1. Министерство образования и науки Российской Федерации
2. Санкт-Петербургский Политехнический Университет Петра Великого
3. —
4. Институт кибербезопасности и защиты информации

**НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ РАБОТА**

**«Определение главных компонент при построении нейронной сети, определяющей сетевые атаки»**

1. Выполнила
2. студентка гр. 4851003/90801 Кулеева А.Г.

1. Руководитель
2. к.т.н. Платонов В.В.
4. Санкт-Петербург
5. 2022
6. Содержание

|  |  |
| --- | --- |
| [Введение](#_Введение)........................................................................................................... | 3 |
| 1. [Методы](#_Теоретические_сведения) машинного обучения................................................................. | 4 |
| * 1. [Метод главных компонент](#_Метод_главных_компонент)..................................................................... | 4 |
| * 1. [Классификация сетевых атак](#_Классификация_сетевых_атак).................................................................. | 4 |
| * 1. [Основные параметры обучаемой системы](#_Основные_параметры_обучающего)............................................ | 5 |
| 1. [Программная реализация метода](#_Реализация_метода_главных).............................................................. | 9 |
| [Заключение](#_Заключение)...................................................................................................... | 10 |
| [Список использованных источников](#_Список_использованных_источников)............................................................. | 11 |

# Введение

На сегодняшний день ни одно предприятие не обходится без компьютерных технологий для обслуживания своей инфраструктуры. Даже малый бизнес нуждается в кассовом аппарате, терминале оплаты, а это уже сеть, хоть и небольшая. Если крупные предприятия могут позволить себе полноценные и дорогостоящие средства защиты, наполненные всевозможными брандмауэрами, средствами обнаружения вторжений, снифферами и тому подобными, то более мелкие предприятия не обладают ресурсами для реализации данной модели защиты. Встает вопрос о создании такой системы обнаружения вторжений, которая могла бы обнаруживать наличие атаки, а также потенциально относить её к некоторому классу, обладая при этом минимальным оптимальным числом параметров.

При обнаружении атак в сетевых инфраструктурах могут использоваться различные наборы параметров. На сегодняшний день самый полный список параметров предоставляет датасет UNSW-NB15[1]. Данный датасет предоставляет всего 47 параметров сети, которые учитывались во время эксперимента.

Целью работы является сокращение количества параметров методом главных компонент, при этом информативность должна быть не менее 95%.

# Методы машинного обучения

## Метод главных компонент

Метод главных компонент является одним из алгоритмов сокращения размерности обрабатываемых данных с наименьшей потерей информации в процессе машинного обучения. Изобретен Карлом Пирсоном в 1901 году [2].

Суть метода заключается в следующем. Строится матрица, где каждая строка ― выборка, то есть числовая характеристика некоторой компоненты. Далее строится ковариационная матрица. Она является обобщением дисперсии на случай многомерных случайных величин и описывает разброс случайной величины так же, как и дисперсия. Диагональные элементы ковариационной матрицы показывают дисперсии по изначальному базису, а ее собственные значения ― по новому (по главным компонентам). Далее необходимо найти собственный вектор матрицы. При перемножении собственного вектора на матрицу ковариации будет получен новый вектор ― проекция, содержащий в себе значения главной компоненты

## Классификация сетевых атак

Исследование UNSW-NB15[1] предлагает следующие состояния сети:

1. Нормальное состояние с естественным потоком данных.
2. Фаззеры ― попытка заставить программу или сеть приостановиться путем подачи ей случайно сгенерированных данных.
3. Анализ ― содержит различные атаки, такие как сканирование портов, спам и проникновение в html-файлы.
4. Бэкдоры ― техника, при которой механизм безопасности системы незаметно обходится для доступа к компьютеру или его данным.
5. DoS ― злонамеренная попытка сделать сервер или сетевой ресурс недоступным для пользователей, обычно путем временного прерывания или приостановки работы узла, подключенного к Интернету.
6. Эксплойты ― злоумышленник знает о проблеме безопасности в операционной системе или части программного обеспечения и эксплуатирует эту уязвимость.
7. Общий ― техника работает против всех блокшифров (с заданным блоком и ключом размера), без учета структуры блок-шифра.
8. Разведывательный ― содержит все способы имитации атак, которые собирают информацию.
9. Shellcode ― небольшой фрагмент кода, используемый в качестве полезной нагрузки при эксплуатации уязвимости программного обеспечения.
10. Черви ― злоумышленник реплицирует себя для того, чтобы распространиться на другие компьютеры.

## Основные параметры обучающего множества

При анализе сгенерированного сетевого трафика учитывалось 47 параметров. Поскольку необходимо сократить их количество, некоторые параметры исключим самостоятельно. Например, такие данные, как ip-адреса и номер портов источника и назначения не имеют значения при обнаружении атаки. Также можно не учитывать протокол, его статус и сервис (https, dns, ssh, и т.д.). Таким образом, удалось сократить количество исследуемых параметров до 40. Все они представлены в таблицах ниже.

Замечание: Tип I: integer, F: float, T: timestamp and B: binary

Таблица 1 ― Базовые параметры

| № | Имя | Тип | Описание |
| --- | --- | --- | --- |
| 1 | dur | F | Общая продолжительность записи |
| 2 | sbytes | I | Количество байт от источника к назначению |
| 3 | dbytes | I | Количество байт от назначения к источнику |
| 4 | sttl | I | Время жизни пакета от источника к назначению |
| 5 | dttl | I | Время жизни пакета от назначения к источнику |
| 6 | sloss | I | Повторно переданные или потерянные пакеты источника |
| 7 | dloss | I | Повторно переданные или потерянные пакеты назначения |
| 8 | sload | F | Загруженность источника (количество бит в секунду) |
| 9 | dload | F | Загруженность назначения |
| 10 | spkts | I | Количество пакетов от источника к получателю |
| 11 | dpkts | I | Количество пакетов от получателя к источнику |

Таблица 2 ― Параметры по содержанию

| № | Имя | Тип | Описание |
| --- | --- | --- | --- |
| 12 | swin | I | Размер окна TCP источника |
| 13 | dwin | I | Размер окна TCP назначения |
| 14 | stcpb | I | Порядковый номер TCP источника |
| 15 | dtcpb | I | Порядковый номер TCP назначения |
| 16 | smeansz | I | Средний размер пакета потока, передаваемого источником |
| 17 | dmeansz | I | Средний размер пакета потока, передаваемого получателем |
| 18 | trans\_depth | I | Глубина соединения транзакции http запрос/ответ |
| 19 | res\_bdy\_len | I | Размер содержимого данных, передаваемых http-сервисом сервера. |

Таблица 3 ― Временные параметры

|  |  |  |  |
| --- | --- | --- | --- |
| № | Имя | Тип | Описание |
| 20 | sjit | F | Джиттер источника (мСек) |
| 21 | djit | F | Джиттер назначения (мСек) |
| 22 | stime | T | Время начала записи |
| 23 | ltime | T | Время последней записи |
| 24 | sintpkt | F | Время прибытия межпакетного потока источника (мСек) |
| 25 | dintpkt | F | Время прибытия межпакетного потока назначения (мСек) |
| 26 | tcprtt | F | Сумма 'synack' и 'ackdat' в TCP. |
| 27 | synack | F | Время между пакетами SYN и SYN\_ACK в TCP. |
| 28 | ackdat | F | Время между пакетами SYN\_ACK и ACK в TCP. |

Таблица 4 ― Дополнительно сгенерированные параметры

| № | Имя | Тип | Описание |
| --- | --- | --- | --- |
| Параметры общего назначения | | | |
| 29 | is\_sm\_ips\_ports | B | Если IP-адреса источника и назначения и номера портов равны, эта переменная принимает значение 1, в противном случае 0 |
| 30 | ct\_state\_ttl | I | Число для каждого состояния в соответствии с определенным диапазоном значений для времени жизни источника/назначения |
| 31 | ct\_flw\_http\_mthd | I | Количество потоков, которые имеют такие методы, как Get и Post в службе http. |
| 32 | is\_ftp\_login | B | Если доступ к ftp-сессии осуществляется по имени пользователя и паролю, то 1, иначе 0. |
| 33 | ct\_ftp\_cmd | I | Количество потоков, имеющих команду в ftp-сессии. |
| Параметры соединений | | | |
| 34 | ct\_srv\_src | I | Количество соединений, содержащих один и тот же сервис и адрес источника в 100 соединениях по данным последнего времени. |
| 35 | ct\_srv\_dst | I | Количество соединений, содержащих один и тот же сервис и адрес назначения в 100 соединениях по данным последнего времени. |
| 36 | ct\_dst\_ltm | I | Количество соединений с одинаковым адресом назначения в 100 соединениях в соответствии с последним временем. |
| 37 | ct\_src\_ltm | I | Количество соединений с одинаковым адресом источника в 100 соединениях за последнее время. |
| 38 | ct\_src\_dport\_ltm | I | Количество соединений с одинаковым адресом источника и портом назначения в 100 соединениях в соответствии с последним временем. |
| 39 | ct\_dst\_sport\_ltm | I | Количество соединений с одинаковым адресом назначения и портом источника в 100 соединениях в соответствии с последним временем. |
| 40 | ct\_dst\_src\_ltm | I | Количество соединений с одинаковыми адресами источника и назначения в 100 соединениях по данным последнего времени. |

# Программная реализация метода

Из датасета UNSW-NB15 [4] были взяты записи и на основе них построены выборки размера 10000 для каждой компоненты. В языке программирования Python метод главных компонент уже реализован в библиотеке sklearn [3]. Необходимо задать, до какого числа компонент будет снижена размерность, и передать методу матрицу значений. В качестве возвращаемого значения будут получены собственные вектора с максимальными собственными числами.

Необходимо выделить столько главных компонент, чтобы они охватывали 95% информации. Формула для вычисления представлена ниже:

Для получения этих данных воспользуемся методом explained\_variance\_ratio\_ того же класса PCA библиотеки sklearn [3]. Как видно из рисунка 1, при выделении всего 2 компонент удалось сохранить 99,7% информации. Однако такой результат считается аномальным. При сопоставлении имён компонент их индексам оказалось, что главными компонентами являются параметры stcpb и dtcpb, которые обозначают порядковый номер TCP источника и назначения соответственно [5, 7]. Порядковому номеру (Sequence number) соответствуют биты 32-63. Поле порядкового номера используется для установки номера в каждом TCP-пакете, чтобы можно было правильно упорядочить поток TCP (например, пакеты приводятся к правильному порядку) [6].

Изображение выглядит как текст

Автоматически созданное описание

Рисунок ― Снижение размерности

Было решено удалить данные параметры из датасета. Однако вновь были получены всего 2 компоненты, что очень мало из 40 исходных. Опытным путем были удалены следующие параметры: загруженность источника и назначения (количество бит в секунду), количество байт от источника к назначению и от назначения к источнику, размер содержимого данных, передаваемых http-сервисом сервера, джиттер источника и назначения, время прибытия межпакетного потока источника и назначения (мСек). Таким образом, было исключено ещё 11 параметров, после чего их осталось всего 29.

Теперь алгоритм выделяет 5 главных компонент, которые дают 98% информации (рисунок 2). Это лучший результат, поскольку 4 компоненты дают лишь 93% информации. Но при этом четвертая и пятая компоненты имеют одно и тоже имя, но разные значения. В ходе поиска данной ошибки в сети Интернет было выяснено, что преобразование PCA не дает гарантии, что входные данные будут разными. Такая функция вносит наибольший вклад, поэтому получаются дублирующиеся компоненты [8]. Таким образом, данный дубликат игнорируется. Тогда общий вклад в информацию будет составлять лишь 93%, что мало. Попробуем получить 95% информативности.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок ― Выделение главных компонент

Если выбрать 8 компонент (по факту 7, один дубликат), то получим ровно 95% информации (рисунок 3).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок ― Повышение информативности

Итак, главными компонентами в данном случае являются: средний размер пакета потока (как отправителя, так и получателя), время начала записи, количество пакетов от источника к получателю и от получателя к источнику, ttl пакетов отправителя и получателя.

# Заключение

В ходе выполнения данной работы были выделены главные компоненты, которые влияют на наличие атаки в сетевой инфраструктуре. Для снижения размерности данных использовался метод главных компонент. Сначала вручную были исключены компоненты, неподдающиеся анализу, либо генерирующие аномальные результаты. В итоге осталось 29 компонент. Из рассматриваемых 29 компонент только 7 несут в себе 95% информации, остальные 22 можно считать шумом. Наибольший вес среди них имеют средний размер пакета потока, время начала записи и количество пакетов от источника к получателю. Они составляют 93% информации. Все последующие компоненты вносят не более 1% информации.

В дальнейшем на основе этих данных может быть написана нейронная сеть, которая будет определять наличие атаки, а также относить атаку к определённому классу.

# Список использованных источников

1. UNSW-NB15: A Comprehensive Data set for Network  
   Intrusion Detection systems – 2015.
2. Метод главных компонент (PCA) [Электронный ресурс]. URL: <https://neerc.ifmo.ru/wiki/index.php?title=Метод_главных_компонент_(PCA)>. – (дата обращения: 12.06.2022).
3. Как работает метод главных компонент (PCA) на простом примере [Электронный ресурс]. URL: <https://habr.com/ru/post/304214/> . – (дата обращения: 12.06.2022).
4. The UNSW-NB15 Dataset [Электронный ресурс]. URL: <https://research.unsw.edu.au/projects/unsw-nb15-dataset>. – (дата обращения: 15.06.2022).
5. [sklearn.decomposition](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.decomposition).PCA [Электронный ресурс]. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html#sklearn.decomposition.PCA.get_feature_names_out>. – (дата обращения: 26.06.2022).
6. Транспортные протоколы TCP и UDP [Электронный ресурс]. URL: [https://hackware.ru/?p=12935](https://hackware.ru/?p=12935#:~:text=Порядковый%20номер%20(Sequence%20number)%20—,пакеты%20приводятся%20к%20правильному%20порядку)). – (дата обращения: 26.06.2022).
7. [Recovering features names of explained\_variance\_ratio\_ in PCA with sklearn](https://stackoverflow.com/questions/22984335/recovering-features-names-of-explained-variance-ratio-in-pca-with-sklearn) [Электронный ресурс]. URL: <https://stackoverflow.com/questions/22984335/recovering-features-names-of-explained-variance-ratio-in-pca-with-sklearn>. – (дата обращения: 26.06.2022).
8. [Why PCA output some components duplicately?](https://stackoverflow.com/questions/67769996/why-pca-output-some-components-duplicately) [Электронный ресурс]. URL: <https://stackoverflow.com/questions/67769996/why-pca-output-some-components-duplicately>. – (дата обращения: 26.06.2021).