети используются для сложных заданий распознавания текста и других медиа-объектов, однако из-за хороших показателей точности отлично показывают себя и в задачах классификации данных. Но из-за их намного более долгого обучения в некритичных системах больше распространены полносвязные нейросети.

Обе сети используют оптимизатор адам (стохастический градиентный спуск) и учатся в течение 10 шагов. Выбор оптимизатора обусловлен желанием максимально уменьшить функцию потерь, чему способствует вычисление на одном случайно выбранном примере градиента (вектора частных производных) на каждом шаге, благодаря чему процесс обучения идет быстрее. Количество шагов выбрано небольшое вследствие большого размера датасетов.

**6 Результаты**

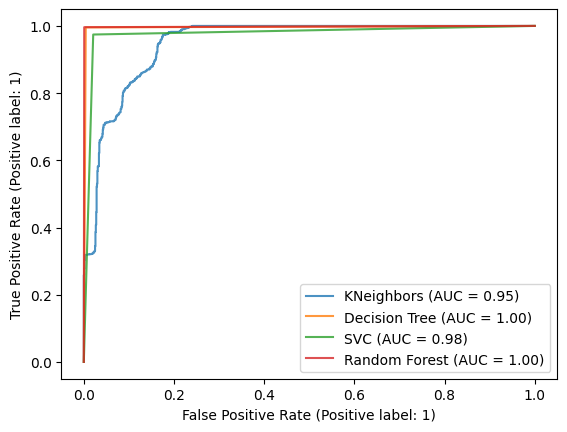
Проанализируем результаты обучения при помощи различных показателей. Для моделей выведем графики roc-кривых, метрики истинно поставленных меток (accuracy), истинно положительных из всех положительных (precision) и всего набора (recall). Для сетей построим графики изменения метрики истинно поставленных меток и функции потерь. Перекрестные таблицы выведем и для моделей, и для сетей.

**6.1 Анализ эффективности моделей**

Мы посмотрим результаты последовательно: сперва для датасета просто отделяющего вредоносный трафик от неопасного, затем для данных о распознавании DDoS-атак.

|  |  |  |
| --- | --- | --- |
| KNeighbors  (метод ближайших соседей) |  |  |
| Decision Tree  (дерево решений) |  |  |
| SVC  (метод опорных векторов) |  |  |
| Random Forest  (лес случайных деревьев) |  |  |

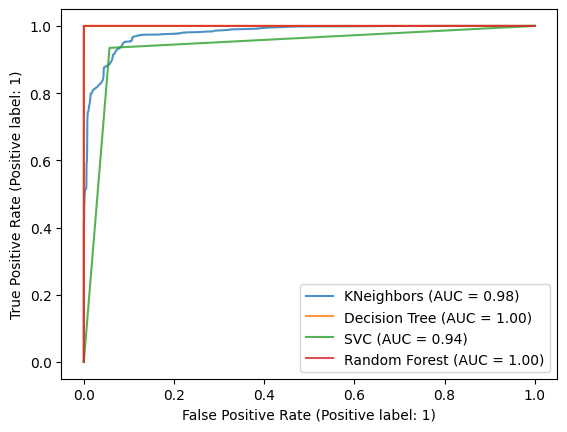
Распознавание вредоносного трафика



Сравнение кривых ошибок для первого датасета

|  |  |  |
| --- | --- | --- |
| KNeighbors  (метод ближайших соседей) |  |  |
| Decision Tree  (дерево решений) |  |  |
| SVC  (метод опорных векторов) |  |  |
| Random Forest  (лес случайных деревьев) |  |  |

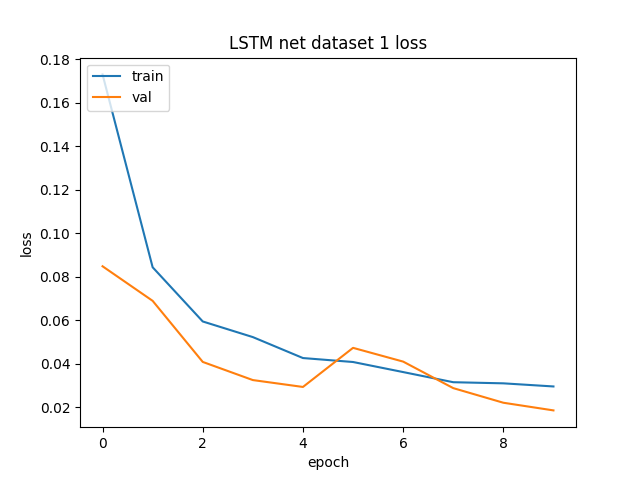
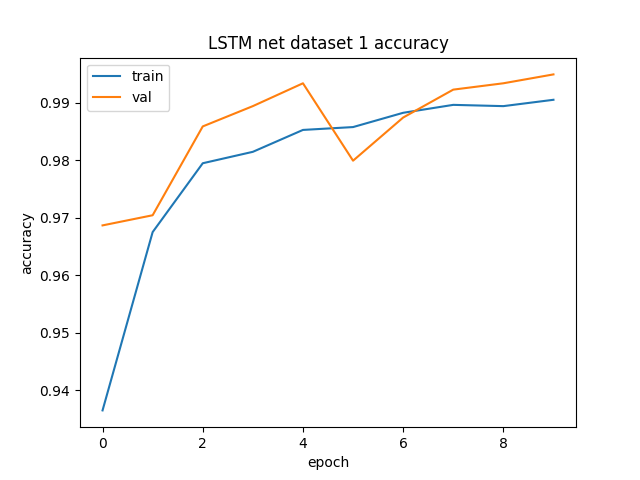
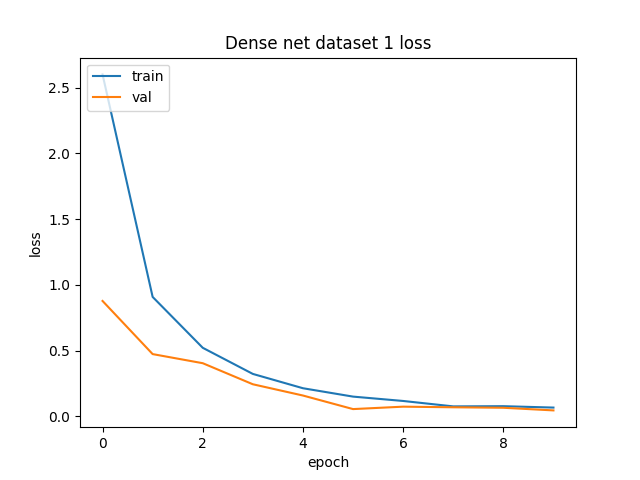
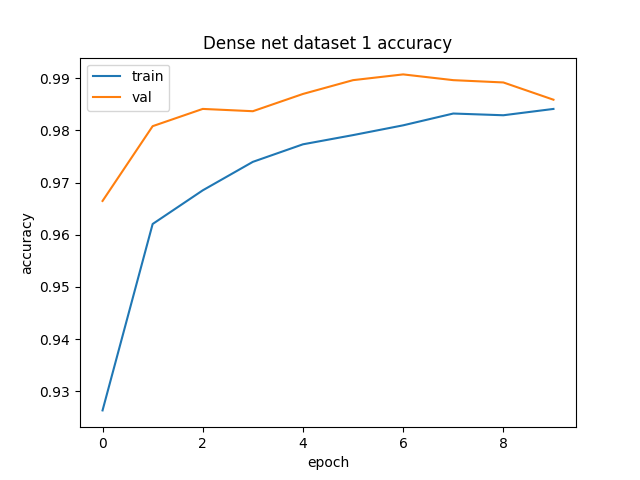
Распознавание DDoS-атак



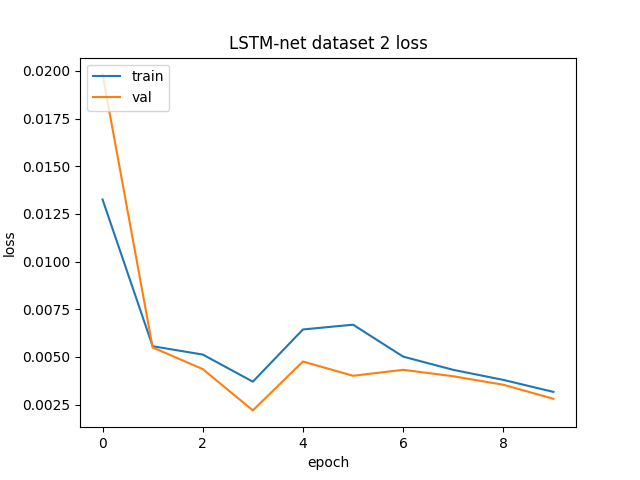
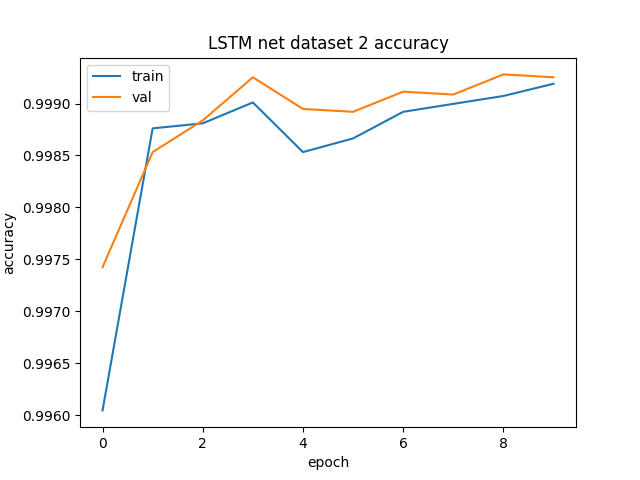
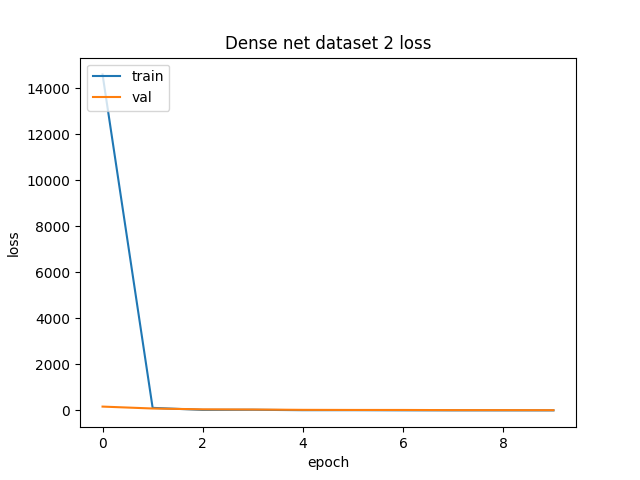
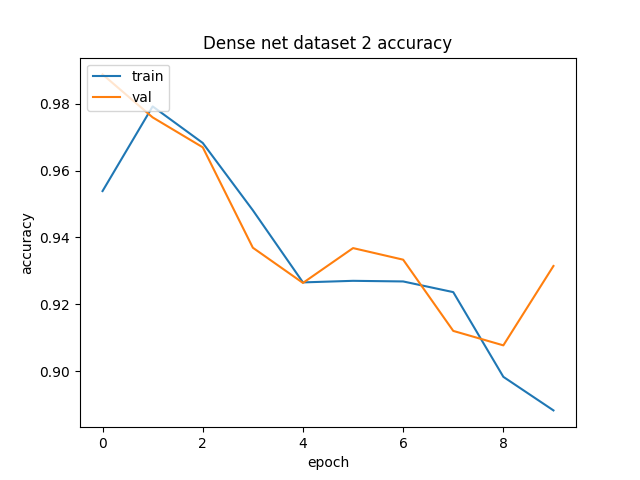
Сравнение кривых ошибок для второго датасета

**6.2 Анализ результатов работы нейросети**

Сначала посмотрим результаты работы полносвязной и LSTM-сети на первом датасете, где нашей целью было отделить вредоносный трафик от безопасного.



Теперь, посмотрим на результаты распознавания DDoS-атак.



|  |  |  |
| --- | --- | --- |
|  | Dense net | LSTM net |
| Распознавание вредоносного трафика |  |  |
| Распознавание DDoS-атак |  |  |

**Заключение**

В решении данной задачи бесспорно среди моделей наилучший результат дал лес случайных деревьев, а LSTM оказалась эффективнее полносвязной сети. Однако не все так однозначно.

Среди моделей отлично себя показал метод построения дерева решений: учится быстрее всех, а по точности уступает только лесу. Так происходит во многом благодаря большому количеству сбалансированных данных. Метод ближайших соседей по той же причине дал хороший результат, но ему понадобилась подсказка, которую не всегда можно дать, – заданное мною минимальное количество точек для выделения отдельного кластера. Метод опорных векторов хорошо обработал первый датасет, но второй плохо по причине переобучения и, возможно, из-за неверно выбранного ядра – линейное дало бы неплохой результат, если судить по высокой точности модели дерева решений.

Ожидаемо рекуррентная сеть показала себя лучше полносвязной. Но здесь есть вполне естественное объяснение – у первой больше слоев обработки. Но у рекуррентной сети есть недостаток в том, что она сравнительно долго обучается, а по метрике ложно отрицательных объектов сильно уступает модели не только леса, но и дерева решений.

На основании проведенных тестов можно увидеть, что нейросеть не всегда оказывается лучше моделей машинного обучения. Но в нашем примере немаловажную роль сыграл большой и хорошо сбалансированный датасет, что является критичным именно для моделей. В противном случае показатели нейросети были бы лучше машинного обучения.

**Список использованных ресурсов**

* + - 1. В. А. Нестеренко. Конспект лекций по курсу «МЕТОДЫ ОБНАРУЖЕНИЯ НАРУШЕНИЙ В СЕТИ»

1. <https://www.tensorflow.org/>
2. <https://www.codecamp.ru/blog/interpret-roc-curve/?ysclid=lwxme5mhpn950157686>
3. <https://scikit-learn.ru/>
4. <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.crosstab.html>
5. <https://neurohive.io/ru/osnovy-data-science/activation-functions/>
6. <https://www.kaggle.com/datasets/>
7. <https://numpy.org/doc/stable/reference/arrays.scalars.html>
8. https://seaborn.pydata.org/generated/seaborn.heatmap.html