|  |  |
| --- | --- |
| Изображение выглядит как текст, эмблема, герб, нашивка  Автоматически созданное описание | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_\_\_\_

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ***

по дисциплине \_\_\_\_\_\_\_\_\_\_\_\_\_ Технологии машинного обучения \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по теме\_ «Выявление подозрительного трафика \_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ с использованием методов машинного обучения»\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент ИУ5-65Б **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** С.С. Костин

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.И. Антонов

(Подпись, дата) (И.О.Фамилия)

Консультант **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

(Подпись, дата) (И.О.Фамилия)

2024 г.

# Аннотация

В курсе предмета «Технологии машинного обучения» была поставлена задача поиска аномальных сессий в определенном наборе сессий. Исследование можно разделить несколько частей: анализ и подготовка датасета, разработка моделей для классификации сессий, анализ результатов. В исследовании будут использоваться несколько Python-библиотек для анализа данных. Сам анализ будет выполняться на платформе в интерактивной среде разработки «Jupyter Notebook».

В результате работы будет получена реальная модель, пригодная для использования в сети для обнаружения подозрительного трафика.

# Содержание

[Аннотация 2](#_Toc167729862)

[Содержание 3](#_Toc167729863)

[ВВЕДЕНИЕ 4](#_Toc167729864)

[Постановка задачи 4](#_Toc167729865)

[Работа с датасетом 6](#_Toc167729866)

[Обзор датасета 6](#_Toc167729867)

[Подготовка датасетов 9](#_Toc167729868)

[Разработка моделей 20](#_Toc167729869)

[Проверка моделей 27](#_Toc167729870)

[Анализ результатов и сравнение моделей 34](#_Toc167729871)

[ЗАКЛЮЧЕНИЕ 35](#_Toc167729872)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 36](#_Toc167729873)

# ВВЕДЕНИЕ

Цель данной работы – разработка модели для обнаружения аномальных сессий в сетевом трафике. Анализ проводится на общедоступном датасете [1]. В ходе работы производится анализ датасета, его преобразование и подготовка; разрабатывается модель, наиболее успешно решающая поставленную цель. Для достижения поставленной цели решаются следующие задачи:

1. Анализ датасета, его подготовка для использования в обучении (факторизация, масштабирование и иные методы) различными способами;
2. Обучение и тестирование собственной модели для поиска аномалий в сетевом трафике с использованием Jupyter Notebook, оценка точности модели, сравнение результатов с другими моделями;
3. Сравнение результатов различных моделей, сравнение результатов различно подготовленных датасетов.

# Постановка задачи

В современном мире машинное обучение используется повсеместно, в том числе его применение разумно для анализа сетевого трафика. Для выявления опасных или подозрительных сессий используются различные подходы и методы. В данном исследовании поставлена задача не только выявить эти сессии, но и определить, какие методы обучения модели подходят для этого лучше всего.

Существует множество методов, относящиеся к решению данной задачи, которые претерпели трансформацию вместе с эволюцией сетевого трафика: классификация по номерам портов, глубокий анализ пакетов, стохастический анализ пакетов и с использованием машинного обучения [2]. В частности, в методах машинного обучения выделяются [2, 3]:

1. Наивный байесовский классификатор
2. Метод опорных векторов
3. Метод k-ближайших соседей
4. Дерево принятия решений
5. Бэггинг
6. Бустинг
7. Нейронные сети

В данной работе рассматривается последний метод, также изучаются влияние исходных данных и архитектуры нейросети на результирующие показания.

# Работа с датасетом

## Обзор датасета

Исходный датасет [1] представлен в 6.26 миллионах строчках и 74 колонках. Колонки с их типом данных и описанием представлены в таблице 1.

Датсет был подготовлен на основе искусственно созданного в контролируемой лабораторной среде трафика, эмулирующего 3 типа конечных устройств: генератор подозрительного трафика, небезопасный хост и обычный пользователь. В ходе работы использовались различные протоколы работы всех трёх машин: IP, TCP, UDP, ICMP, HTTP, DNS, SMTP, POP, IMAP, FTP [1].

*Таблица 1. Обзор колонок исходного датасета.*

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Тип данных** | **Описание** |
| frame.number | int64 | Номер пакета в файле захвата. |
| frame.len | int64 | Длина пакета. |
| frame.time | object | Временная метка момента захвата пакета. |
| frame.time\_epoch | float64 | Временная метка в секундах с начала эпохи (1 января 1970 года) момента захвата пакета. |
| frame.protocols | object | Список всех протоколов, используемых в пакете. |
| eth.src | object | MAC-адрес источника. |
| eth.dst | object | MAC-адрес назначения. |
| eth.type | object | Поле типа Ethernet-кадра. |
| ip.src | object | IP-адрес источника. |
| ip.dst | object | IP-адрес назначения. |
| ip.len | float64 | Общая длина IP-пакета, включая заголовки и данные. |
| ip.ttl | float64 | Значение времени жизни (TTL) для IP-пакета. |
| ip.flags | object | Флаги, установленные в заголовке IP. |
| ip.frag\_offset | float64 | Смещение фрагментации для IP-пакета. |
| ip.proto | float64 | Протокол, используемый в IP-пакете. |
| ip.version | float64 | Версия IP-протокола (IPv4 или IPv6). |
| ip.dsfield | object | Поле Дифференцированных Служб (для обеспечения качества обслуживания). |
| ip.checksum | object | Контрольная сумма заголовка IP. |
| tcp.srcport | float64 | Порт источника для TCP-пакетов. |
| tcp.dstport | float64 | Порт назначения для TCP-пакетов. |
| tcp.len | float64 | Длина сегмента данных TCP. |
| tcp.seq | float64 | Номер последовательности TCP-пакета. |
| tcp.ack | float64 | Номер подтверждения TCP-пакета. |
| tcp.flags | object | Флаги TCP-пакета (SYN, ACK, FIN и т.д.). |
| tcp.flags.syn | float64 | Флаг SYN TCP-пакета. |
| tcp.flags.ack | float64 | Флаг ACK TCP-пакета. |
| tcp.flags.fin | float64 | Флаг FIN TCP-пакета. |
| tcp.flags.reset | float64 | Флаг RESET TCP-пакета. |
| tcp.window\_size | float64 | Размер окна TCP-пакета (для управления потоком). |
| tcp.checksum | object | Контрольная сумма заголовка TCP. |
| tcp.stream | float64 | Индекс TCP-потока, к которому относится пакет. |
| udp.srcport | float64 | Порт источника для UDP-пакетов. |
| udp.dstport | float64 | Порт назначения для UDP-пакетов. |
| udp.length | float64 | Длина UDP-пакета. |
| udp.checksum | float64 | Контрольная сумма заголовка UDP. |
| icmp.type | float64 | Тип ICMP-пакета. |
| icmp.code | float64 | Код ICMP-пакета. |
| icmp.checksum | float64 | Контрольная сумма ICMP-пакета. |
| http.request.method | object | Метод HTTP-запроса (GET, POST и т.д.). |
| http.request.uri | object | URI, на который направлен HTTP-запрос. |
| http.request.version | object | Версия HTTP, используемая в запросе. |
| http.request.full\_uri | object | Полный URI HTTP-запроса. |
| http.response.code | float64 | Код состояния HTTP-ответа. |
| http.user\_agent | object | Поле заголовка User-Agent в HTTP-запросах. |
| http.content\_length\_header | float64 | Поле заголовка Content-Length в HTTP-запросах/ответах. |
| http.content\_type | object | Поле заголовка Content-Type в HTTP-запросах/ответах. |
| http.cookie | float64 | Поле заголовка Cookie в HTTP-запросах. |
| http.host | object | Поле заголовка Host в HTTP-запросах. |
| http.referer | float64 | Поле заголовка Referer в HTTP-запросах. |
| http.location | float64 | Поле заголовка Location в HTTP-ответах. |
| http.authorization | float64 | Поле заголовка Authorization в HTTP-запросах. |
| http.connection | object | Поле заголовка Connection в HTTP-запросах. |
| dns.qry.name | float64 | Запрашиваемое доменное имя в DNS-запросах. |
| dns.qry.type | float64 | Тип DNS-запроса (A, AAAA, MX и т.д.). |
| dns.qry.class | float64 | Класс DNS-запроса (обычно IN для Интернета). |
| dns.flags.response | float64 | Указывает, является ли DNS-сообщение ответом. |
| dns.flags.recdesired | float64 | Указывает, желательна ли рекурсия в DNS-запросах. |
| dns.flags.rcode | float64 | Код ответа в DNS-ответах. |
| dns.resp.ttl | float64 | Время жизни в DNS-ответах. |
| dns.resp.len | float64 | Длина DNS-ответа. |
| smtp.req.command | float64 | Команда в SMTP-запросах. |
| smtp.data.fragment | float64 | Фрагмент данных в SMTP. |
| pop.request.command | float64 | Команда в запросах POP3. |
| pop.response | float64 | Ответ в POP3. |
| imap.request.command | float64 | Команда в запросах IMAP. |
| imap.response | float64 | Ответ в IMAP. |
| ftp.request.command | float64 | Команда в запросах FTP. |
| ftp.request.arg | float64 | Аргумент к команде в запросах FTP. |
| ftp.response.code | float64 | Код ответа в FTP. |
| ftp.response.arg | float64 | Аргумент к ответу в FTP. |
| ipv6.src | object | IPv6-адрес источника. |
| ipv6.dst | object | IPv6-адрес назначения. |
| ipv6.plen | float64 | Длина полезной нагрузки в IPv6-пакетах. |
| alert | object | Классификация пакетов как "безопасный" или "подозрительный". |

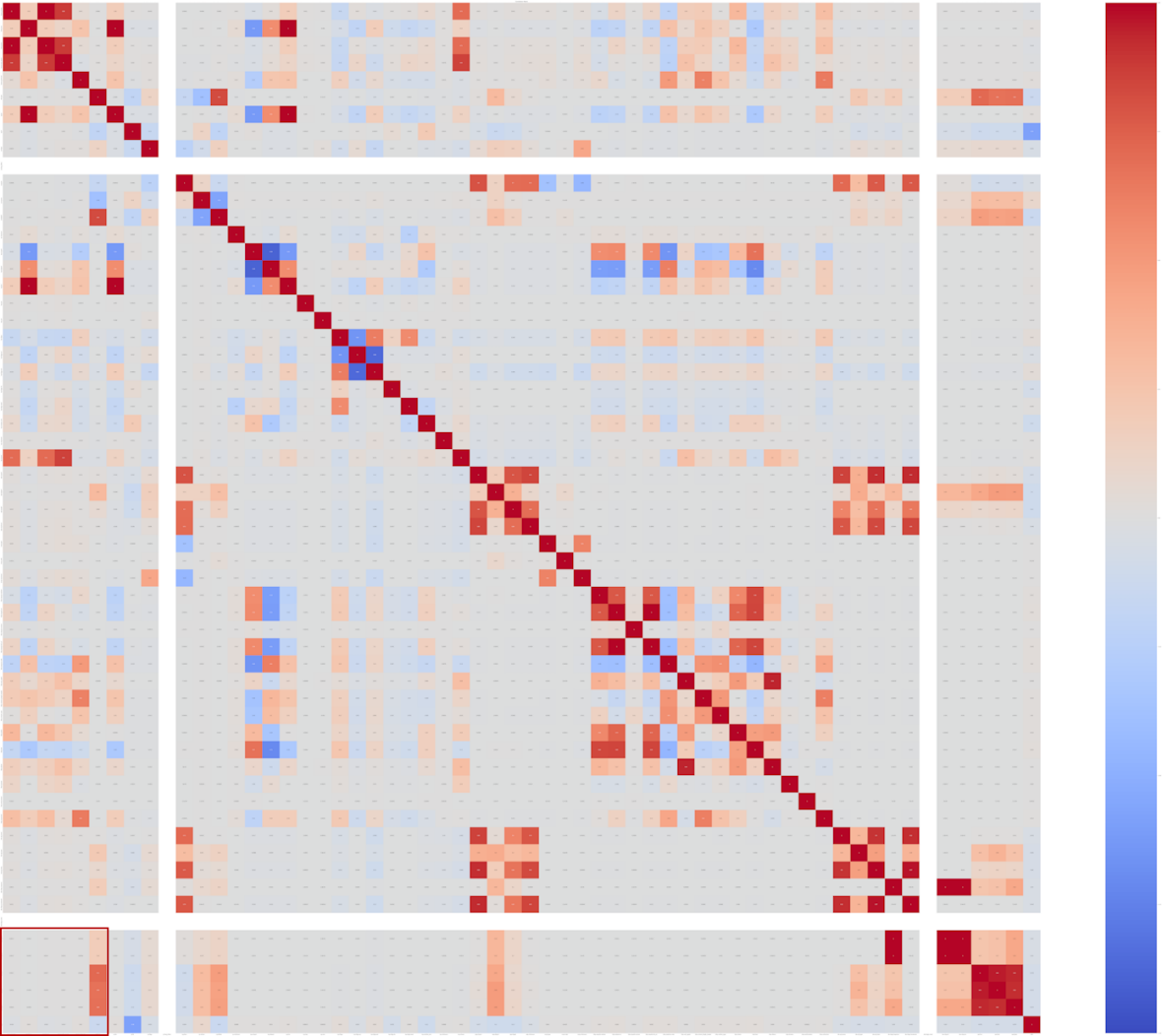
## Подготовка датасетов

Датасет был подготовлен двумя способами [4]. Общим действием для обоих типов подготовки было исключение столбцов eth.src, eth.dst, ip.src и ip.dst, так как подготовленные в лабораторных условиях данные содержат однозначную зависимость подозрительности трафика от источника или получателя трафика, ограниченные малым количеством конечных устройств (3 штуки) и малым количеством отправителей/получателей (до 16 уникальных записей).

Первый датасет («low\_served») был подготовлен так называемо «в лоб»: была произведена очистка столбцов, не содержащих значений. Были закодированы все нечисловые столбцы. Данные были отмасштабированы минимаксом.

Корреляционная матрица первого датасета представлена на рис. 1 [5, 6].

*Рисунок 1.а. Корреляционная матрица low\_served. Общий вид.*



*Рисунок 1.б. Корреляционная матрица low\_served. Увеличенный сегмент.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, диаграмма

Автоматически созданное описание

Второй датасет («high\_served») был подготовлен к работе несколькими методами после изучения его содержимого. В первую очередь были удалены все столбцы, не содержавшие никаких значений (все столбцы категорий ‘smtp.\*’, ‘pop.\*’, ‘imap.\*’, ‘ftp.\*’). Были удалены столбцы общего назначения ‘frame.number’, ‘frame.time’, ‘frame.time\_epoch’.

Столбцы ‘frame.protocols’, ‘http.cookie’, ‘http.content\_type’ содержали массивы данных (перечисления) с высоким количеством уникальных записей, элементы которых имели периодичность повторения (примеры записей столбцов приведены в табл. 2). Для повышения корректности работы алгоритмов было принято решение замены столбцов каждого типа на серии столбцов с маркером присутствия каждого из уникальных вхождений их элементов (метод one-hot-encoding).

*Таблица 2. Примеры записей столбцов-массивов.*

|  |  |  |
| --- | --- | --- |
| **Название столбца** | **Пример записи** | **Пример заменяющих столбцов** |
| frame.protocols | eth:ethertype:ip:tcp | eth, ethertype, ip, tcp |
| http.cookie | acopendivids=swingset,jotto, phpbb2,redmine; acgroupswithpersist=nada | acopendivids, acgroupswithpersist |
| http.content\_type | text/html; charset=iso-8859-1 | text/html, charset |

Следующим шагом была обработка плохо заполненных элементов датасета перед импьютацией. Пакеты некоторых протоколов занимают малую часть общего объёма датасета (ipv6 – 108 записей, dns – 2273 записи и др.). Импьютация их составных частей произвольными, средними или иными конкретными значениями бессмысленна и может нарушить целостность картины (например, для ipv6.src, udp.srcport и др.), в связи с чем было принято решение введение дополнительных столбцов с маркерами признака (содержащие 0, если значение записи отсутствовало до импьютации, и 1, если присутствовало) для всех подобных столбцов, после чего пропущенные значения записей были перезаписаны значением 0.

Для кодирования и масштабирования использовались идентичные методы для всех столбцов за некоторыми исключениями. Уникальные категориальные значения преобразовывались в свои уникальные кодовые значения. Масштабирование производилось классическим делением на максимальное значение столбца.

Исключения составили: группы tcp/udp srcport/dstport (кодирование не производилось; масштабирование по максимально доступному порту – 65535), 'http.request.method' (кодирование и масштабирование производились вручную по всему объёму возможных методов HTTP, 9 типов), 'http.response.code' (кодирование и масштабирование производились вручную по всему объёму возможных кодов HTTP, 61 тип;). Так же исключения составили все искусственные столбцы маркеров признака, полученные в результате one-hot-encoding и маркировки присутствия.

Заключительным этапом подготовки стало составления матрицы корреляции (рис. 2) [5, 6] и пост-обработка датасета на её основе: были исключены столбцы, имеющие высокую (более 0.7) корреляцию друг относительно друга (из пары столбцов выбирался оказывающий большее абсолютное влияние на целевой признак), а также столбцы, имеющие низкую (менее 0.00005) корреляцию с целевым признаком – 18 столбцов (большую часть составили различные искусственные столбцы от one-hot-encoding столбца http.content\_type).

*Рисунок 2.a. Корреляционная матрица high\_served. Общий план.*

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

*Рисунок 2.б. Корреляционная матрица high\_served. Увеличенный сегмент.*

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание

Итоговые виды обоих датасетов приведены в сравнительной таблице 3.

*Таблица 3. Сравнение столбцов* low\_served и high\_served датасетов*.*

|  |  |  |
| --- | --- | --- |
| **low\_served** | **high\_served** | **Комментарий** |
| frame.number | - |  |
| frame.len | frame.len |  |
| frame.time | - |  |
| frame.time\_epoch | - |  |
| frame.protocols | - |  |
| eth.type | - |  |
| ip.len | - |  |
| ip.ttl | ip.ttl |  |
| ip.flags | ip.flags |  |
| ip.frag\_offset | ip.frag\_offset |  |
| ip.proto | - |  |
| ip.version | - |  |
| ip.dsfield | - |  |
| ip.checksum | ip.checksum |  |
| tcp.srcport | - |  |
| tcp.dstport | - |  |
| tcp.len | - |  |
| tcp.seq | tcp.seq |  |
| tcp.ack | tcp.ack |  |
| tcp.flags | tcp.flags |  |
| tcp.flags.syn | tcp.flags.syn |  |
| tcp.flags.ack | - |  |
| tcp.flags.fin | tcp.flags.fin |  |
| tcp.flags.reset | tcp.flags.reset |  |
| tcp.window\_size | tcp.window\_size |  |
| tcp.checksum | tcp.checksum |  |
| tcp.stream | tcp.stream |  |
| udp.srcport | - |  |
| udp.dstport | udp.dstport |  |
| udp.length | - |  |
| udp.checksum | - |  |
| icmp.type | icmp.type |  |
| icmp.code | icmp.code |  |
| icmp.checksum | icmp.checksum |  |
| http.request.method | - |  |
| http.request.uri | http.request.uri |  |
| http.request.version | http.request.version |  |
| http.request.full\_uri | - |  |
| http.response.code | - |  |
| http.user\_agent | http.user\_agent |  |
| http.content\_length\_header | http.content\_length\_header |  |
| http.content\_type | - |  |
| http.cookie | - |  |
| http.host | http.host |  |
| http.referer | - |  |
| http.location | http.location |  |
| http.authorization | http.authorization |  |
| http.connection | http.connection |  |
| dns.qry.name | - |  |
| dns.qry.type | - |  |
| dns.qry.class | dns.qry.class |  |
| dns.flags.response | dns.flags.response |  |
| dns.flags.recdesired | - |  |
| dns.flags.rcode | dns.flags.rcode |  |
| dns.resp.ttl | - |  |
| dns.resp.len | - |  |
| ipv6.src | - |  |
| ipv6.dst | ipv6.dst |  |
| ipv6.plen | - |  |
| alert | alert |  |
| - | browser |  |
| - | ssh |  |
| - | image-jfif |  |
| - | icmpv6 |  |
| - | igmp |  |
| - | tls |  |
| - | tcp |  |
| - | image-gif |  |
| - | xml |  |
| - | png |  |
| - | lldp |  |
| - | ethertype |  |
| - | nbss |  |
| - | imap |  |
| - | mime\_multipart |  |
| - | media |  |
| - | http |  |
| - | x509sat |  |
| - | ip |  |
| - | eth |  |
| - | message-http |  |
| - | llmnr |  |
| - | udp |  |
| - | dhcp |  |
| - | ssdp |  |
| - | ipv6 |  |
| - | frame.protocols\_other |  |
| - | ip.src\_mark | *Маркер присутствия* |
| - | ip.dst\_mark | *Маркер присутствия* |
| - | ip.len\_mark | *Маркер присутствия* |
| - | ip.ttl\_mark | *Маркер присутствия* |
| - | ip.flags\_mark | *Маркер присутствия* |
| - | ip.frag\_offset\_mark | *Маркер присутствия* |
| - | ip.proto\_mark | *Маркер присутствия* |
| - | ip.version\_mark | *Маркер присутствия* |
| - | ip.dsfield\_mark | *Маркер присутствия* |
| - | ip.checksum\_mark | *Маркер присутствия* |
| - | tcp.srcport\_mark | *Маркер присутствия* |
| - | tcp.dstport\_mark | *Маркер присутствия* |
| - | tcp.len\_mark | *Маркер присутствия* |
| - | tcp.seq\_mark | *Маркер присутствия* |
| - | tcp.ack\_mark | *Маркер присутствия* |
| - | tcp.flags\_mark | *Маркер присутствия* |
| - | tcp.flags.syn\_mark | *Маркер присутствия* |
| - | tcp.flags.ack\_mark | *Маркер присутствия* |
| - | tcp.flags.fin\_mark | *Маркер присутствия* |
| - | tcp.flags.reset\_mark | *Маркер присутствия* |
| - | tcp.window\_size\_mark | *Маркер присутствия* |
| - | tcp.checksum\_mark | *Маркер присутствия* |
| - | tcp.stream\_mark | *Маркер присутствия* |
| - | udp.srcport\_mark | *Маркер присутствия* |
| - | udp.dstport\_mark | *Маркер присутствия* |
| - | udp.length\_mark | *Маркер присутствия* |
| - | udp.checksum\_mark | *Маркер присутствия* |
| - | icmp.type\_mark | *Маркер присутствия* |
| - | icmp.code\_mark | *Маркер присутствия* |
| - | icmp.checksum\_mark | *Маркер присутствия* |
| - | http.request.method\_mark | *Маркер присутствия* |
| - | http.request.uri\_mark | *Маркер присутствия* |
| - | http.request.version\_mark | *Маркер присутствия* |
| - | http.request.full\_uri\_mark | *Маркер присутствия* |
| - | http.response.code\_mark | *Маркер присутствия* |
| - | http.user\_agent\_mark | *Маркер присутствия* |
| - | http.content\_length\_header\_mark | *Маркер присутствия* |
| - | http.content\_type\_sfish'",,,keep-alive" |  |
| - | http.content\_type\_ASP.NET\_SessionId |  |
| - | http.content\_type\_application/x-javascript |  |
| - | http.content\_type\_multipart/form-data |  |
| - | http.content\_type\_sfish\\'\\",,,keep-alive" |  |
| - | http.content\_type\_application/xml |  |
| - | http.content\_type\_image/x-icon |  |
| - | http.content\_type\_application/rss+xml |  |
| - | http.content\_type\_charset |  |
| - | http.content\_type\_application/rdf+xml |  |
| - | http.content\_type\_text/plain |  |
| - | http.content\_type\_text/xml |  |
| - | http.content\_type\_application/x-shockwave-flash |  |
| - | http.content\_type\_application/x-www-form-urlencoded |  |
| - | http.content\_type\_text/x-perl |  |
| - | http.content\_type\_sfish\'\",,,keep-alive" |  |
| - | http.content\_type\_application/pdf |  |
| - | http.content\_type\_application/atom+xml |  |
| - | http.content\_type\_text/javascript |  |
| - | http.content\_type\_text/css |  |
| - | http.content\_type\_application/x-asp-net |  |
| - | http.content\_type\_application/x-httpd-php |  |
| - | http.content\_type\_10.20.30.101 |  |
| - | http.content\_type\_nan |  |
| - | http.content\_type\_application/octet-stream |  |
| - | http.content\_type\_other |  |
| - | dvr\_camcnt |  |
| - | b\_id |  |
| - | GALLERYSID |  |
| - | unionname |  |
| - | username |  |
| - | acopendivids |  |
| - | statecode |  |
| - | security |  |
| - | showhints |  |
| - | nan |  |
| - | zap-xss-cookie-basic |  |
| - | security\_level |  |
| - | virtuemart |  |
| - | info-cookie-no-http-only |  |
| - | d5a4bd280a324d2ac98eb2c0fe58b9e0 |  |
| - | jiveAutoLogin |  |
| - | user: |  |
| - | http.cookie\_other |  |
| - | http.host\_mark | *Маркер присутствия* |
| - | http.referer\_mark | *Маркер присутствия* |
| - | http.location\_mark | *Маркер присутствия* |
| - | http.authorization\_mark | *Маркер присутствия* |
| - | http.connection\_mark | *Маркер присутствия* |
| - | dns.qry.name\_mark | *Маркер присутствия* |
| - | dns.qry.type\_mark | *Маркер присутствия* |
| - | dns.qry.class\_mark | *Маркер присутствия* |
| - | dns.flags.response\_mark | *Маркер присутствия* |
| - | dns.flags.recdesired\_mark | *Маркер присутствия* |
| - | dns.flags.rcode\_mark | *Маркер присутствия* |
| - | dns.resp.ttl\_mark | *Маркер присутствия* |
| - | dns.resp.len\_mark | *Маркер присутствия* |
| - | ipv6.src\_mark | *Маркер присутствия* |
| - | ipv6.dst\_mark | *Маркер присутствия* |
| - | ipv6.plen\_mark | *Маркер присутствия* |

# Разработка моделей

Для подготовленных датасетов были созданы несколько моделей с использованием библиотек Numpy [7], Sklearn [8], Tensorflow [9]. Гиперпараметры модели: optimizer – Adam, Loss – Binary Crossentropy. Параметры обучения: 16 эпох, размер батча 48/32.

Первичная модель содержала следующие слои:

* Dense(1024, input\_dim=inp\_size, activation='relu')
* Dropout(0.5)
* Dense(512, activation='relu')
* Dropout(0.5)
* Dense(128, activation='relu')
* Dense(32, activation='relu')
* Dense(1, activation='sigmoid')

Эта и все последующие модели не меняли гиперпараметры. Активаторы слоёв не изменялись. Для компактности записи введём обозначения слоёв (выходной слой одинаков и не включается):

* Dx – Dense, где x – число нейронов слоя (Dense(16) – D4)
* Rx – Dropout, где x – параметр слоя \* 10 (Dropout(0.5) – R5)
* B – BatchNormalization

Результаты экспериментов представлены в сводной таблице 4. Столбец «Модель» содержит описание слоёв модели. Столбец «Потери» содержит долю потерь на тестовой выборке. Столбец «Точность» содержит долю точности на тестовой выборке. Столбец ‘L’ – относится к ‘low\_served’ датасету; столбец ‘H’ – к ‘high\_served’ датасету. Все модели, кроме помеченные символом ‘\*’ обучались с batch size 48; модели со звёздочкой при batch size 32. Детализация обучения содержится на графиках на рис. 3.1–3.6, отражающих изменение тестовых и валидационных метрик точности и потерь сквозь эпохи обучения.

*Таблица 4. Сравнение метрик моделей на тестовой выборке.*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **Модель** | **Потери** | | **Точность** | |
| **L** | **H** | **L** | **H** |
| 1 | D1024 R5 D512 R5 D128 D32 | 0.0432 | 0.0745 | 0.9861 | 0.9758 |
| 2 | D128 R4 D256 R2 D64 D8 | 0.0361 | 0.0731 | 0.9885 | 0.9751 |
| 3 | D128 R4 B D256 R2 B D64 D8 | 0.0478 | 0.1451 | 0.9830 | 0.9436 |
| 4 | D128 D256 D128 D32 D8 | **0.0317** | 0.0356\* | **0.9898** | 0.9885\* |
| 5 | D32 D64 D16 D32 D8 D16 D4 D8 D2 D4 | 0.6931 | 0.0706\* | 0.4995 | 0.9781\* |
| 6 | D230 D412 R4 D412 D192 D72 D27 D3 | 0.0396 | 0.6932\* | 0.9867 | 0.4995\* |

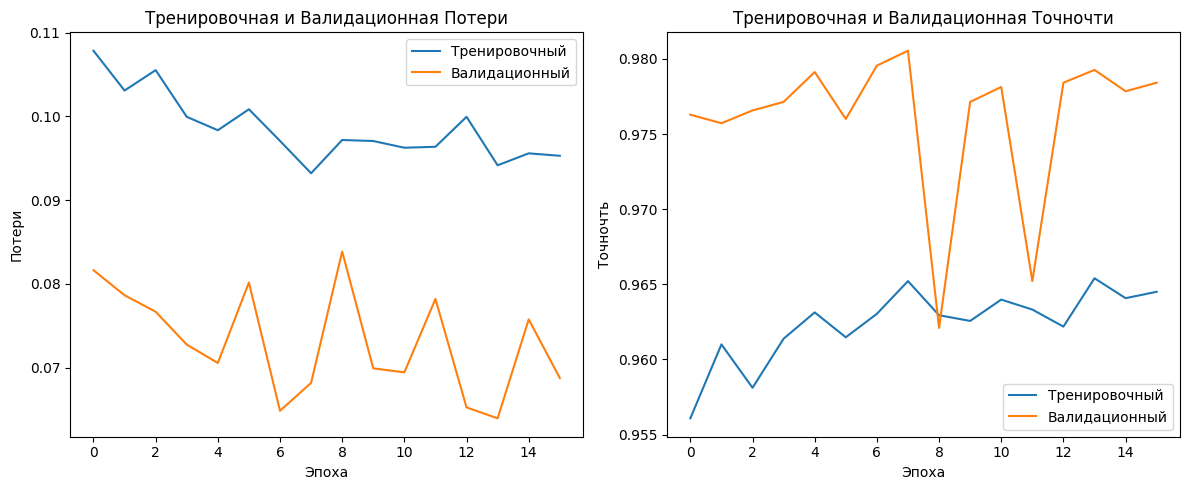
*График 1. Сравнение метрик моделей на тестовой выборке.*

*Рисунок 3.1.а. История обучения модели L1.*

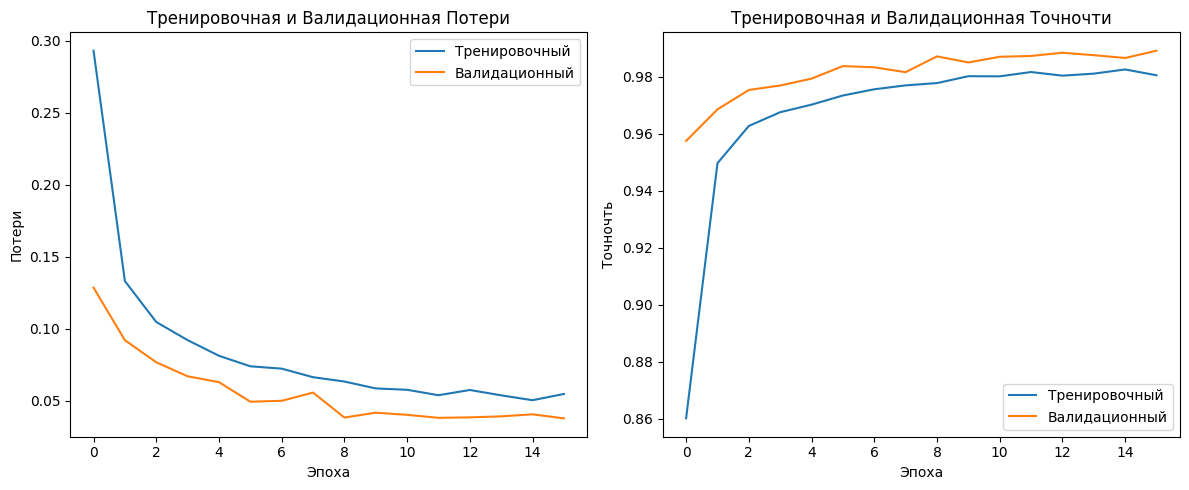
Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

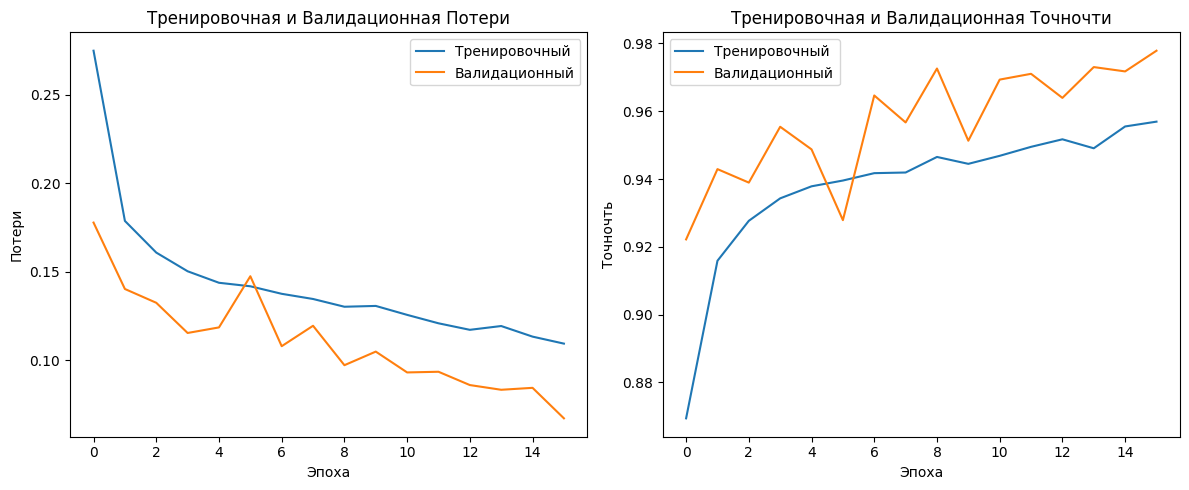
*Рисунок 3.1.б. История обучения модели H1.*



*Рисунок 3.2.а. История обучения модели L2.*



*Рисунок 3.2.б. История обучения модели H2.*



*Рисунок 3.3.а. История обучения модели L3.*

Изображение выглядит как текст, диаграмма, линия, График

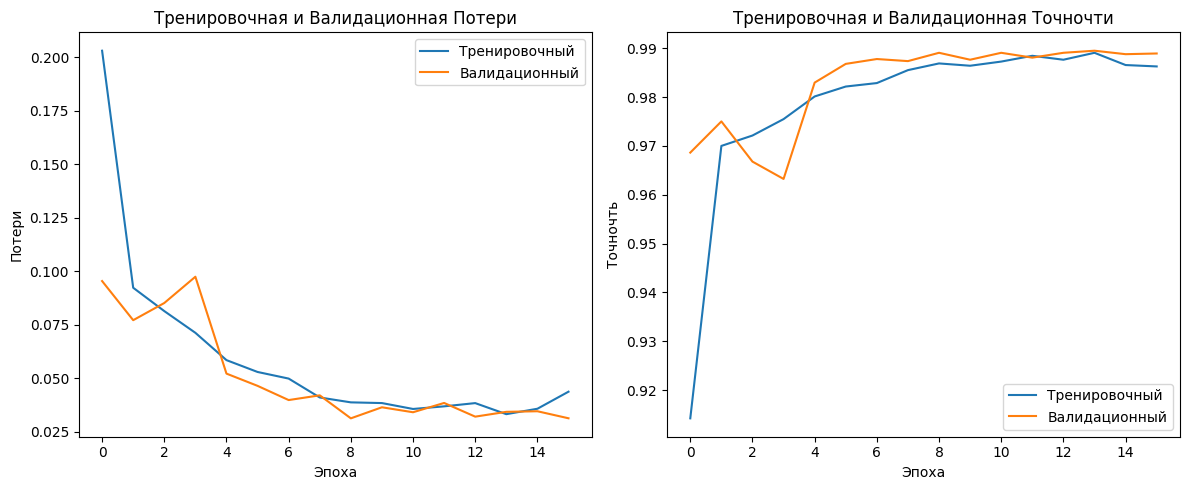
Автоматически созданное описание

*Рисунок 3.3.б. История обучения модели H3.*

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

*Рисунок 3.4.а. История обучения модели L4.*



*Рисунок 3.4.б. История обучения модели H1.*

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

*Рисунок 3.5.а. История обучения модели L5.*

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

*Рисунок 3.5.б. История обучения модели H5.*

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

*Рисунок 3.6.а. История обучения модели L6.*

Изображение выглядит как текст, диаграмма, линия, График

Автоматически созданное описание

*Рисунок 3.6.б. История обучения модели H6.*

Изображение выглядит как текст, линия, диаграмма, График

Автоматически созданное описание

Из приведённых в таблице 1 данных видно, что лучший результат точности показала архитектура 4 на выборке L (‘low\_served’). Следом за ней идут H4 и L2, из которых по показателю потерь немного выигрывает H4. В целом, модели, обученные на первом датасете, смотрятся увереннее своих аналогов на ‘high\_served’. Неодназначную картину показали архитектуры 5 и 6 попарно поочерёдно обвалившие все метрики обоих выборок.

Из графиков, представленных на рисунках 3.1–3.6 видно, что первый датасет ‘low\_served’ показывает большую стабильность обучения. Кроме того, увеличивающаяся сложность архитектуры до определённого момента положительно влияет на рост точности модели при обучении на ‘high\_served’ выборке.

# Проверка моделей

Для оценки реальной применимости моделей была произведена проверка на полном датасете (6.3 млн записей), с большей частью которого модель не была знакома. Также были сняты сняты метрики быстродействия каждой модели (мкс на каждый шаг). Матрицы ошибок каждого из экспериментов представлена на рис. 4.1–4.6. Проверка моделей L5 и H6 не производилась в связи с крайне низким результатами предыдущего эксперимента.

*Таблица 5. Результаты быстродействия моделей.*

|  |  |  |
| --- | --- | --- |
| **Модель** | **L, us/step** | **H, us/step** |
| 1 | >1000 | >1000 |
| 2 | 620 | 600 |
| 3 | 680 | 650 |
| 4 | 680 | 660 |
| 5 | - | 660 |
| 6 | >1000 | - |

*Рисунок 4.1.а. Матрица ошибок модели L1.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, диаграмма

Автоматически созданное описание

*Рисунок 4.1.б. Матрица ошибок модели H1.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

*Рисунок 4.2.а. Матрица ошибок модели L2.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

*Рисунок 4.2.б. Матрица ошибок модели H2.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

*Рисунок 4.3.а. Матрица ошибок модели L3.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, диаграмма

Автоматически созданное описание

*Рисунок 4.3.б. Матрица ошибок модели H3.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

*Рисунок 4.4.а. Матрица ошибок модели L4.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

*Рисунок 4.4.б. Матрица ошибок модели H4.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

*Рисунок 4.5. Матрица ошибок модели H5.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

*Рисунок 4.6. Матрица ошибок модели L6.*

Изображение выглядит как текст, снимок экрана, Прямоугольник, дизайн

Автоматически созданное описание

Анализ матрицы ошибок показывает, что обычный трафик, в целом, определяется достаточно верно. Исходя из поставленных задач наибольший интерес представляет левый нижний элемент матрицы: объём подозрительного сетевого трафика, отмеченного моделью как безвредный («Обычным»). Рисунок 5 отображает данную зависимость для каждой из моделей.

*Рисунок 5. Одобрение подозрительного трафика различными моделями.*

Исходя из данных видно, что лучший результат по фильтрации трафика показала модель L6 (около 52000 пропущенных пакетов, менее 1%). Модель L4 идёт следом (около 71000 пропущенных пакетов, около 1%). Лучшая модель серии H не уступает только моделям L1 и L3: 167 тысяч, 187 тысяч и 178 тысяч пропущенных пакетов. При этом очевидно, что общая эффективность всей серии, несмотря на успешные тестовые результаты, сильно отстаёт от эффективности серии L.

# Анализ результатов и сравнение моделей

В ходе группы экспериментов были получены значения точностей и потерь обучения, пропуска подозрительного трафика на полной выборке, а также быстродействия различных моделей для различных датасетов. Итоговые результаты для моделей L4, L6, H3, H4 и H5 представлены в сводной таблице 6.

*Таблица 6. Сводная таблица результатов экспериментов.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Модель | Loss, % | Accuracy, % | Пропуск, шт | Скорость, мкс/шаг |
| L4 | 3.17 | 98.98 | 70929 | 680 |
| H4 | 3.56 | 98.85 | 196015 | 660 |
| L6 | 3.96 | 98.67 | 51912 | >1000 |
| H5 | 7.06 | 97.81 | 168626 | 660 |
| H3 | 14.51 | 94.36 | 144949 | 650 |

Выделить однозначно превосходящую модель достаточно проблематично: каждая из моделей обладает рядом преимуществ и недостатков по сравнению с остальным. Однако некоторый баланс достигается у модели L4: самый высокий показатель точности модели, самый низкий показатель потерь, вторая позиция по рейтингу пропусков подозрительного трафика и сопоставимое с остальными моделями быстродействие.

Также на основе экспериментов можно заключить, что модели серии H, обученные на датасете ‘high\_served’ обладают значительно меньшими показателями относительно их соответствующих аналогов относительно моделей серии L, обученных на датасете ‘low\_served’. Обобщённый вывод данного явления может быть сформулирован в виде гипотезы, что данный датасет был переподготовлен и его дополнительные преобразования являются избыточными. Данное утверждение не может быть подтверждено проведёнными экспериментами и требует дополнительных исследований.

# ЗАКЛЮЧЕНИЕ

В ходе научно-исследовательской работы был проведен анализ кейса «Поиск аномалий в сетевом трафике», была обучена модель для прогнозирования аномальных сессий в сетевом трафике. Результаты задач, поставленных перед работой:

1. Был проанализирован исходный датасет. Датасет был подготовлен 2 способами, изменившими общую структуру данных.
2. В общей сложности было разработано 12 моделей: 6 архитектур, обученных на 2 разных датасетах. Оценены их быстродействие и точность.
3. Было произведено сравнение ключевых показателей всех 12 моделей, выявлены наиболее подходящие для решение целевой задачи модели.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Attack Simulation Lab [Электронный ресурс] // Режим доступа: https://www.kaggle.com/datasets/moradrawashdeh/attack-simulation-lab (Дата обращения: 14.03.2024).
2. Шыхалиев Р.Г., «Анализ и классификация сетевого трафика компьютерных сетей» // Институт Информационных Технологий НАНА – Баку, 2010. – С.15–23.
3. А. И. Гетьман, М. К. Иконникова, «Обзор методов классификации сетевого трафика с использованием машинного обучения» // Труды ИСП РАН. – 2020. – С.137–154.
4. Pandas Official Documentation [Электронный ресурс] // Режим доступа: https://pandas.pydata.org/docs/ (Дата обращения: 03.05.2024).
5. Matplotlib: Visualization with Python Official Documentation [Электронный ресурс] // Режим доступа: https://matplotlib.org/stable/users/index.html (Дата обращения: 03.05.2024)
6. Seaborn: Statistical Data Visualization Official Documentation [Электронный ресурс] // Режим доступа: https://seaborn.pydata.org/tutorial.html (Дата обращения: 03.05.2024)
7. Numpy Official Documentation [Электронный ресурс] // Режим доступа: https://numpy.org/doc/ (Дата обращения: 03.05.2024)
8. Sklearn User Guide [Электронный ресурс] // Режим доступа: https://scikit-learn.org/stable/user\_guide.html (Дата обращения: 03.05.2024)
9. Tensorflow Official Documentation [Электронный ресурс] // Режим доступа: https://www.tensorflow.org/api\_docs (Дата обращения: 03.05.2024)