

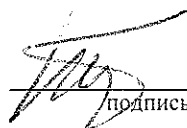
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой №52

д.т.н., профессор

должность, уч. степень, звание



14.06.2022

подпись, дата

А.М. Тюрликов

инициалы, фамилия

БАКАЛАВРСКАЯ РАБОТА

на тему Обнаружение аномального трафика «Интернета вещей» методами машинного
обучения

выполнена Сверликовым Александром Владимировичем

фамилия, имя, отчество студента в творительном падеже

по направлению подготовки

11.03.02

код

Инфокоммуникационные технологии

наименование направления

и системы связи

наименование направления

направленности

03

код

Программно-защищенные


наименование направленности

инфокоммуникации

наименование направленности

Студент группы №

5823



14.06.2022

подпись, дата

А.В. Сверликов

инициалы, фамилия

Руководитель

доктор тех. наук, профессор

должность, уч. степень, звание



14.06.2022

подпись, дата

Т.М. Татарникова

инициалы, фамилия

В работе не содержится информации с ограниченным до
ходом и отсутствием сведений, представляющих интерес
и ценность

Санкт-Петербург 2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

УТВЕРЖДАЮ

Заведующий кафедрой №52

д.т.н., профессор

должность, уч. степень, звание



подпись, дата

А.М. Тюрликов

инициалы, фамилия

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ БАКАЛАВРСКОЙ РАБОТЫ

студенту группы

5823

номер

Сверликов Александр Владимирович

фамилия, имя, отчество

на тему Обнаружение аномального трафика «Интернета вещей» методами машинного
обучения

утвержденную приказом ГУАП от 31.03.2022 № 11-352/22

Цель работы: Дополнение технологий анализа трафика на основе методов машинного
обучения

Задачи, подлежащие решению: 1. Определить основные понятия трафика
2. Изучить методы анализа интернет трафика методами машинного обучения
3. Реализация машинного обучения для анализа трафика

Содержание работы (основные разделы): 1. Понятие трафика
2. Определение критериев выявления аномалий в сетевом трафике
3. Применение методов машинного обучения для анализа трафика
4. Практика

Срок сдачи работы « 02 » июня 2022 г.

Руководитель

доктор тех. наук, профессор

должность, уч. степень, звание



подпись, дата

Т.М. Татарникова

инициалы, фамилия

Задание принял(а) к исполнению

студент группы №



подпись, дата

А.В. Сверликов

инициалы, фамилия

Оглавление

Введение	4
1 Основные понятия.....	6
1.1 Понятие трафика	6
1.2 Анализа трафика.....	8
1.3 Методы сбора трафика по требованиям к памяти	9
1.4 Методы сбора трафика по архитектуре системы.....	11
1.5 Методика анализа трафика	15
2 Определение критериев выявления аномалий в сетевом трафике	19
2.1 Исследование стандартной модели поведения IoT устройств.....	19
2.2 Структура сети IoT	23
2.3 Архитектура IDS интернета вещей	24
3 Применение методов машинного обучения для анализа трафика	27
3.1 Способы и методы машинного обучения	27
3.2 Методы машинного обучения: нейронная сеть	29
3.3 Методы машинного обучения: случайный лес	32
3.4 Методы машинного обучения: кластеризация	34
4 Практика	36
4.1 Описание обучающей выборки	36
4.2 Формирование признакового пространства.....	38
4.3 Оценка обучения машины	41
4.4 Результаты моделирования.....	42
Выводы	43
Источники.....	44

Введение

Развитие технологий построения измерительных систем на основе интернета вещей (Internet of Things, IoT) является трендом современных инфокоммуникационных сетей.

Устройства IoT, как правило используют батареи в качестве источника питания, обладают низкой вычислительной мощностью, ограниченным объемом памяти [1]. Эти ограничения обусловили разработку протоколов, обеспечивающих передачу и обработку данных с минимальной вычислительной и коммуникационной нагрузкой, что делает устройства интернета вещей уязвимыми к аномальному трафику – различным атакам, реализация которых может нанести серьезный ущерб эксплуатируемому оборудованию IoT и даже физический ущерб людям [2].

Известно, что системы обнаружения вторжений (Intrusion Detection System, IDS) не увеличивают нагрузку на сенсорные устройства и поэтому могут быть использованы для своевременного обнаружения аномального трафика, генерируемого интернетом вещей [3].

IDS для IoT представляет собой программное обеспечение, которое позволяет выявить несанкционированное получение доступа или вредоносную атаку на данные и оповестить о нарушении безопасности. Методы, используемые в системах обнаружения вторжения, относятся к активным и реализуют [4]:

- поиск по сигнатуре;
- поиск по аномалии.

Поиск по сигнатуре – способ, позволяющий определить атаку по некоторому шаблону, составленному по признакам известных на данное время атак. При совпадении признаков трафика и одного из шаблонов создается оповещение о нарушении безопасности. Такой метод позволяет определять только те вторжения, которые были ранее.

Поиск по аномалии – способ, при котором система сравнивает активность трафика с некоторой моделью стандартного (корректного)

поведения того или иного устройства. При отклонении от нормы система также оповестит о нарушении безопасности.

Исходя из скорости развития информационных технологий в целом и отрасли интернета вещей в частности, есть проблема быстрого устаревания систем обнаружения вторжения. IDS нуждаются в постоянных доработках и обновлениях, чтобы корректно реагировать на появление новых угроз, что в свою очередь требует больших временных и финансовых затрат.

В работе предлагается интеграция метода машинного обучения на основе самообучаемой нейронной сети в анализ трафика, генерируемого интернетом вещей. Самообучаемая нейросеть способна самостоятельно адаптироваться к обновляемой среде и оставаться актуальной на протяжении длительного времени.

Таким образом, цель работы: дополнение технологий анализа трафика на основе методов машинного обучения. Для реализации поставленной цели необходимо решить следующие задачи (оглавление для создания 4х задач: понятия трафика, определение критериев выявления аномалий в сетевом трафике, применение методов машинного обучения для анализа трафика, практика)

1 Основные понятия

1.1 Понятие трафика

Слово «трафик» обозначает движение (англ. traffic — движение). Под сетевым же трафиком подразумевается вся информация, передаваемая через участок сети за промежуток времени. Объём трафика измеряется в битах, байтах (и из производных таких как килобайт и мегабайт). Так же существует измерение трафика в пакетах – специально сформированных блоках данных.

Больше всего трафика исходит от непосредственных переходов на сайты через браузеры. Следом по количеству идут системы поиска и ссылки. Самым малым объёмом отличается трафик переходов на сайты из мессенджеров (приложение для обмена мгновенными сообщениями) и рекламы.

Плотность трафика зависит от страны, её населения и общего благосостояния. Крупные страны, такие как Российская Федерация, США или Великобритания, являются основными генераторами трафика в сети Интернет.

Зная цифры входящего и исходящего трафика можно анализировать ценность той или иной информации. То, насколько она актуальна и нужна обществу. Таким образом, например, можно отслеживать эффективность рекламы, общие интересы пользователей и т.д.

Все устанавливаемые связи между источниками в Интернете взаимозависимы. Источник и пользователь при подключении непрерывно обмениваются данными – это порождает входящий и исходящий трафик. Объём трафика при подключении будет зависеть от устройства, времени и действий во время подключения.

Самыми экономными в использовании сети являются мобильные устройства. Они портативны и их алгоритмы с протоколами работы направлены на то, чтобы максимально оптимизировать процесс передачи трафика. Это сделано для увеличения их срока автономности.

Длительное подключение ожидаемо приведёт к увеличению трафика. Так же трафик пассивного подключения будет отличаться от трафика, когда устройства обменивались большими файлами с данными. Всё это, как и расчёт общего числа пользователей в конкретный момент времени, требуется для расчета показателей использования данного ресурса. Помимо моментного интервала могут рассматриваться более длинные временные отрезки, такие как день, месяц, год и т.д. Это будет зависеть от того, для чего именно собирается информация.

Например, для определения времени подключения к источнику какого-то конкретного пользователя может хватить небольшого временного интервала в сутки или неделю, но для сбора статистики посещения сайта интернет-магазина такого короткого промежутка будет недостаточно. Сбор статистики требует гораздо больший временной период.

Как сказано выше, при подключении пользователя к источнику, создаётся обратная связь, поэтому весь трафик принято разделять на исходящий и входящий:

- входящий трафик – данные, полученные устройством во время подключения;
- исходящий трафик – данные, отправленные с устройства во время подключения.

Зная примерный объём потоков входящих и исходящих данных, можно отследить деятельность некоторых вредоносных программ (вирусов). При этом исходящий трафик устройства может сильно отличаться от его «нормального» поведения, так как вирус будет передавать данные устройства в сеть вместе с обычным трафиком. Внутри самой сети будет крайне трудно отследить эти данные, поскольку в современном мире при развитии скорости передачи найти данные почти невыполнимая задача.

Однако всё ещё остаётся возможность определить устройство и время передачи. Скорость передачи данных мобильного устройства в сети Интернет заметно медленнее, чем трафик с персонального компьютера, а

время отправления и получения зачастую прописаны в самих пакетах. Даже если информация о времени намеренно изменена, то по общему состоянию сети можно определить примерный временной промежуток, когда была совершена передача. Например, днём трафик более плотный, чем ночью.

Обычно в только что купленных устройствах уже существуют встроенные утилиты для отслеживания входящего и исходящего трафика. Кроме поиска вредоносных данных, эта функция необходима для мониторинга оставшегося трафика в подключённом тарифе. Более продвинутые программы отслеживания трафика способны самостоятельно выявлять возможные угрозы безопасности устройства. Они помогают предотвратить нежелательные соединения и утечку данных.

1.2 Анализа трафика

Анализ трафика подразумевает под собой исследование сети. Обычно это делается при помощи специальных утилит, позволяющих получить полный доступ к интернет адаптеру. Сканировать можно трафик любого уровня семиуровневой модели ISO (Рисунок 1), что зависит лишь от типа проверки и возможностей, которая предоставляет нам программа-анализатор.



Рисунок 1 – Семиуровневая модель OSI

Используя данные, которые получены в процессе анализа, можно сделать выводы о необходимости проведения тех или иных действий. Затем, на основе принятых решений, проводятся работы по улучшению безопасности, маркетингового продвижения или общей оптимизации

ресурса. Например, на рисунке (Рисунок 2) изображён общие показатели трафика для трёх популярных поисковых систем – Яндекс, Google и Mail. Проанализировав его, можно сказать, что в период с декабря 2017 года по июнь 2018 года наблюдается рост использования ресурса Яндекс. Следовательно, необходимо оптимизировать его для работы с большим количеством пользователей, улучшить безопасность и т.д.

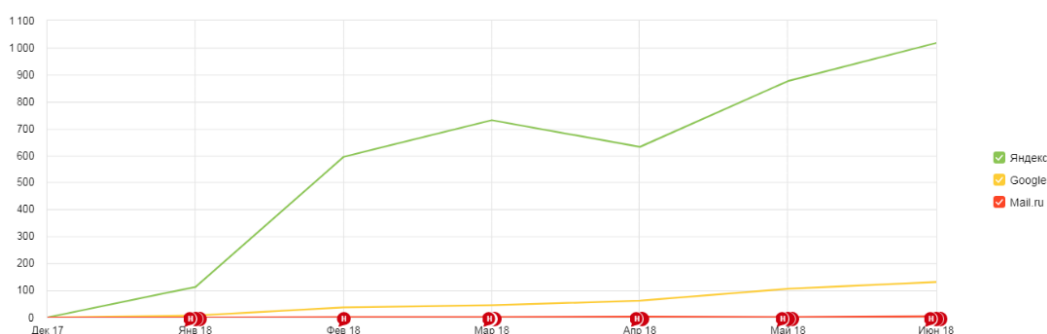


Рисунок 2 – График показателя трафика сайтов Яндекс, Google и Mail.ru

1.3 Методы сбора трафика по требованиям к памяти

Существуют три основных метода сбора трафика, разделённые по требованиям к объёму памяти:

- сбор всех пакетов данных;
- сбор сетевого потока;
- сбор расширенного потока.

Основная задача сбора всех пакетов является копирование всего сетевого трафика, генерируемого устройствами в сети, и хранение копий пакетов вместе с их заголовками и информацией. Т.е. все данные собираются для последующего анализа на каком-либо стороннем носителе. Эти сведения содержат наиболее полную информацию о трафике, которая может понадобиться при его исследовании. Благодаря сохранению полной информации о заголовках пакетов и их передаваемой информации, такой метод мониторинга может быть наиболее универсальным, так как большой объём информации может интенсивно храниться и обрабатываться [5].

Сетевой поток – это множество IP-пакетов, которые проходят через выбранную точку анализа трафика за определённый временной период.

Смысл сбора по сетевому потоку заключается в том, что IP-пакеты, принадлежащие к одному потоку, имеют общие свойства. Узнать эти свойства и требования к потокам можно в документах RFC 3917: Требования к IPFIX (IP Flow Information Export) [6].

Исходя из определения сетевого потока, он представляет собой поток сетевых пакетов, которые отвечают следующим условиям:

- единовременны (появляются в один и тот же период времени);
- родственны (имеют одинаковые адрес и порт);

Основным свойством потока является то, что если пренебречь некоторой информацией в пакетах и полей заголовков пакетов, а также объединить некоторые пакеты, то уменьшится объем данных, что приводит к уменьшению требуемой памяти для хранения потоков. Однако это затрудняет анализ трафика [7].

При сборе расширенного потока собирается информация о всех пакетах и сетевых потоках. Информация потока добавляется к информации из заголовков пакетов или из передаваемой в пакетах информации. Вместе с тем расширенный поток также может содержать дополнительную информацию о каком-то внешнем источнике, например, о географическом расположении IP-адресов источника и назначения. Поэтому некоторые решения сбора расширенного потока рассматривают эту информацию как метаданные [8].

Большая часть современных исследований в области сбора сетевого трафика посвящена вопросам сбора пакетов в скоростных сетях с минимальной потерей данных и сжатием данных после сбора, то есть снижению объемов. Например, в работах [8, 9] авторы соответственно обсуждают вопросы преобразования данных для их эффективного хранения и обработки и мониторинга в облаке (Cloud). В работах [10, 11] авторы предлагают подход к разработке приложений по сбору данных в скоростных сетях, основанный на стандартных аппаратных средствах. А в работе [12] для полного анализа сетевого трафика авторы предлагают метод агрегации потоков.

Однако на сегодняшний день существуют лишь небольшое количество общеизвестных технологий подобных подходов, обеспечивающих объединение линий связи на уровне канального протокола передачи данных, и сравнительно недавно разработанные технологии, реализующие объединение на физическом уровне протокола.

При объединении на канальном уровне, растёт количество служебных пакетов, что приводит к уменьшению коэффициента использования канала. Так же растёт произвольная задержка в передаче данных, из-за которой возникают проблемы с качеством операторской связи.

1.4 Методы сбора трафика по архитектуре системы

По различиям в архитектуре сети способы сбора трафика можно разделить на следующие методы:

- Host Based;
- Network Based;
- Flow Based.

При использовании Host Based (Рисунок 3) сбор трафика происходит через конкретные хосты (в данном случае хостами являются L3-устройства). Для сбора используются межсетевые экраны, маршрутизаторы или обычные стационарные/портативные компьютеры пользователя. Если используются межсетевые экраны или роутеры, то можно воспользоваться встроенными утилитами сбора трафика (например, tcpdump).

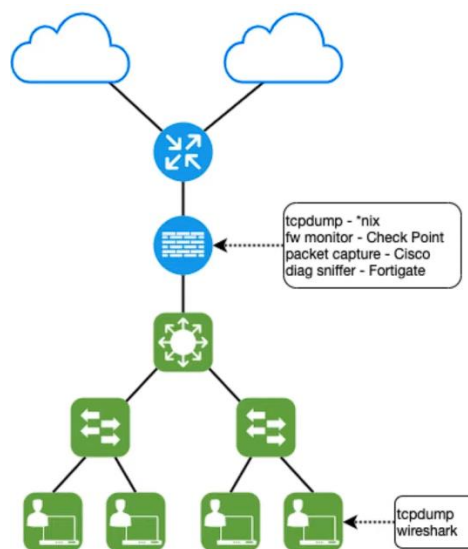


Рисунок 3 – Архитектура метода сбора трафика Host Based

Это метод позволяет быстро начать работу с анализом трафика, так как в большинстве случаев не требует установки каких-либо дополнительных устройств и утилит. Многие программы уже встроены по умолчанию, и они не требуют специфичных настроек. Если же анализ трафика происходит на компьютере, то присутствует возможность работы с полноценным графическим интерфейсом. Бюджет метода минимален, так как существует огромное количество бесплатных утилит сбора.

Данный метод применяется гораздо чаще остальных, поскольку позволяет быстро настроить сканирование трафика для обнаружения простых, а вследствие, самых часто возникаемых проблем.

При работе с большим объёмом данных, данный метод может не подойти. Все данные проходят через одно устройство, и при больших нагрузках всё сильнее будет заметна низкая производительность этого способа. Помимо этого, есть вероятность «уронить» устройство. Т.е. из-за большой нагрузки приходящий трафик будет игнорироваться до тех пор, пока устройство не проработает уже вошедшие данные.

Так же, метод Host Based не является всесторонним. Возможность анализа трафика есть только на одном, выбранном устройстве. Этого может оказаться недостаточно для полномасштабного поиска проблем и ошибок в проверяемой сети.

Метод Network Based (Рисунок 4) является более серьёзным способом анализа сети. Сбор трафика производится с L2-устройств или физических линков. Иными словами, здесь идёт работа на канальном уровне. Нет IP-адресов, но отправитель и получатель идентифицируются по MAC-адресам. Сам трафик разбит на «фреймы» (кадры).

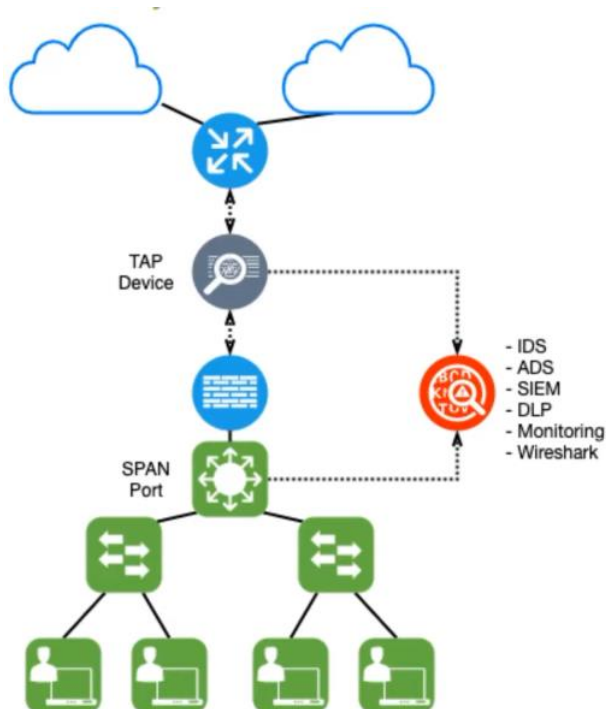


Рисунок 4 – Архитектура метода сбора трафика Network Based

Процесс сбора трафика производится при помощи SPAN-портов и TAP-устройств. У коммутатора ядра, через который проходит трафик, настраивается SPAN-порт (зеркальный) на его конкретном интерфейсе. Здесь не важно, физический интерфейс или логический. Через него производится пересылка копии трафика на конкретный порт, где располагается прослушивающее устройство. Этим устройством может быть IDS, ADS (Anomaly Detection System), различные системы мониторинга и т.д.

Теперь пропускная способность анализируемого трафика упирается лишь в пропускную способность настроенного порта. Сама нагрузка по сбору трафика переходит на стороннее устройство, что позволяет проводить анализ сети без нарушения её работы. Так же, при настройке зеркального порта не требует изменения топологии сети.

Если устройство не поддерживает SPAN-соединение, или отсутствуют свободные порты, то используются TAP-устройства. Они подключаются к месту разрыва кабеля сети. По сути, таким устройством может стать любой управляемый коммутатор с функцией SPAN-порта. Будет изменена топология сети, но проведённых для этого работ будет минимально.

Проблемы такого подхода заключается в том, что обычно приходится работать с огромным количеством копированного трафика, так как метод с зеркальными портами в основном применяется на главных интернет магистралях. Помимо этого, существуют серьёзные пробелы при использовании данного метода, так как далеко не весь трафик идёт через главные магистрали.

Решить эти проблемы призван метод Flow Based (Рисунок 5). Он основан на специальных протоколах, которые позволяют собирать с сетевого устройства телеметрию трафика. Иными словами, телеметрия трафика это информация об IP-адреса источника и получателя, номера портов, протоколов, TCP-флагов и т.д. Всё это составляет заголовок пакета без его содержимого.

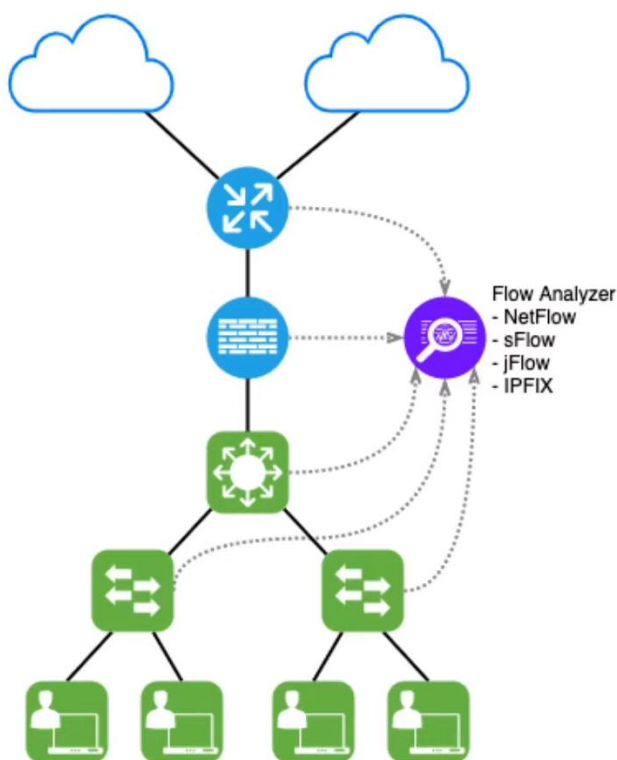


Рисунок 5 – Архитектура метода сбора трафика Flow Based

Вся телеметрия загружается в flow-коллектор (Flow Analyzer), который позволяет проводить различную аналитику, строить отчёты по трафику и помогать в поиске каких-либо проблем сети. Собирать информацию при таком подходе можно с фаерволов, роутеров, коммутаторов и обычных компьютеров.

Таким образом возможно сканировать всю сеть сразу, вплоть до трафика между портами одного коммутатора (при обмене информации двух компьютеров в одной сети). Собираемая информация минимальна по объёму, так как производится сбор не всего трафика, а лишь его телеметрии. Отсюда следует, что для обработки полученных данных требуются минимальные ресурсы.

Уменьшение количества информации ведёт так же к ухудшению качества анализа. Снижается точность, так как телеметрия собирается не у каждого проходящего пакета (сэмплирование). Из всего потока берётся каждый сотый или тысячный пакет. Так же далеко не всё оборудование поддерживает flow-протоколы.

1.5 Методика анализа трафика

Модель трафика представляет собой закономерность изменения его основных характеристик во времени. В описании трафика обычно учитываются три характеристики [7]:

- тренд – описание поведения трафика во времени;
- сезонность – закономерность роста или падения объема трафика, связанных, например, с конкретным промежутком времени;
- случайная составляющая – остальные характеристики, оставшиеся после исключения основных характеристик, собственно, в которых и следует искать аномалии.

После выбора способа анализа, необходимо трафик разложить на составляющие:

- выделить тренд и сгладить исходные данные, например, при помощи скользящего окна, экспоненциального сглаживания или регрессии;
- вычесть или разделить сезонную составляющую из исходных данных, что зависит от выбранного способа;
- после удаления сезонного фактора и тренда остается случайная составляющая, которая и принимается за аномалию.

Примерами аномалий могут быть выбросы, сдвиги, изменения распределения значений, отклонения от среднего и совместные аномалии.

В качестве примера можно рассмотреть анализ трафика, основанный на циклическом алгоритме, состоящий из:

- отбора данных;
- сглаживания тренда;
- поиска возможных циклов;
- удаления трендовых компонент;
- проверки циклов;
- предсказания будущих циклов.

На этапе отбора данных определяется анализируемый временной ряд X_q (Рисунок 6). На рисунке изображен временной ряд, где по оси ординат отложен объем трафика V , а по оси абсцисс – период наблюдения T . Для дальнейшего анализа, временная шкала дискретизируется равными интервалами Δt наблюдения динамики изменения объема трафика.

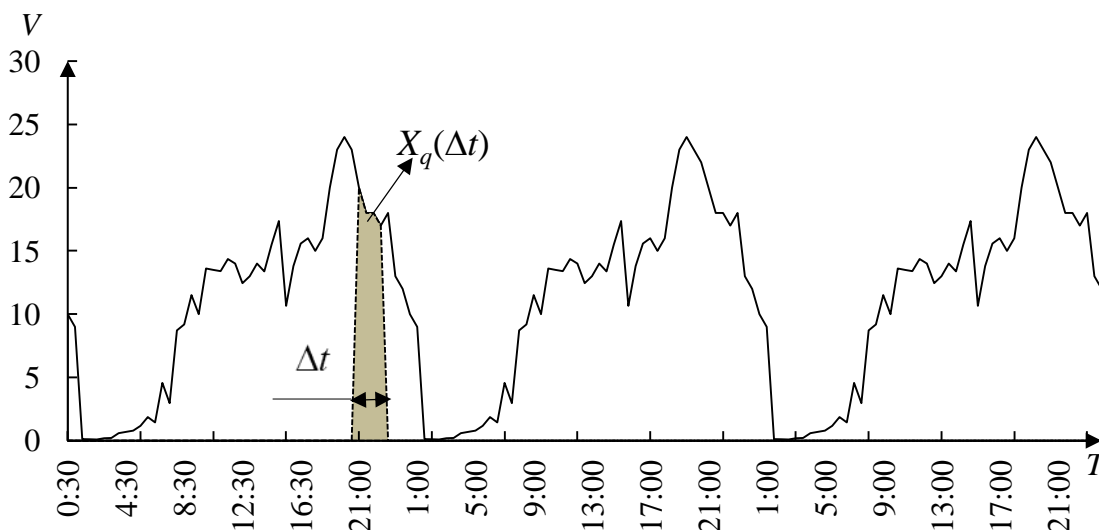


Рисунок 6 – Временной ряд трафика

Каждый интервал Δt содержит агрегацию данных, объем которых зафиксирован в этом интервале

$$X_q(\Delta t) = \sum_{j=1}^R V_j$$

где q – номер интервала ($q = 1, 2, \dots, Q$);

R – число пакетов в интервале Δt ;

X_q – ряд данных.

Для сглаживания необходимо исключить одиночные выбросы – $(L - 1)$ точку. Применяя, например, метод краткосрочной центрированной скользящей средней осуществляется переход к новому сглаженному ряду X_k , длина которого равна $N = Q - (L - 1)$, $k = \overline{1, N}$.

$$X_k = \frac{1}{L} \sum_{j=k}^{k+L-1} X_j.$$

После удаления случайных колебаний и выравнивания значений переходят к поиску циклов. Здесь применяется метод спектрального анализа. На графике спектра мощности (Рисунок 7) видны пики, образующиеся возле определенных частот. Пики указывают на возможные циклы.

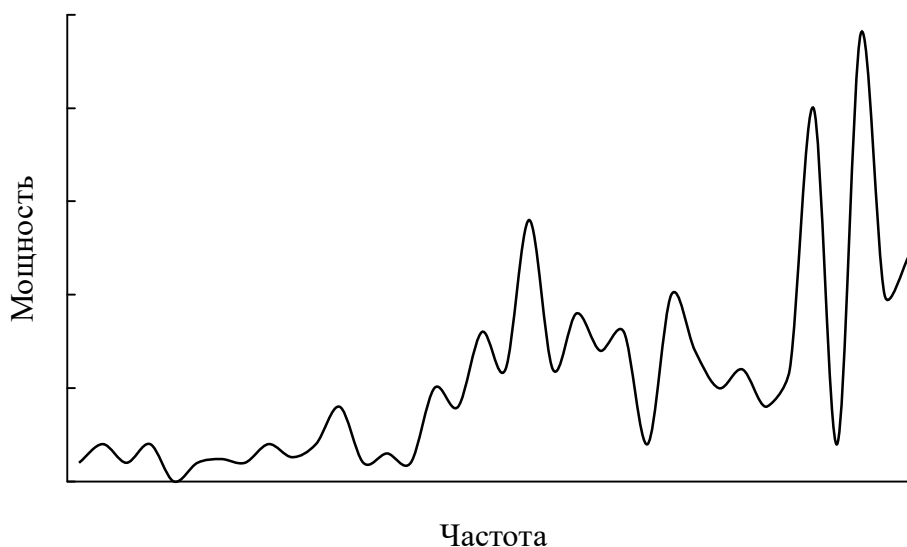


Рисунок 7 – Спектр мощности

Пики свидетельствует только о вероятности наличия циклов. Поэтому проводят еще несколько шагов для проверки наличия циклов. Одним из таких критериев является удаление трендовых компонентов. Применяя метод скользящей средней, удаляются силы роста в данных, а затем исходя из статистической значимости методами F -коэффициентов или хи-квадрат ищется отклонение от распределения спектра мощности.

2 Определение критериев выявления аномалий в сетевом трафике

2.1 Исследование стандартной модели поведения IoT устройств

Исходя из того, что для рядового пользователя подключение «умного» устройства означает внедрение IoT прибора в домашнюю Wi-Fi сеть, будет рассмотрено именно такое беспроводное подключение. Такой прибор будет потреблять небольшой объём энергии и, скорее всего, тот объём трафика, с которым оперирует устройство, не сможет привести к какому-нибудь негативному эффекту. Однако стоит учитывать, что в перспективе интернет вещей подразумевает объединение всех подобных устройств в одну единую сеть, что приведёт к увеличению потенциального ущерба, нанесённого вредоносным влиянием на трафик. Например, перегрузка устройств данными и, как следствие, их выход из строя.

Чтобы в дальнейшем различать аномальный трафик в потоке данных, необходимо выяснить, как данные ведут себя при обычной работе устройств интернета вещей. Для этого будут использованы имитационные модели, основанные на IP-сетях, и использующие сенсорные датчики. Датчики необходимы для проверки передачи данных, которые они будут считывать и отправлять на обработку.

Таким образом, можно определить не только нормальное поведение потока данных, но и выявить его стандартные структуры для различных устройств. Такой подход считается гибким, так как сама модель исследования подразумевает модульность схемы[1].

За основу модели была взята плата с wi-fi модулем, а именно ESP8266. Выбор обусловлен тем, что при помощи подобных плат можно относительно легко самостоятельно собрать аналог IoT устройства. К тому же, такие платы широко распространены. Для считывания объёма потока данных было выбрано два способа[2]:

- анализ при помощи DeviceHive;
- анализ при помощи Blynk.

DeviceHive — платформа для работы с устройствами интернет вещей с открытым исходным кодом. Программа позволяет соединять устройства между собой, контролировать их работу, обеспечивать безопасность соединения и проводить диагностику IoT решений.

В основе DeviceHive лежит работа с микросервисами: единое приложение строится как набор небольших сервисов, каждый из которых работает в собственном процессе и коммуницирует с остальными используя легковесные механизмы, как правило HTTP.

Эти сервисы построены вокруг одной цели и развертываются независимо с использованием полностью автоматизированной среды. Существует абсолютный минимум централизованного управления этими сервисами. Сами по себе эти сервисы могут быть написаны на разных языках и использовать разные технологии хранения данных. Это упрощает поддержку базы данных или шины сообщений и допускает переключение между ними без изменений в коде, что позволяет вводить дополнительные функциональные элементы, а конечный пользователь может настроить платформу и расширять ее функционал.

Blynk – платформа представляет собой облачный сервис для создания графических пультов управления. Подходит для широкого спектра микрокомпьютеров и микроконтроллеров. Программа обладает готовыми решениями для реализации интерфейса управления устройством интернет вещей. Blynk может работать в локальном режиме, в таком случае подключение к интернету не нужно.

Данные программы имеют открытый исходный код, что позволяет провести необходимые модификации, если появится такая необходимость. Так как DeviceHive и Blynk отличаются между собой методами работы с потоком данных, это поможет гораздо точнее определить стандартную модель поведения трафика.

Для более полного объёма исследований устройство будет рассмотрено в трёх состояниях:

- фоновое – устройство включено, но не ведёт активную передачу данных и не получает команд;
- получение команд – устройство получает команды для выполнения;
- отправка данных – устройство передаёт данные.

Ниже (Рисунок 8) изображён график фоновой работы устройства на платформе DeviceHive.

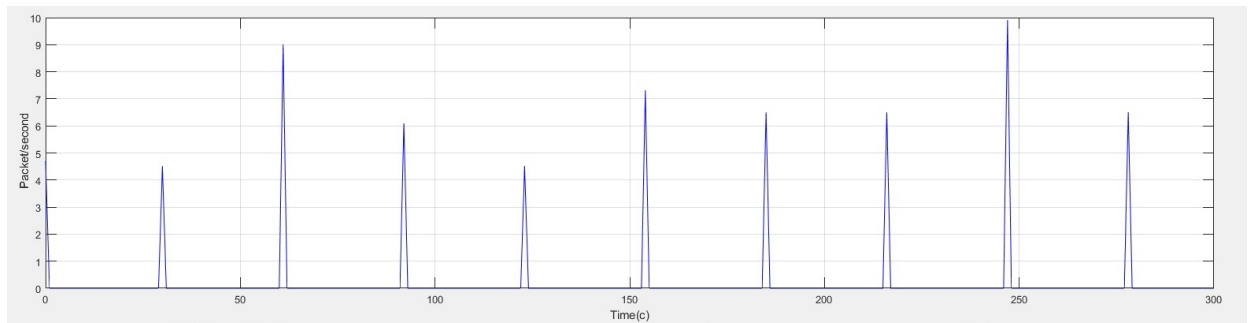
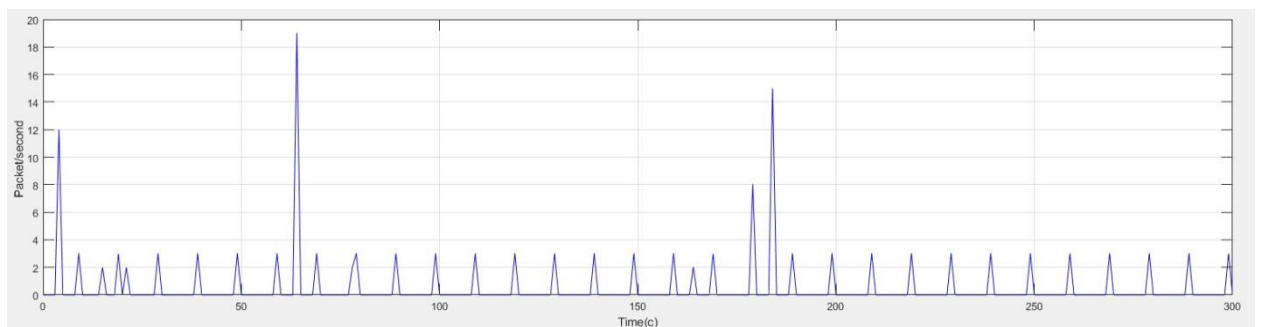


Рисунок 8 – Фоновая работа устройства (DeviceHive)

Пакеты, передаваемые устройством в фоновом режиме, являются служебными. Они передаются в равные промежутки времени и цикличны. Промежуток отправки зависит от устройства и его настроек.

Ниже (Рисунок 9) изображён график фоновой работы устройства на платформе Blynk.



Платформа Blynk работает с более частой отправкой пакетов, но сами пакеты при этом меньше, чем у DeviceHive.

Ниже (Рисунок 10) изображён график получения команды на платформе DeviceHive.

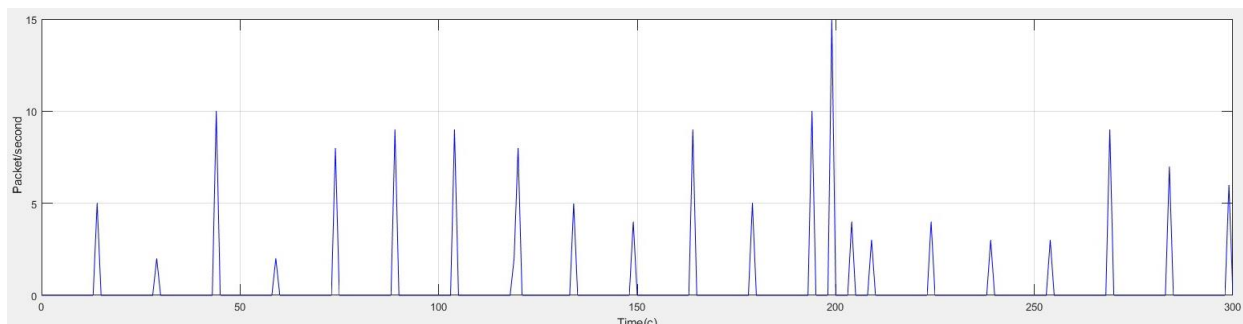


Рисунок 10 – Получение команды устройством (DeviceHive)

На графике видно, что передача пакетов происходит гораздо чаще, так как помимо стандартных пакетов-отчётов в трафике присутствуют пакеты с командами для устройства.

Аналогичная ситуация с использованием Blynk (Рисунок 11).

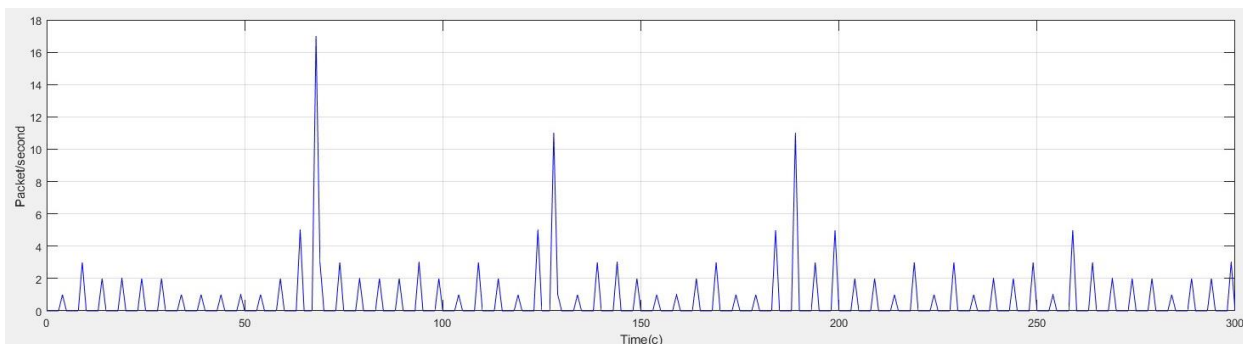


Рисунок 11 – Получение команды устройством (Blynk)

На следующих изображениях (Рисунок 12-13) изображены графики передачи данных на платформе DeviceHive и Blynk соответственно.

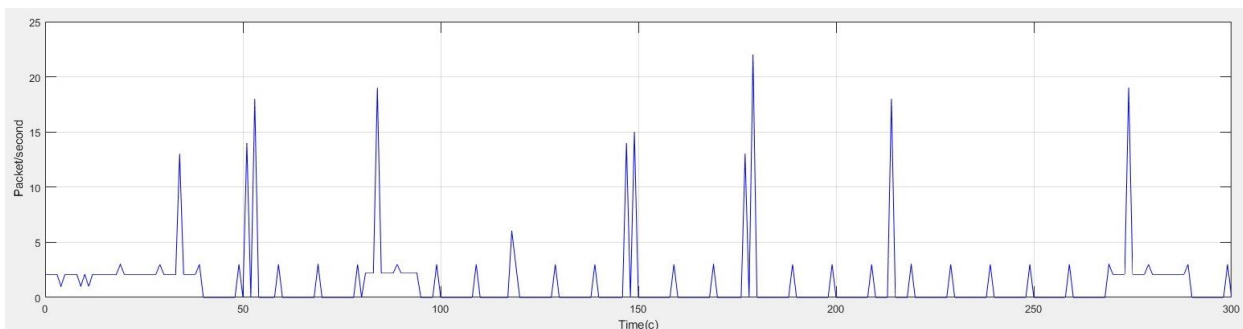


Рисунок 12 – Передача данных устройством (DeviceHive)

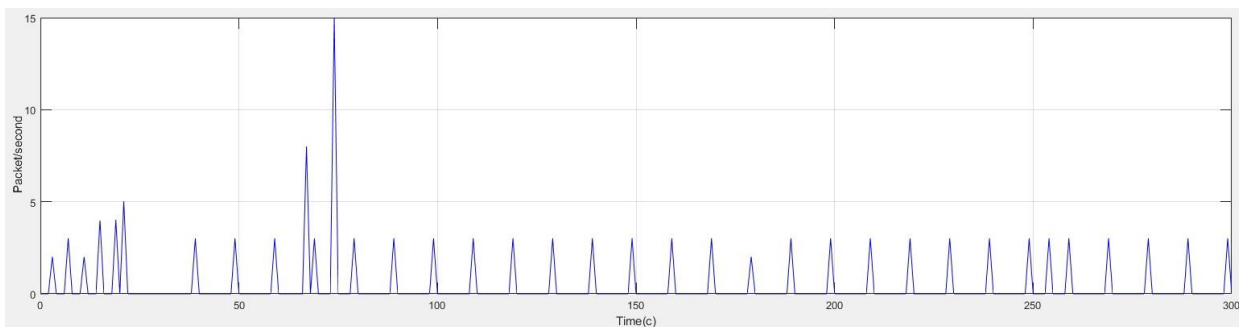


Рисунок 13 – Передача данных устройством (Blynk)

После проведения исследований трафика интернет вещей в стандартных состояниях можно переходить к выявлению аномальных составляющих.

2.2 Структура сети IoT

Взаимодействие Machine-to-Machine (M2M) является основой интернета вещей и подключения новых устройств в единую сеть [5]. Организация M2M взаимодействия основано на протоколе CoAP (Constrained Application Protocol). Этот протокол разработан специально для маломощных устройств, совместим с HTTP (HyperText Transfer Protocol) и реализует передачу данных с минимальными энергозатратами такими, как они есть – без модификаций и инкапсулирования [6].

Протокол CoAP является бинарным и работает поверх UDP (User Datagram Protocol – протокол пользовательских датаграмм), что позволяет работать с любым типом данных. Передача данных происходит эффективнее, так как бинарный код подразумевает короткие и маловесные пакеты – уменьшается объем, требуемый для передачи данных и появляется гибкость при работе с различными устройствами.

В общем случае, взаимодействие клиента с умным устройством интернета вещей выглядит, как на рис. 1, где подразумевается, что пакеты данных с сенсорного устройства проходят через прокси-сервер, где инкапсулируются в HTTP-пакеты.

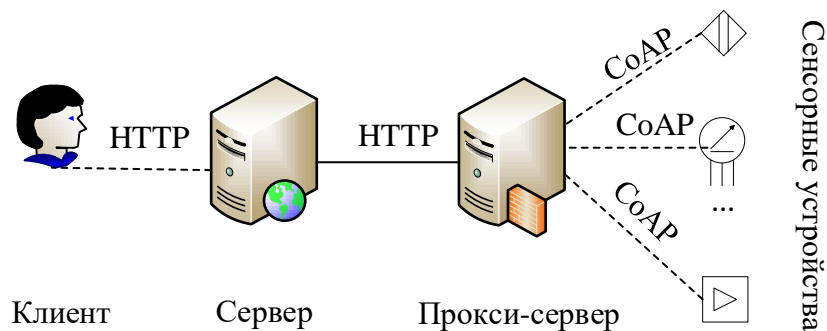


Рисунок 14 – Схема взаимодействия пользователя с сенсорным устройством

Протокол CoAP не поддерживает режим шифрования, предусмотренный в протоколе TCP (Transmission Control Protocol – протокол управления передачей), а поддерживает более упрощенный вариант шифрования DTLS (Datagram Transport Layer Security – протокол датаграмм безопасности транспортного уровня).

Протокол DTLS может работать в одном из четырех режимов:

- NoSec – в этом режиме шифрование отключено;
- PreSharedKey – шифрование включено, добавляется список общих ключей шифрования AES (Advanced Encryption Standard – симметричный алгоритм блочного шифрования) и узлов, между которыми ведется конфиденциальный обмен данными;
- RawPublicKey – шифрование включено, используются асимметричные пары ключей с шифрованием по алгоритму AES;

Сертификат – шифрование включено, устройство использует сертификаты X.509 с цифровыми подписями для распределения открытых ключей.

2.3 Архитектура IDS интернета вещей

IDS для интернета вещей, имеет иерархическую структуру, как и сама сеть IoT – три компонентных уровня (Рисунок 15).

На нижнем уровне иерархии находятся кластерные беспроводные сенсорные сети [8]. Головной узел кластера отвечает за авторизацию других

членов кластера, маршрутизатор отвечает за авторизацию головных узлов кластера, а шлюз за авторизацию маршрутизаторов.

Модуль обнаружения аномального поведения установлен в каждом кластере, интеллектуальная система обнаружения сетевых атак на маршрутизаторах и шлюзах.

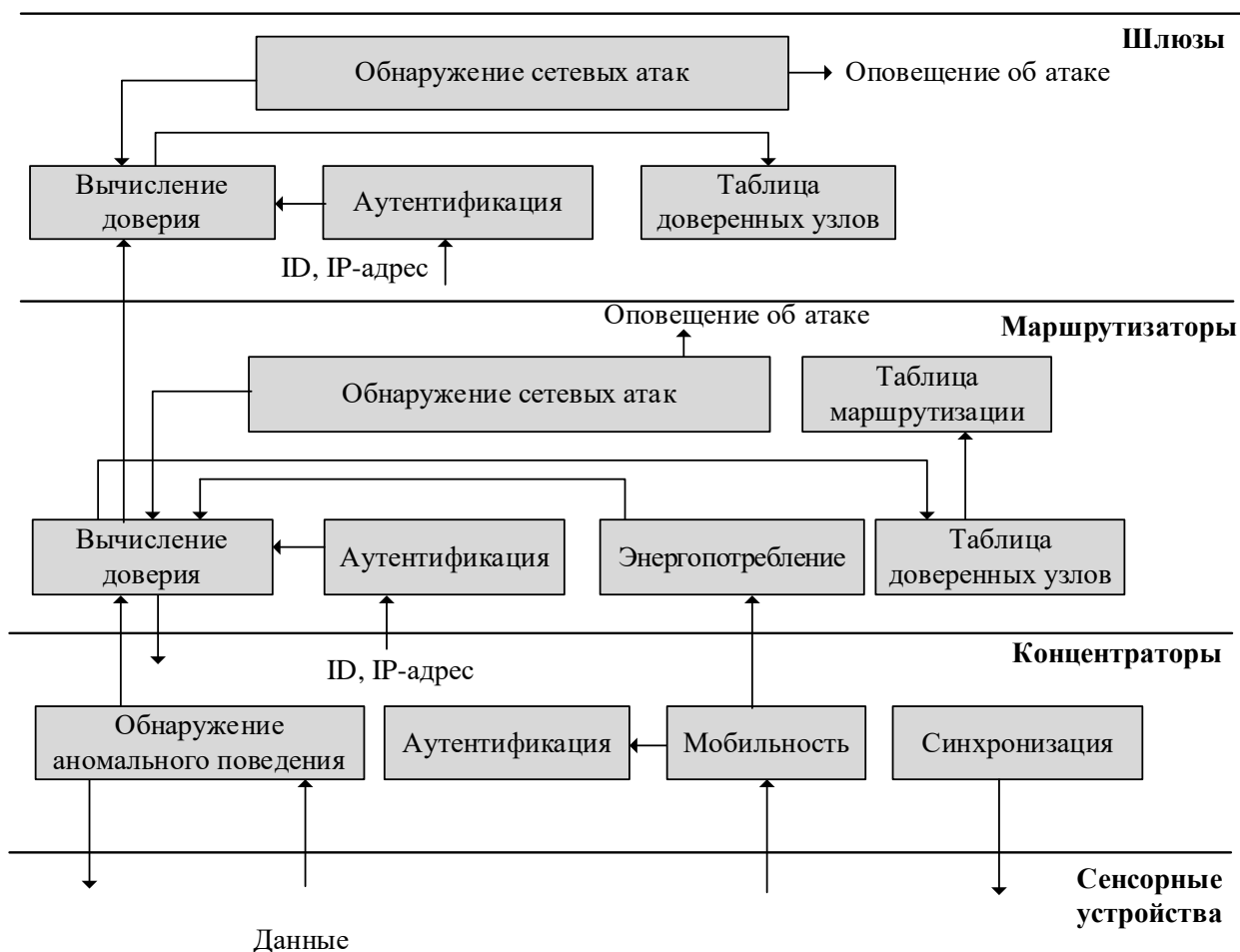


Рисунок 15 – Компоненты IDS интернета вещей

IoT-устройства оповещают о своем присутствии в конкретном кластере, направляя идентификатор (ID), например MAC-адрес в модуль мобильности. Модуль мобильности отвечает за регистрацию вновь прибывших в кластер устройств и удаление вышедших из кластера устройств. Любой вновь прибывший узел проходит процедуру аутентификации [9].

Модуль синхронизации представляет собой тактовый генератор – формирование временных окон (слотов) и соответственно раундов

беспроводной сенсорной сети, во время которых происходит опрос IoT-устройств.

Модуль обнаружения аномального поведения реализуется программно по приведенной в статье методике анализа трафика.

На уровне сети обнаружение атак реализуется программной моделью глубокого обучения, например нейронной сетью или случайным лесом [20].

Модуль вычисления доверия позволяет в онлайн режиме удалять скомпрометированные узлы из списка доверенных узлов, на основании которого формируется таблица маршрутизации.

3 Применение методов машинного обучения для анализа трафика

3.1 Способы и методы машинного обучения

Существует несколько способов машинного обучения:

- обучение с учителем;
- обучение без учителя;
- обучение с частичным привлечением учителя;
- обучение с подкреплением;
- глубинное обучение.

Обучение с учителем (Supervised learning) – данный способ применяется, если заранее известно, чему необходимо обучить машину. Компьютер анализирует крупные обучающие выборки до тех пор, пока не получит на выходе требуемые результаты. Проверка обучения производится путём контрольного прогноза для тестовых данных, которые ещё не известны машине.

Иными словами, алгоритмом обучения с учителем – это любой алгоритм машинного обучения, где алгоритму выдаётся то, как ему необходимо вести себя с точки зрения разработчика.

В основном обучение с учителем применяется для задач, связанных с классификацией и прогнозированием.

Обучение без учителя (Unsupervised learning) – машина без учителя ищет в выборке коррелирующие данные и строит зависимости между разными переменными. Способ применяется для группировки данных в кластер по их статистическим свойствам.

Отличием от предыдущего алгоритма является то, что обучение без учителя не руководствуется явными указаниями, как поступить в той или иной ситуации. Существует только общая оценка его конечного результата работы.

Обучение с частичным привлечением учителя (Semi-Supervised learning) – это гибрид двух предыдущих методов. При работе с данными,

учитель частично указывает машине, какие решения необходимо принять для достижения нужного результата.

Обучение с подкреплением (Reinforcement learning) – метод подразумевает систему «вознаграждения» машины за достижение правильного результата. Если автоматизировать процесс выдачи вознаграждения, то обучение машины будет проходить в автономном режиме.

Глубинное обучение (Deep learning) – метод может использовать как обучение без учителя, так и обучение с подкреплением. В данном методе часто симулируется реальное человеческое обучение, поэтому часто прибегают к аналогу человеческого мозга – нейронной сети. Нейронные сети позволяют более подробно уточнять характеристики наборов данных.

Помимо способов обучения, существуют методы построения архитектуры обучающейся машины:

- нейронные сети;
- дерево решений;
- случайный лес;
- кластеризация.

Нейронные сети приближены к человеческому мозгу как в представлении, так и в работе: существуют искусственные нейроны, которые связаны между собой. Нейронные сети многослойны. Нейрон одного слоя получает извне и передают их нейронам другого слоя. В итоге, когда данные достигнут выходного слоя, нейронная сеть выдаст решение задачи.

Дерево решений работает с объектами на основе принятия решений в узловых точках. От принятого решения будет зависеть ветвь, по которой данные передадутся в другой узел, до тех пор, пока данные не дойдут до конечного ответа, называемого «листом» древа.

Случайный лес — механизм быстрого обучения для поиска связей в наборе данных. Если для древа решений необходимо длительное обучение для получения точных ответов, то алгоритм случайного леса (random forest)

использует «комитет» таких решающих деревьев, созданных случайным образом. «Голосованием» древ выбирается самый популярный вариант ответа, который подаётся на выход.

Кластеризация группирует элементы данных по сходным характеристикам при помощи статистических алгоритмов. Это метод обучения без учителя, который можно использовать для решения задач классификации.

3.2 Методы машинного обучения: нейронная сеть

Нейронные сети появились за счёт стремления учёных воссоздать модель работы человеческого мозга в программном исполнении. На рисунке (Рисунок 18) изображена схожесть устройства биологического нейрона с его математическим представлением. Такое представление используется в машинном обучении.

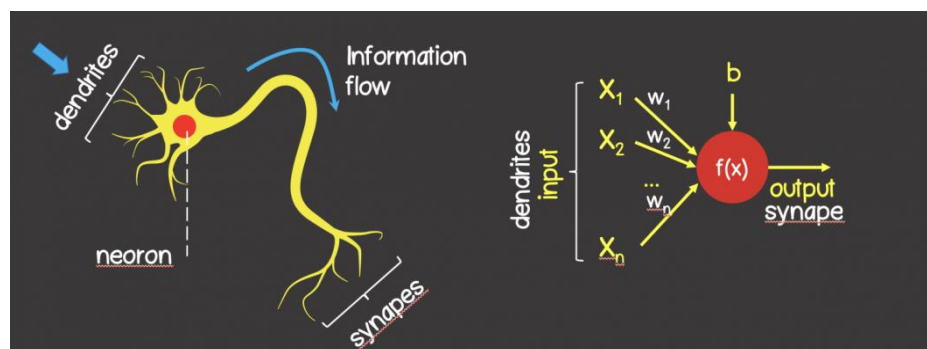


Рисунок 18 – Биологический и программный нейроны

Биологический нейрон получает сигнал от своих ответвлений – дендритов, которые связывают его с другими нейронами. Электрический сигнал, достигая определённого порога, возбуждает нейрон, что инициирует передачу сигнала к другим нейронам.

Персептрон – математическое представление нейронной сети, которое состоит из одного нейрона. Это нейрон выполняет две последовательные операции (Рисунок 19):

- вычисление суммы входных сигналов по их весам;

$$sum = \vec{X}^T \vec{W} + \vec{B} = \sum_{i=1}^n x_i w_i + b$$

- применение к сумме функции-активатора.

$$out = \varphi(sum)$$

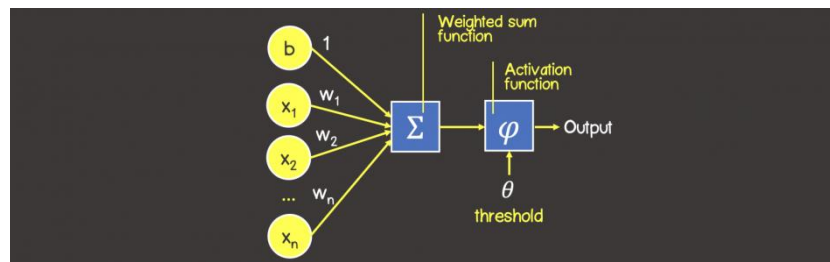


Рисунок 19 – Математическая модель персептрона

Функцией-активатором может быть любая дифференцируемая функция. Например, линейная функция или сигмоид. Обычно, выбор функции зависит от решаемой задачи. В большинстве случаев заранее известно, какой функцией необходимо решать какую-либо задачу, но, в противном случае, можно выбрать её методом простого перебора.

Изначально нейронная сеть рассматривается в условном состоянии, когда веса распределены на предыдущем этапе или же заданы в произвольном порядке, если это первый цикл обучения.

Вектор входных данных (input data) содержит в себе 2 параметра (feature) x_1 и x_2 , где x_1 и x_2 – рассматриваемые данные, которые могут быть любыми. Все вектора входных данных сопоставлены с векторами ожидаемых выходных данных (expected output).

Целиком обучение персептрона разделено на несколько шагов:

- прямое распространение ошибки (feedforward process);
- расчет функции ошибки;
- обратное распространение ошибки (backpropagation).

На первом шаге вычисляется сумма входных сигналов с учетом их весов. Затем, к ним применяется активационная функция. В итоге, всё сводится к вычислению тензора.

Тензор – своего рода контейнер данных, содержащий N осей и произвольное число элементов по каждой оси. Примерами тензоров является

вектор, как одноосевой тензор, и матрицы – двумерный тензор.

Во втором шаге рассчитывается функция ошибки – метрика, которая отражает различие между полученными данными с ожидаемым результатом.

Чаще всего используются следующие функции ошибок:

– среднеквадратичная ошибка (Mean Squared Error) – функция использует квадрат разности ожидаемого и получившегося результата, поэтому чувствительна к выбросам, то есть к значениям, сильно удалённым от других значений в наборе данных:

$$L = \frac{1}{N} \sum_{i=1}^N (Y_{predicted(i)} - Y_{expected(i)})^2$$

– среднеквадратичное отклонение (Root Mean Squared Error) – является аналогичной функцией, что и среднеквадратичная ошибка, однако упрощает задачу для человеческого анализа, так как имеет реальную физическую единицу измерения того, что анализирует нейронная сеть:

$$L = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_{predicted(i)} - Y_{expected(i)})^2}$$

– среднее отклонение (Mean Absolute Error) – отличается от выше перечисленных меньшей чувствительностью к выбросам:

$$L = \frac{1}{N} \sum_{i=1}^N |Y_{predicted(i)} - Y_{expected(i)}|$$

– перекрестная энтропия (Cross entropy) – применяется для задач, связанных с классификацией:

$$L = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M Y_{expected(ij)} \log(Y_{predicted(ij)})$$

, где

N – общее количество элементов в тренировочном наборе;

M – количество классов, получаемых при решении задач на классификацию данных;

$Y_{expected}(ij)$ — выходное ожидаемое значение;

$Y_{predicted}(ij)$ — выходное практическое значение.

Третий шаг обучения нейронной сети — обратное распределение ошибки. Его задача привести значения функции ошибки к минимуму. Для этого применяется метод градиентного спуска — в начале каждого нового шага обучения менять веса на противоположные по направлению к вектору градиенту:

$$\vec{w}^{(k+1)} = \vec{w}^k - \mu \nabla L(\vec{w}^k)$$

, где k — k -ый цикл обучения сети;

μ — шаг обучения (learning rate);

∇L — градиент функции-ошибки.

Если с каждым шагом ошибка уменьшается, то веса распределяются в правильном направлении.

3.3 Методы машинного обучения: случайный лес

Алгоритм случайного леса (Random Forest) — алгоритм машинного обучения, который является универсальным алгоритмом для решения широкого спектра задач. Так же, этот алгоритм не требует сложной теоретической основы. Суть случайного леса состоит в создании «ансамбля» решающих деревьев. Чем больше деревьев в ансамбле, тем точнее результат.

В общем случае порядок алгоритма выглядит следующим образом:

- загрузка данных;
- определение случайной выборки, по которым будут строиться деревья;
- построение дерева решений по каждой выборке;
- построение дерева ограничено количеством объектов в выборке или его «высотой»;
- получение результатов прогноза от каждого из деревьев;
- «голосование» — выбирается лучший признак, по которому делается разбиение в дереве до тех пор, пока не закончится выборка;

– прогноз с наибольшим количеством голосов считается окончательным результатом прогнозирования.

Теоретическая формула алгоритма случайного леса выглядит так:

$$a(x) = \frac{1}{N} \sum_{i=1}^N b_i(x)$$

, где

$a(x)$ – итоговый классификатор;

N – число деревьев;

i – i -е дерево;

b – решающее дерево;

x – сгенерированная выборка, основанная на данных.

Стоит уточнить, что для задач на классификацию данных необходимо выбирать решение голосования «по большинству», а для задач на регрессию – среднее значение.

На практике при увеличении числа деревьев качество теста перестаёт увеличиваться. Это сигнал о том, что модель переобучается. Избежать этого можно путём подбора оптимального числа решающих деревьев. Иными словами, необходимо зафиксировать количество деревьев на том моменте, когда система ещё не перешла в асимптотическое состояние.

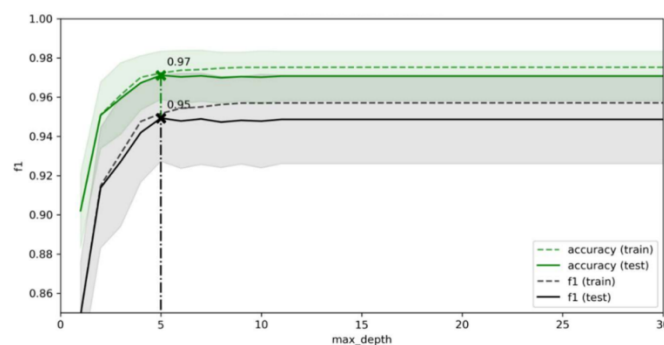


Рисунок 20 – Графики зависимости метрики качества и переход в асимптотическое состояние

При меньшей глубине дерева, алгоритм случайного леса работает быстрее. С увеличением глубины резко растёт качество обучения, но вместе с ним, увеличивается и время работы алгоритма. Обычно используют

максимальную глубину, кроме тех случаев, когда количество объектов слишком велико. Неглубокие деревья применяются в задачах, где присутствует большое количество выбросов.

3.4 Методы машинного обучения: кластеризация

Кластеризация – модель, относящаяся к модели обучения без учителя. Распространён в сферах распределения и статистического анализа больших объёмов данных.

Метод кластеризации помогает выявить среди крупного массива однотипных данных группы по которым информацию можно рассортировать и использовать это для дальнейшего анализа. Не существует конкретных критериев оптимальной кластеризации. Метод подразумевает под собой индивидуальные настройки параметров, зависящих от разработчика и ситуации.

Существует несколько методов кластеризации данных:

- метод k-средних;
- иерархическая кластеризация;
- метод t-SNE.

Метод k-средних является самым распространённым методом кластеризации данных. Он является итеративным алгоритмом, содержащим в основе минимизацию суммарных квадратичных отклонений точек кластеров от средних координат (центроидов).

Кластеризация начинается с определения числа кластеров, необходимых для работы. После этого из входных данных случайно выбираются несколько элементов выборки. В соответствие к ним, строятся кластера. Каждый кластер содержит в себе одну точку, причём эти точки при являются центроидом кластера.

Затем, по методу ближайшего соседа, находится вторая точка, рядом с центроидом и пересчитывается новый центр кластера с учётом новых точек. Завершением для алгоритма является условие, когда центроид перестаёт

менять свои координаты. Так центроид содержит в себе набор признаков своего кластера, которые описывают усреднённые значения выделенных классов.

Метод иерархической кластеризации создаёт иерархическое древо кластеров. В начале алгоритма каждому элементу присваивается свой персональный кластер. Затем, два соседних кластера объединяются в один общий. Таким образом алгоритм работает до тех пор, пока не сформирует один общий кластер. Работу алгоритма можно представить в виде дендрограммы.

При работе с большим объёмом данных, этот метод показывает себя хуже, чем метод k-средних, так как временная сложность метода k-средних $O(n)$, а для метода иерархической кластеризации – $O(n^2)$.

Метод t-SNE (t-distributed stochastic neighbor embedding) моделирует каждый объект пространства высокой размерности в двух- или трехкоординатную точку таким образом, что близкие по характеристикам элементы данных в многомерном пространстве (например, датасете с большим числом столбцов) проецируются в соседние точки, а разнородные объекты с большей вероятностью моделируются точками, далеко отстоящими друг от друга. Математическое описание работы метода можно найти [здесь](#).

4 Практика

4.1 Описание обучающей выборки

В настоящее время активно обсуждаются способы применения машинного обучения для мониторинга интернет трафика. Основным аспектом исследований в данном направлении является поиск возможностей применения и оценка существующих и разрабатываемых алгоритмов.

Например, в исследовании [18] сетевой трафик классифицируется и фильтруется методом «случайного леса». Представлены результаты как с фоновым трафиком, так и без него. Метод показал высокую эффективность и скорость при работе с записанными данными трафика, однако при работе с потоком в режиме реального времени теряется точность из-за чувствительности модели к временным промежуткам. Для online режима работы необходимо оптимизировать модель, но в работе отсутствуют итоговые настройки.

В описанном выше и подобных ему исследованиях фиксируются оценки точности работы системы для обнаружения атак. Данная задача подразумевает построение макета, реализующим на практике методы машинного обучения.

Программа-макет разработана на языке Python с использованием пакета в открытом доступе scikit-learn. В качестве датасета для обучения системы был выбран один из популярных комплект данных – «Intrusion Detection Evaluation Dataset» CICIDS2017, разработанный и подготовленный Канадским институтом кибербезопасности.

Датасет получен путём моделирования деятельности 25ти легальных пользователей и различных вредоносных программ, иммитирующих нарушителей безопасности трафика [20]. Датасет содержит примерно 50 Гб необработанных данных в виде PCAP и 8 файлов в формате CSV, прошедших предобработку. Эти файлы содержат информацию о проведённых сессиях и сделанных за это время наблюдениях.

В табличном виде файлы и число прошедших по моделируемой сети данных представлены в таблицах ниже (Табл. 1 и Табл. 2).

Таблица 1 – Наборы данных CICIDS2017

№	Название файла	Содержащиеся атаки
1	Monday-WorkingHours.pcap_ISCX.csv	Benign (обычный трафик)
2	Tuesday-WorkingHours.pcap_ISCX.csv	Benign, FTP-Patator, SSH-Patator
3	Wednesday-workingHours.pcap_ISCX.csv	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
4	Thursday-WorkingHours-MorningWebAttacks.pcap_ISCX.csv	Benign, Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS
5	Thursday-WorkingHours-AfternoonInfiltration.pcap_ISCX.csv	Benign, Infiltration
6	Friday-WorkingHours-Morning.pcap_ISCX.csv	Benign, Bot
7	Friday-WorkingHours-AfternoonPortScan.pcap_ISCX.csv	Benign, PortScan
8	Friday-WorkingHours-AfternoonDDos.pcap_ISCX.csv	Benign, DDoS

Таблица 2 – Число данных CICIDS2017

№	Тип записи	Количество записей
1	BENIGN	2359087
2	DoS Hulk	231072
3	PortScan	158930
4	DDoS	41835
5	DoS GoldenEye	10293
6	FTP-Patator	7938
7	SSH-Patator	5897
8	DoS slowloris	5796
9	DoS Slowhttptest	5499
10	Bot	1966
11	Infiltration	36
12	Heartbleed	11

13	Web Attack – Brute Force	1507
14	Web Attack – XSS	652
15	Web Attack – SQL Injection	21

4.2 Формирование признакового пространства

Для сокращения времени обучения был создан единый класс WebAttacks с использованием данных файла Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv. Этот набор содержит 458968 записей. 2180 записей из набора – записи об атаках. Остальные записи описывают нормальное поведение трафика.

Для разработки системы машинного обучения критически важно знать, какие критерии необходимо руководствоваться для вынесения решений. Это влияет на точность ответов системы и на длительность обучения [21]. Это обязательная процедура не только для подготовительного этапа обучения, но и для более поздних этапов жизни машины, таких как корректирующая выборка [22].

Изначально исключены из признакового пространства такие признаки, как «Source IP», «Source Port», «Destination IP», «Destination Port» и «Protocol» так как признаки, соответствующие статистикам сетевого трафика, более подходят при рассмотрении общих случаев. Так же эти характеристики легко подделать, поэтому учитывать их в обучаемой модели не нужно [19].

Метод `sklearn.ensemble.RandomForestClassifier` (атрибут `feature_importances_`) – встроенный метод пакета `scikit-learn`. Он реализует энтропийный подход к оценке важности признаков.

Благодаря ему, из признакового пространства были убраны «Init_Win_bytes_backward» и «Init_Win_bytes_forward». Эти характеристики имели слишком высокую взаимосвязь с метками класса в обучающей выборке. Это может свидетельствовать о погрешностях, допущенных в наборе данных. Итоги распределительного анализа представлены на изображении ниже (Рисунок 16).

На изображении (Рисунок 17) корреляционная матрица с линейными коэффициентами корреляции (коэффициентами корреляции Пирсона), рассчитанными для всех пар наиболее значимых признаков.

Из анализа следует, что некоторые пакеты имеют между собой сильную корреляцию, а именно:

- 5) «Fwd Packets/s» и «Flow Packets/s»;
- 6) «Flow IAT Max» и «Fwd IAT Max»;
- 2) «Total Length of Fwd Packets» и «Subflow Fwd Bytes»;
- 4) «Flow Duration» и «Fwd IAT Total»;
- 1) «Packet Length Mean» и «Average Packet Size»;
- 3) «Avg Fwd Segment Size» и «Fwd Packet Length Mean».

Исходя из корреляционного анализа были удалены: «Subflow Fwd Bytes», «Packet Length Mean», «Fwd IAT Max», «Fwd IAT Total», «Fwd Packets/s», «Avg Fwd Segment Size». После исключения признаков пространство сократилось до 10ти признаков:

- 1) «Fwd Packet Length Mean», средняя длина переданных пакетов;
- 2) «Flow Bytes/s», скорость потока данных;
- 3) «Fwd Packet Length Max», максимальная длина пакета;
- 4) «Average Packet Size», длина пакета;
- 5) «Fwd Header Length», сумма длин заголовков пакетов;
- 6) «Total Length of Fwd Packets», сумма длин переданных пакетов;
- 7) «Fwd IAT Std», среднеквадратическое отклонение значения межпакетного интервала;
- 8) «Flow IAT Mean», среднее значение межпакетного интервала;
- 9) «Max Packet Length», максимальная длина пакета;
- 10) «Fwd IAT Min», минимальное значение межпакетного интервала.

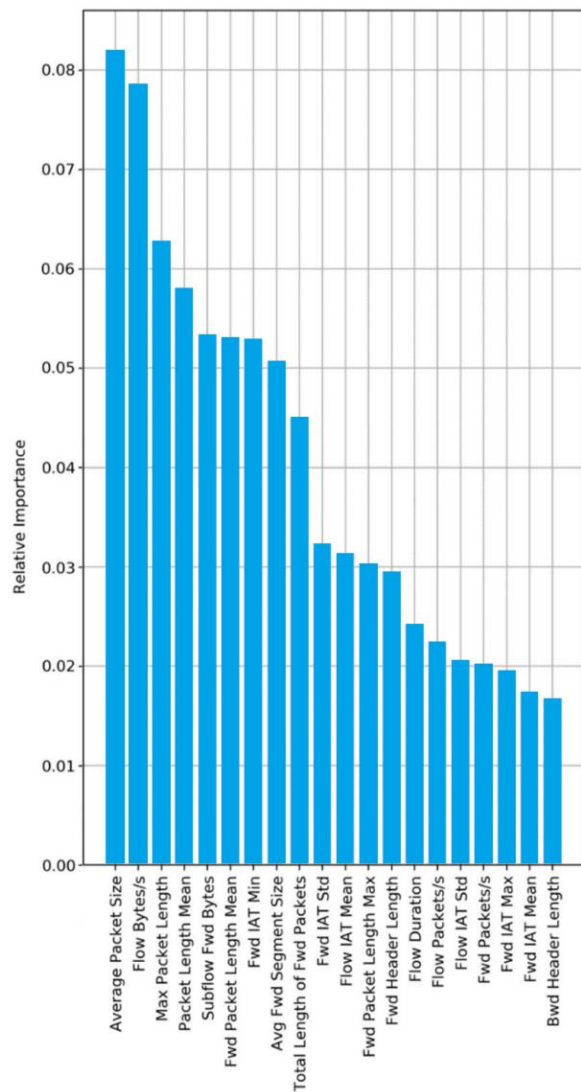


Рисунок 16 – Результаты анализа важности признаков

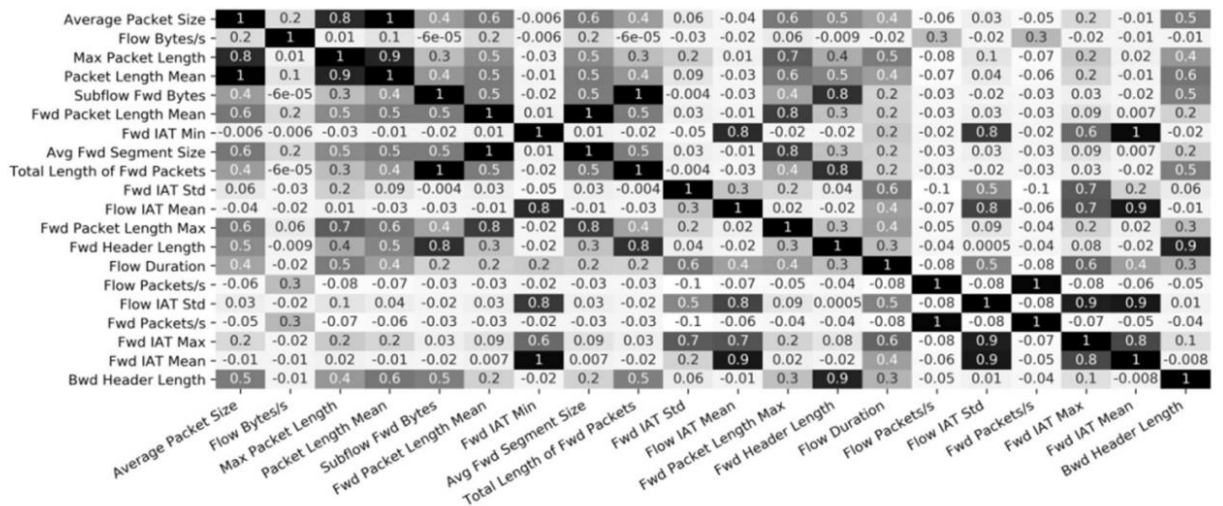


Рисунок 17 – Корреляционный анализ значимых признаков

4.3 Оценка обучения машины

Корректность ответов проверяется при помощи специальных метрик:

- доля правильных ответов (accuracy);

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- точность (precision, доверие к классификатору);

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

- полнота (recall, сколько объектов необходимого класса определяют классификатор);

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

- F1-мера (F1-measure, является гармонической средней между точностью и полнотой).

$$F1 = 2 \times Precision \times \frac{Recall}{Precision + Recall}$$

, где

TP (True Positive) – истинно-положительный ответ;

TN (True Negative) – истинно-отрицательный ответ;

FP (False Positive) – ложное срабатывание (ошибка первого рода);

FN (False Negative) – пропуск ответа (ошибка второго рода).

Когда определяются метрики оценки качества обучения, составляют матрицу ошибок (confusion matrix), элементы которой соответствуют верных и ошибочных ответов по результатам тестирования классификатора (Таблица 3).

Таблица 3 – Описание матрицы ошибок

	Ответ классификатора: «−»	Ответ классификатора: «+»
Правильный ответ: «−»	TN	FP
Правильный ответ: «+»	FN	TP

4.4 Результаты моделирования

В первом случае смоделирована трёхслойная нейронная сеть размерностью (INPUT_DIM x 128) - (128x128) - (128xNUM_CLASSES). Применяв оценку по доле правильных ответов, получаем точность 0.84.

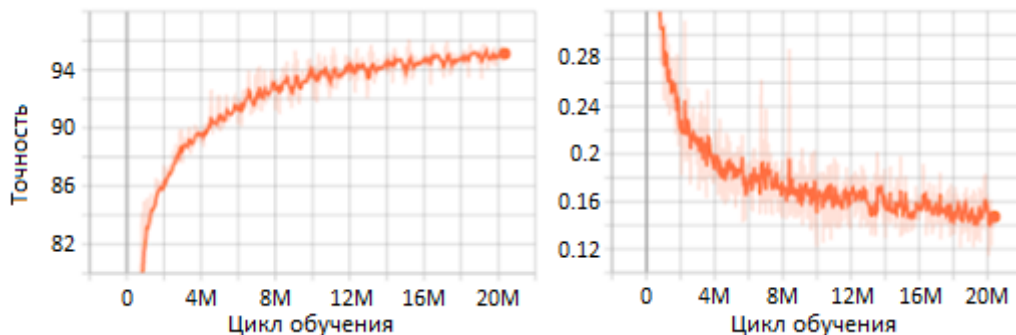


Рисунок 21 – Графики процесса обучения нейронной сети точности (левый) и ошибки (правый)

Во втором случае было увеличено число циклов обучения и применён метод перекрёстной проверки (k-Fold Cross Validation). Этот метод проверяет навыки обучения нейронной сети более точно, нежели предыдущий метод.

Существуют общие тактики, которые вы можете использовать для выбора значения k для вашего набора данных. Существуют широко используемые варианты перекрестной проверки, такие как стратифицированная и повторная кросс-валидация, которые доступны в `scikit-learn`.

Кросс-валидация, или перекрестная проверка — это процедура повторной выборки, используемая для оценки моделей машинного обучения на ограниченной Выборке (Sample) данных.

Процедура имеет единственный параметр, называемый k , который означает количество групп, на которые должна быть разбита данная выборка данных. Таким образом, эту процедуру часто называют k -кратной перекрестной проверкой. Когда выбрано конкретное значение k , его можно использовать вместо k в ссылке на модель, например, $k=5$.

Это популярный метод, поскольку он прост для понимания и обычно приводит к менее предвзятой или менее оптимистичной оценке навыков

модели, чем другие методы.

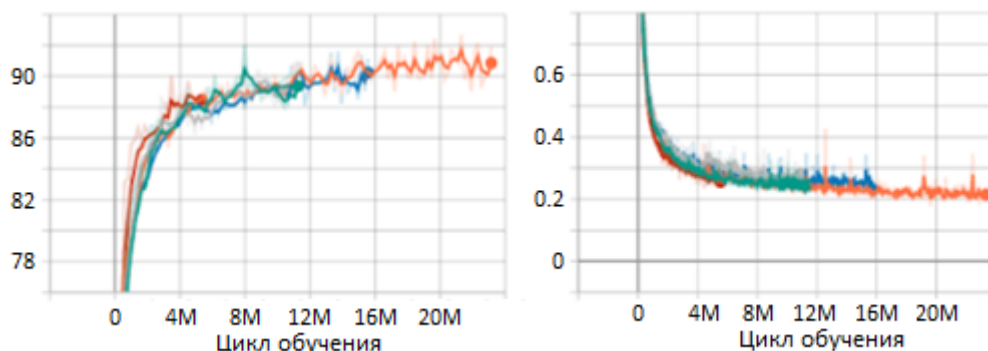


Рисунок 22 – Графики процесса обучения нейронной сети с пятикратной точностью (левый) и ошибки (правый)

Таблица 4 – Результаты оценки обучения нейронной сети

Трёхслойная нейронная сеть	Трёхслойная нейронная сеть с перекрёстной проверкой ($k = 5$)
0.952	0.933

Исходя из результатов моделирования, перекрёстная проверка даёт более точную оценку результатам обучения нейронной сети.

Датасет CICIDS-2017 применим для задачи поиска аномалий трафика, однако имеет некоторые недоработки, что может вызвать непредвиденные осложнения в процессе применения обученных на его основе нейронных сетей для анализа реального трафика.

Выводы

Рассмотрены основные понятия анализа трафика, методы сбора трафика, а так же его анализ. Исследована структура сети интернет вещей, стандартная модель их поведения, и изучена последовательность шагов для выделения из трафика, генерируемого устройствами интернета вещей случайной составляющей, оставшиеся после исключения основных характеристик, в которой может содержаться аномалия.

Изучены методы машинного обучения и устройство обучающего датасета CICIDS-2017. Рассмотрены способы построения архитектуры машинного обучения, и проведено исследование самого процесса обучения.

Источники

1. Wang C., Lin H., & Jiang H. CANS: Towards congestion-adaptive and small stretch emergency navigation with wireless sensor networks // IEEE Transactions on Mobile Computing. 2015. Vol. 15(5). P. 1077-1089.
2. Татарникова Т.М., Богданов П.Ю., Краева Е.В. Предложения по обеспечению безопасности системы умного дома, основанные на оценке потребляемых ресурсов // Проблемы информационной безопасности. Компьютерные системы. 2020. № 4. С. 88-94.
3. Baddar S.A.-H., Merlo A., Migliardi M. Anomaly Detection in Computer Networks: A State-of-the-Art Review // Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications. 2014. vol. 5. no. 4. P. 29–64.
4. Шелухин О.И., Сакалема Д.Ж., Филинова А.С. Обнаружение вторжений в компьютерные сети. Сетевые аномалии. – М.: Горячая линия – Телеком, 2013. – 220 с.
5. Bejtlich Richard., Why Collect Full Content Data? <http://taosecurity.blogspot.com>, 2012
6. Quittek J., Zseby T., Claise B., Zander S., RFC 3917: Requirements for IP Flow Information Export (IPFIX). Internet Engineering Task Force, 2004. <http://tools.ietf.org/html/rfc3917>
7. RFC 7011, Specification of the IP Flow Information Export (IPFIX) Protocol, a standardized network flow format, provides a more technical definition of flow. <http://tools.ietf.org/search/rfc7011>
8. National Information Standards Organization (NISO). Understanding Metadata. NISO, 2004.
9. Aceto G., Botta A., Pescape A., Westphal C. Efficient Storage and Processing of High-Volume Network Monitoring Data, IEEE Transactions on Network and Service Management, vol. 10, issue 2, pp. 162-175, 2013.
10. Aceto G., Botta A., de Donato W., Pescape A., Cloud Monitoring: A Survey, Computer Networks vol.57, issue 9, pp. 2093-2115, 2013.

11. Deri L., Cardigliano A., Fusco F., 10 Gbit Line Rate Packet-to-Disk Using n2disk, Proceedings IEEE INFOCOM. Turin, Italy, Apr. IEEE, pp. 3399-3404, 2013.
12. Banks D., Custom Full Packet Capture System, SANS, 2013.
13. Francois J., State R., Engel T., Aggregated Representations and Metrics for Scalable Flow Analysis, IEEE Conference on Communications and Network Security (CNS). Washington, D.C., pp. 478-482, 2013.
14. Lee Perry. Internet of Things Architects. Packt Publishing, 2018. 514 p.
15. Сахаров Д. В., Козлов Д. С. Обнаружение аномального поведения устройства IoT в сети на основе модели трафика. 2018. С. 51-55.
16. Basford P. J., Johnston S. J., Perkins C. S., Garnock-Jones T., Tso F. P., Pezaros D., Cox S. J. Performance analysis of single board computer clusters. Future Generation Computer Systems. 2020. Vol. 102. P. 278-291.
17. Dziubenko I. N. and Tatarnikova T. M. Algorithm for Solving Optimal Sensor Devices Placement Problem in Areas with Natural Obstacles. 2018 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF). 2018. P. 1-5. DOI: 10.1109/WECONF.2018.8604325.
18. Шелухин О.И., Ванюшина А.В., Габисова М.Е. Фильтрация нежелательных приложений интернеттрафика с использованием алгоритма классификации Random Forest. Вопросы кибербезопасности, № 2 (26), 2018 г., стр. 44-51. / Sheluhin O., Vanyushina A., Gabisova M. The Filtering of Unwanted Applications in Internet Traffic Using Random Forest Classification Algorithm. Voprosy kiberbezopasnosti, № 2 (26), 2018, pp. 44-51 (in Russian).
19. Kostas K. Anomaly Detection in Networks Using Machine Learning. Master thesis. School of Computer Science and Electronic Engineering, University of Essex, 2018, 70 p.
20. Sharafaldin I., Lashkari A.H., Ghorbani Ali A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proc. of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108-116.

21. Leskovec J., Rajaraman A., Ullman J. Mining Of Massive Datasets. Cambridge University Press, 2014. 476 p.

22. Domingos P. A Few Useful Things to Know about Machine Learning. Communications of the ACM, vol. 55, № 10, 2012. pp. 78-87.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет
аэрокосмического приборостроения»

ОТЗЫВ РУКОВОДИТЕЛЯ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

на тему Обнаружение аномального трафика «Интернета вещей» методами машинного обучения

выполненную студентом группы № 5823

Сверликовым Александром Владимировичем

фамилия, имя, отчество студента

по направлению подготовки/
специальности
системы связи

11.03.02
(код)

Инфокоммуникационные технологии и
наименование направления подготовки/специальности)

(наименование направления подготовки/специальности)

Актуальность темы работы:

Известны несколько методов детектирования аномального трафика, в основном базирующиеся на моделях сигнатурного и штатного поведения, но с появлением проблемы больших данных активно изучается и апробируется нейросетевой подход.

В бакалаврской работе, выполненной Сверликовым Александром Владимировичем предложен подход к обнаружению аномального трафика интернета вещей, основанный на нейронной сети. Обученная нейронная сеть может стать одним из модулей системы обнаружения вторжений, работающей на уровне умных устройств. В настоящей работе производится обучение нейронной сети на известной выборке детектировать атаки на качественном уровне (есть/нет атака).

Цель и задачи работы:

Цель работы состояла в том, чтобы апробировать нейросетевой подход в детектировании аномального трафика, в частности DDOS атак.

Общая оценка выполнения поставленной перед студентом задачи, основные достоинства и недостатки работы:

Общая оценка выполнения поставленной перед Сверликовым А.В. задачи положительная. К достоинствам работы следует отнести то, что соискателю удалось продемонстрировать возможности нейронной сети прямого распространения при детектировании аномального трафика. К недостаткам предложенного инструмента в виде нейронной сети можно отнести необходимость иметь данные для ее обучения.

Степень самостоятельности и способности к исследовательской работе студента (умение и навыки поиска, обобщения, анализа материала и формулирования выводов):

Александр Владимирович продемонстрировала высокую степень самостоятельности в изучении современных методов детектирования аномального трафика, умения и навыки поиска актуальной информации по теме исследования, обобщения и анализа найденного материала, формулирования выводов по результатам исследования.

Проверка текста выпускной квалификационной работы с использованием системы Антиплагиат (www.antiplagiat.ru), проводившаяся «14» июня 2022 г. показывает оригинальность содержания на уровне 74,7%

Степень грамотности изложения и оформления материала:

Пояснительная записка хорошо структурирована и ее содержание соответствует последовательному описанию предлагаемого решения задачи детектирования аномального трафика. Должное внимание уделено анализу существующих коммерческих решений, их достоинствам и недостаткам. Проведенный анализ позволил соискателю апробировать нейросетевой подход к детектированию атак. Пояснительная записка содержит подробное описание всех сопутствующих нейросетевому подходу процедур – нормализацию входных данных, выбор метрик обучения, параметров нейронной сети, архитектуры нейронной сети и т.п.

Оценка деятельности студента в период подготовки выпускной квалификационной работы (добросовестность, работоспособность, ответственность, аккуратность и т.п.):

Сверликов Александр Владимирович показал себя ответственным и исполнительным студентом. Возможно, ему удалось бы достичь больших результатов, поскольку работа может иметь продолжение. Все поставленные перед Александром Владимировичем задачи выполнены в полном объеме, отмечу высокую работоспособность в определенный период, способность к поиску и анализу решений.

Общий вывод:

Считаю, что Сверликов Александр Владимирович заслуживает присвоения квалификации бакалавра по направлению 11.03.02 – Инфокоммуникационные технологии и системы связи.

В работе не содержится информация с ограниченным доступом и отсутствуют сведения, представляющие коммерческую ценность

Руководитель

проф., д.т.н., проф.

должность, уч. степень, звание



14.06.2022

подпись, дата

Т.М. Татарникова

инициалы, фамилия



guap.ru

ОБНАРУЖЕНИЕ АНОМАЛЬНОГО ТРАФИКА «ИНТЕРНЕТА ВЕЩЕЙ» МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ

Сверликов А.В.

Научный руководитель: Т.М. Татарникова

Ограничения устройств IoT:

- используют батареи в качестве источника питания,
- обладают низкой вычислительной мощностью,
- ограниченным объемом памяти.



Ограничения обусловили разработку протоколов, обеспечивающих передачу и обработку данных с минимальной вычислительной и коммуникационной нагрузкой.



Устройства интернета вещей уязвимы к аномальному трафику – различным атакам, реализация которых может нанести серьезный ущерб эксплуатируемому оборудованию IoT и даже физический ущерб людям

Взаимодействие Machine-to-Machine (M2M) – основа интернета вещей и подключения новых устройств в единую сеть.

Организация M2M – по протоколу CoAP (Constrained Application Protocol).

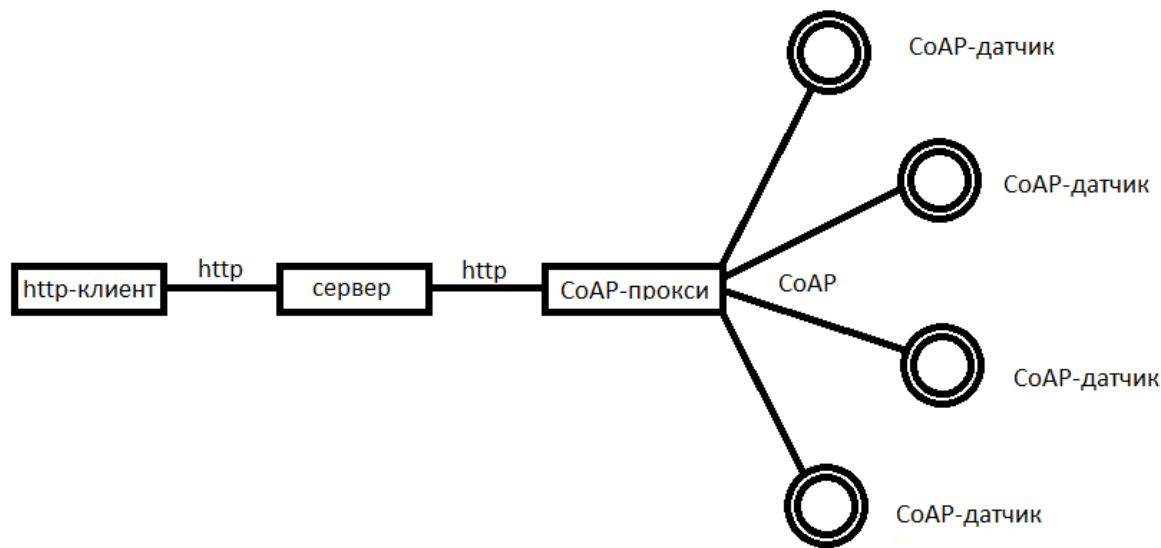


Рис. 1 – Общий случай взаимодействия клиента с умным устройством

IDS для интернета вещей, имеет иерархическую структуру, как и сама сеть IoT.

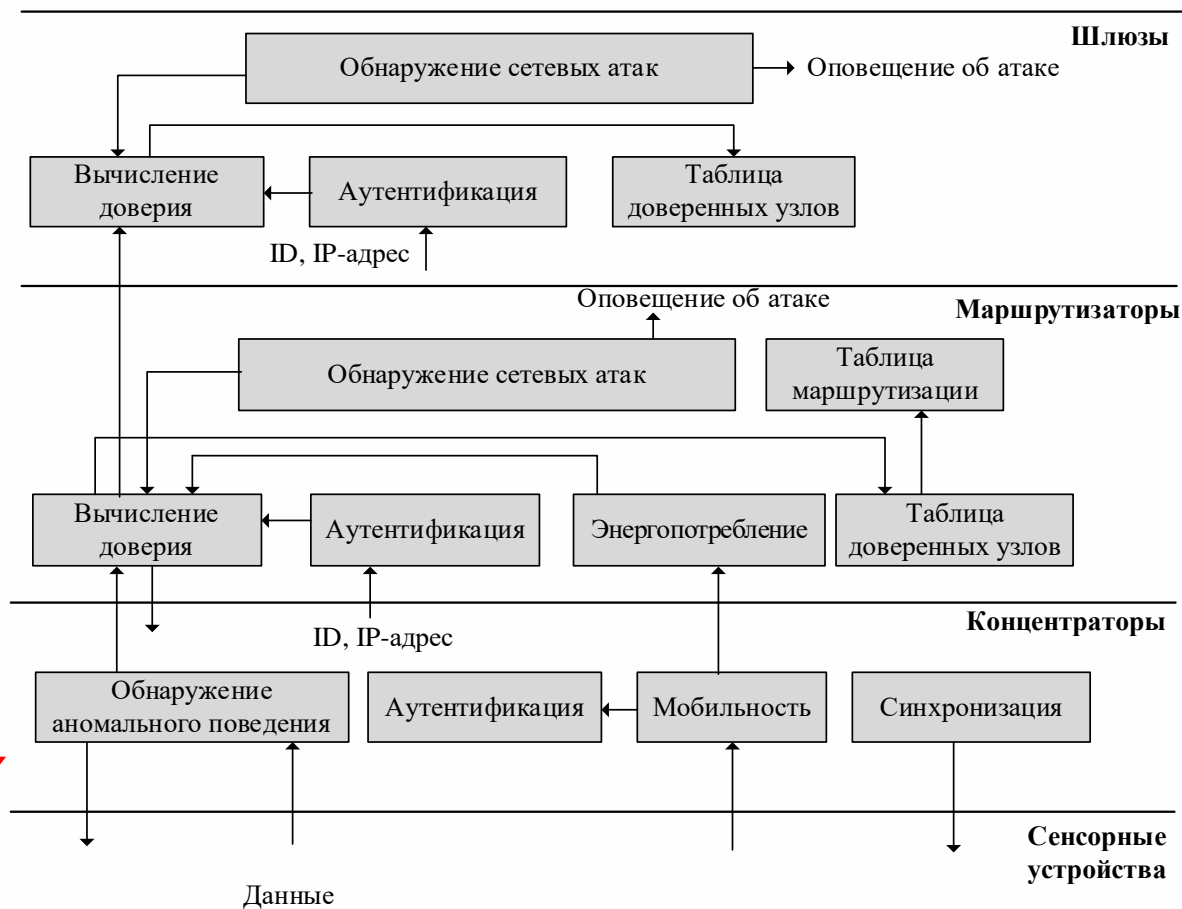


Рис. 5 – Компоненты IDS интернета вещей

Способы машинного обучения:

- обучение с учителем;
- обучение без учителя;
- обучение с частичным привлечением учителя;
- обучение с подкреплением;
- глубинное обучение.

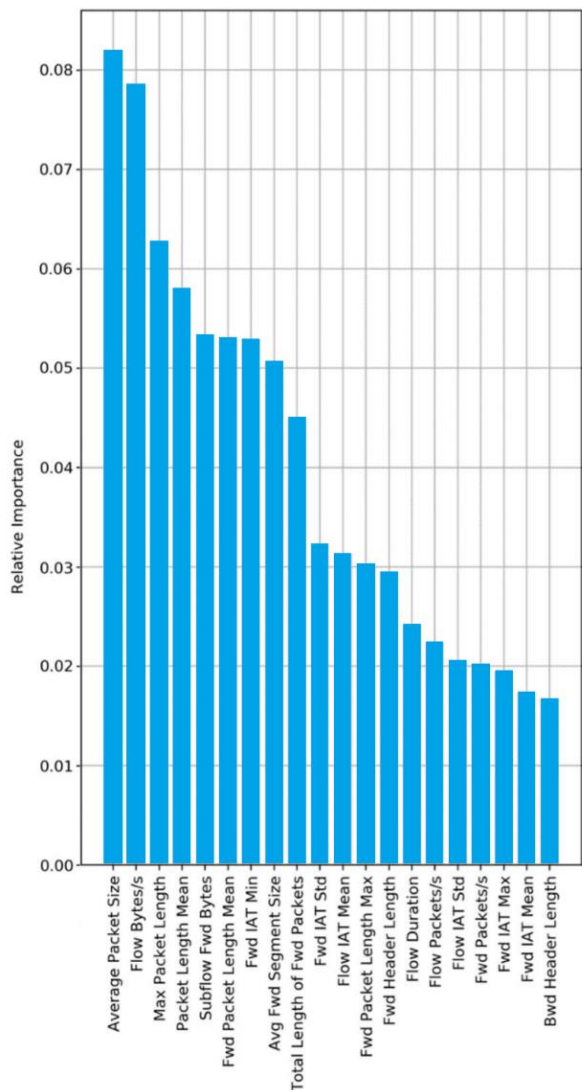
Методы построения архитектуры машинного обучения:

- нейронные сети;
- дерево решений;
- случайный лес;
- кластеризация.

Для проведения исследований будет построена трёхслойная нейронная сеть (INPUT x 128 - 128x128 - 128x OUTPUT), основанная на обучении с учителем.

№	Название файла	Содержащиеся атаки
1	Monday-WorkingHours.pcap_ISCX.csv	Benign (обычный трафик)
2	Tuesday-WorkingHours.pcap_ISCX.csv	Benign, FTP-Patator, SSH-Patator
3	Wednesday-workingHours.pcap_ISCX.csv	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
4	Thursday-WorkingHours-MorningWebAttacks.pcap_ISCX.csv	Benign, Web Attack – Brute Force, Web Attack – Sql Injection, Web Attack – XSS
5	Thursday-WorkingHours-AfternoonInfiltration.pcap_ISCX.csv	Benign, Infiltration
6	Friday-WorkingHours-Morning.pcap_ISCX.csv	Benign, Bot
7	Friday-WorkingHours-AfternoonPortScan.pcap_ISCX.csv	Benign, PortScan
8	Friday-WorkingHours-AfternoonDDos.pcap_ISCX.csv	Benign, DDoS

№	Тип записи	Количество записей
1	BENIGN	2359087
2	DoS Hulk	231072
3	PortScan	158930
4	DDoS	41835
5	DoS GoldenEye	10293
6	FTP-Patator	7938
7	SSH-Patator	5897
8	DoS slowloris	5796
9	DoS Slowhttptest	5499
10	Bot	1966
11	Infiltration	36
12	Heartbleed	11
13	Web Attack – Brute Force	1507
14	Web Attack – XSS	652
15	Web Attack – SQL Injection	21



Формирование признакового пространства

Для разработки системы машинного обучения критически важно знать, какие критерии необходимо руководствоваться для вынесения решений. Это влияет на точность ответов системы и на длительность обучения.

Процедура является обязательной не только для подготовительного этапа обучения, но и для более поздних этапов жизни машины, таких как корректирующая выборка.

В исследованиях для оценки обучаемости нейронной сети применяется метрика доли правильных ответов (Accuracy):

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

	Ответ классификатора: «-»	Ответ классификатора: «+»
Правильный ответ: «-»	TN	FP
Правильный ответ: «+»	FN	TP

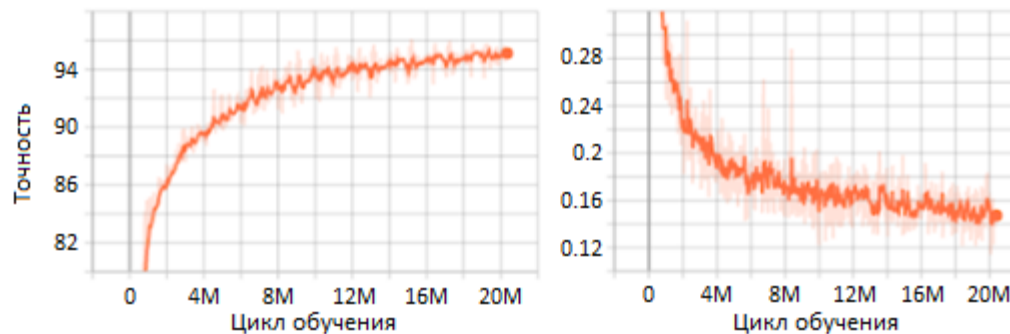
TP (True Positive) – истинно-положительный ответ

TN (True Negative) – истинно-отрицательный ответ

FP (False Positive) – ложное срабатывание

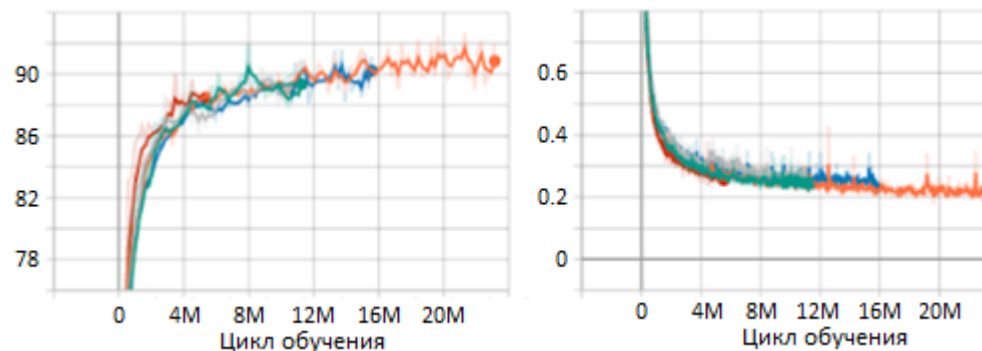
FN (False Negative) – пропуск срабатывания

Графики оценки точности обучения нейронной сети



Лучшая точность: 95,2%

Оценка точности обучения нейронной сети с использованием перекрёстной проверки ($k = 5$)



Лучшая точность: 93,3%

1. Рассмотрены основные понятия анализа трафика, методы сбора трафика, а так же его анализ. Исследована структура сети интернет вещей, стандартная модель их поведения, и изучена последовательность шагов для выделения из трафика, генерируемого устройствами интернета вещей случайной составляющей, оставшиеся после исключения основных характеристик, в которой может содержаться аномалия.

2. Изучены методы машинного обучения и устройство обучающего датасета CICIDS-2017. Рассмотрены способы построения архитектуры машинного обучения, и проведено исследование самого процесса обучения.



guap.ru

Спасибо за внимание!