МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**КУРСОВАЯ РАБОТА**

**РЕАЛИЗАЦИЯ СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.Э. Айрапетов

(подпись)

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем  курс    3

Направленность Технология программирования

Научный руководитель

д-р физ.-мат. наук А.И. Миков

(подпись)

Нормоконтролер

доц. А.В. Харченко

(подпись)

Краснодар

2023

**РЕФЕРАТ**

Курсовая работа содержит 31 страницу, 17 рисунков, 5 источников.

ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ, СИСТЕМЫ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ, МАШИННОЕ ОБУЧЕНИЕ, SCIKIT-LEARN.

Объектом исследования являются системы обнаружения вторжений и применение методов машинного обучения в решении задач кибербезопасности, что является важным направлением в современной области развития решений по обеспечению информационной безопасности.

Целью данной курсовой работы является исследование и анализ эффективности применения методов машинного обучения, в частности, с использованием библиотеки scikit-learn, в системах обнаружения вторжений.

Работа направлена на изучение особенностей различных методов обнаружения вторжений, оценку их производительности на различных наборах данных, а также на выявление преимуществ и недостатков каждого метода. Основной задачей было провести эксперименты с моделями машинного обучения, такими как, например, деревья решений, изоляционные леса и автоэнкодеры, в контексте задачи обнаружения вторжений.

В результате выполнения курсовой работы были получены оценки качества применения методов машинного обучения в данной области на двух наборах данных. Сформированные выводы могут послужить практическим руководством для выбора и настройки моделей обнаружения вторжений в зависимости от характеристик конкретной задачи и набора данных.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc153369818)

[1 Обзор предметной области 5](#_Toc153369819)

[1.1 Задача обнаружения вторжений 5](#_Toc153369820)

[1.2 Подходы к решению задачи 9](#_Toc153369821)

[1.2.1 Обучение с учителем 10](#_Toc153369822)

[1.2.2 Обучение без учителя 12](#_Toc153369823)

[1.3 Работа с данными 13](#_Toc153369824)

[1.4 Метрики оценки качества 16](#_Toc153369825)

[2 Проведение экспериментов 19](#_Toc153369826)

[ЗАКЛЮЧЕНИЕ 30](#_Toc153369827)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 31](#_Toc153369828)

# ВВЕДЕНИЕ

В современном информационном обществе, где технологические достижения растут в геометрической прогрессии, вопросы кибербезопасности становятся все более актуальными. Один из ключевых аспектов в этой области – обнаружение вторжений, которое представляет собой сложную задачу в условиях постоянно эволюционирующих атак.

На сегодняшний день широкое применение находят алгоритмы машинного обучения, а также нейронные сети, которые благодаря своей способности к извлечению сложных закономерностей из данных, могут быть использованы в задаче анализа трафика сети для выявления угроз. Однако, остаются открытыми некоторые вопросы об адаптации этих моделей к новым, ранее неизвестным атакам, для которых они не обучались.

Целью данной работы является исследование проблем реализации систем обнаружения вторжений в условиях отсутствия обучающих данных обо всех возможных атаках. Основное внимание уделяется адаптивным механизмам обучения, позволяющим модели успешно справляться с новыми угрозами, несмотря на отсутствие предварительного опыта обучения.

В ходе исследования рассматриваются следующие аспекты: выбор данных для обучения модели, построение эффективных моделей обнаружения вторжений, а также оценка точности и надежности различных подходов к обучению в условиях неизвестных атак.

Это исследование имеет важное практическое значение, поскольку позволяет разработать более гибкие и адаптивные системы обнаружения вторжений, способные эффективно функционировать в постоянно меняющемся мире.

# 1 Обзор предметной области

## 1.1 Задача обнаружения вторжений

Обнаружение вторжений (Intrusion Detection) представляет собой одну из ключевых задач в области информационной безопасности, направленную на выявление несанкционированных активностей в компьютерных системах. Современные технологии создают сложные сетевые среды, где защита от киберугроз становится все более актуальной.

Кибератаки можно классифицировать в зависимости от действий и целей злоумышленника. Каждый тип атаки можно отнести к одному из следующих четырех классов [1]:

* атаки типа «отказ в обслуживании» (DoS) направлены на блокирование или ограничение услуг, предоставляемых сетью, компьютером пользователям;
* зондирующие атаки направлены на получение информации о сети или компьютерной системе;
* атаки типа User-to-Root (U2R) направлены на получение непривилегированным пользователем доступа root или admin на конкретном компьютере или системе, к которой злоумышленник имел доступ на уровне пользователя;
* атаки Remote-to-Local (R2L) предполагают отправку пакетов на компьютер жертвы. Киберпреступник узнает о действиях пользователя и получает привилегии, которые мог бы иметь конечный пользователь в компьютерной системе.

В рамках этих категорий существует множество различных форм компьютерных атак.

Система обнаружения вторжений (IDS) представляет собой комплекс программных или программно-аппаратных средств, направленных на выявление несанкционированных попыток доступа, атак и других вредоносных действий в информационных системах. Основная задача IDS – обнаруживать аномалии или шаблоны поведения, которые могут свидетельствовать о наличии вторжения.

IDS широко применяются в корпоративных сетях, системах управления, облачных платформах и других сферах, где безопасность информации играет ключевую роль. На рисунке 1 представлена упрощенная схема сети, в которой установлена система обнаружения вторжений.

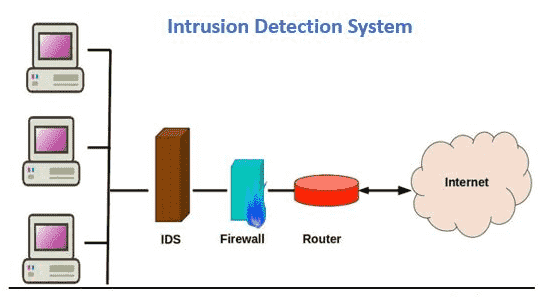


Рисунок 1 – Схема сети с установленной IDS

Системы обнаружения вторжений делятся на два основных типа: сигнатурные и аномалийные:

* IDS на основе сигнатур отслеживают входящий сетевой трафик, обнаруживая определенные шаблоны и последовательности, которые соответствуют известным сигнатурам атак. Несмотря на свою эффективность, она не способна обнаруживать неидентифицированные атаки без известных шаблонов.
* IDS на основе аномалий – это относительно новая технология, предназначенная для обнаружения неизвестных атак и не ограничивающаяся определением сигнатур атак. Этот тип обнаружения использует машинное обучение для анализа больших объемов сетевых данных и трафика.

Обычно IDS используются только для идентификации и создания «события», которое передается в средства сбора и анализа информации о событиях безопасности (SIEM) аналитикам безопасности. Тем не менее, IDS можно использовать и напрямую. На рисунке 2 изображён менеджер событий EveBox системы обнаружения вторжений с открытым исходным кодом Suricata.



Рисунок 2 – Пример идентификации IDS различных событий

Системы обнаружения могут иметь некоторые недостатки:

* статичность. Большинство классических IDS основаны на сигнатурном подходе, использующем правила и шаблоны для определения известных атак. Это означает, что они ориентированы на предварительно известные угрозы и могут быть неэффективны в случае новых, ранее неизвестных атак или техник обхода;
* ложные срабатывания. Классические IDS часто сталкиваются с проблемой высокого уровня ложных срабатываний. Это происходит из-за того, что сигнатурные методы могут реагировать на обычные или нормальные сетевые активности, интерпретируя их как потенциальные атаки. Это усложняет работу сотрудников по безопасности, так как им приходится фильтровать большой объем ложных предупреждений;
* сложность поддержки. Настройка и поддержка классических IDS могут быть трудоемкими процессами. Постоянное обновление сигнатур, определение правил и обработка новых атак требуют высокой квалификации и ресурсов, что делает их использование сложным для многих организаций.

Для решения этих проблем могут использоваться методы машинного обучения, что может позволить IDS более гибко адаптироваться к новым угрозам и уменьшить количество ложных срабатываний. Модели машинного обучения могут обучаться на исторических данных и выявлять необычные или подозрительные события.

Однако всё ещё существует проблема обобщения моделей на новые типы атак, которые могут не входить в обучающий набор данных.

Также существует проблема, заключающаяся в различии структуры компьютерных сетей. Если модель, обученную на трафике одной сети, применить для распознавания вторжений в другой сети, с высокой долей вероятности будут часто происходить ложные срабатывания или, что еще хуже, будет игнорироваться вредоносный трафик. Это может быть связано с различием, прежде всего, в пропускной способности сети или общей скоростью передачи данных. Также проблема может крыться в том, что модель в своем обучении учитывала IP-адреса отправителя и адресата, соответственно, в другой сети она просто не смогла применить свои «знания». Ещё одна проблема, которая может возникнуть при использовании машинного обучения – переобучение. При адаптации модели к постоянно изменяющимся атакам и растущим угрозам, могут быть достигнуты, так называемые, «тупики в развитии». Модель может изменить свои параметры в сторону чрезмерной специализации на конкретных данных. Это явление представляет собой серьезное ограничение при применении машинного обучения в задаче обнаружения вторжений.

Выбор и оптимизация параметров модели, а также выбор хороших данных для обучения, обработка и анализ этих данных, играют важную роль в эффективности системы обнаружения вторжений. Разработка более точных и устойчивых моделей требует глубокого понимания характера киберугроз, а также непрерывного мониторинга и анализа новых угроз, чтобы адаптировать систему к быстро меняющемуся киберпространству.

## 1.2 Подходы к решению задачи

В этой работе будут рассмотрены некоторые из наиболее применимых методов машинного обучения в задаче распознавания вредоносного трафика. Для распознавания потенциально опасного трафика могут быть применены 2 общих подхода.

Первый и самый очевидный – классификация. Мы можем рассматривать приходящий трафик как объекты из разных классов и распределять их в соответствии со знаниями, полученными из обучения. Однако, этот метод не учитывает возникновение новых, ранее не известных, атак. Существует вероятность что атака будет похожа на одну из тех, на которых обучалась модель, но это зависит от множества факторов, таких как, объём обучающей выборки и её репрезентативность, частота появления новых атак и т. д. Даже если брать в расчёт возможность обновления модели, эта операция может быть дорогостоящей: в условиях постоянно растущих угроз, новые виды атак могут появляться каждый день и, соответственно, наша обучающая выборка и модель могут стать довольно «тяжелыми».

В связи с этим, было принято решение сузить задачу до бинарной классификации. Модели бинарной классификации могут обобщить данные атак и, при дообучении на новых данных, скорректировать закономерности в определении атак. Тем не менее, при постоянной доработке, модели могут упереться в потолок своего развития и вовсе потерять свою пользу. Чтобы избежать такого исхода нужно ответственно подходить к формированию наборов данных и настройке методов.

Второй подход – определение аномалий. Существует ряд методов обучения без учителя, которые своей целью ставят изучение «нормальных» данных, дабы при работе в реальных условиях замечать любые отклонения. Этот подход может оказаться наиболее подходящим, так как он не подразумевает наличие атак в обучающей выборке. Тем не менее, эти методы обычно требуют очень тонкой настройки гиперпараметров при каждой итерации обновления модели.

Далее рассмотрим наиболее перспективных представителей описанных подходов и оценим их применимость к нашей задаче. Большая часть моделей была рассмотрена в рамках реализации библиотеки scikit-learn [2].

### 1.2.1 Обучение с учителем

*Метод опорных векторов* (Support Vector Machines — SVM) — это набор контролируемых методов обучения, используемых для классификации, регрессии. Основная идея метода заключается в построении гиперплоскости, разделяющей объекты выборки оптимальным способом. Из преимуществ можно выделить эффективность в пространствах больших размеров и в случае небольших выборок. К недостаткам опорных векторных машин можно отнести то, что они не предоставляют напрямую оценки вероятностей, они рассчитываются с использованием дорогостоящей пятикратной перекрестной проверки.

*Логистическая регрессия*, несмотря на свое название, представляет собой скорее линейную модель классификации, чем регрессию. В этой модели вероятности, описывающие возможные результаты одного испытания, моделируются с использованием логистической функции. Модель хорошо справляется с задачами бинарной классификации, особенно в случаях, когда классы линейно разделимы. Также модель довольно быстро обучается даже на больших объемах данных. Однако, ограничение линейной зависимости может быть недостаточным для моделирования сложных взаимосвязей в данных.

*Деревья решений* (Decision Tree) — это метод обучения, используемый для классификации и регрессии, цель которого состоит в том, чтобы создать модель, которая предсказывает значение целевой переменной, изучая простые правила принятия решений, выведенные из характеристик данных. Для модели не требуется серьезная подготовка данных, а также стоимость использования дерева является логарифмической по количеству точек данных, используемых для обучения дерева. Тем не менее, модель способна создать слишком сложное дерево, что может привести к переобучению. Помимо этого, небольшие изменения в данных могут привести к созданию совершенно другого дерева.

*Случайный лес решений* (Random Forest) – это метод ансамблевого обучения для классификации, регрессии и других задач, который работает путем построения множества деревьев решений во время обучения. Для задач классификации результатом случайного леса является класс, выбранный большинством деревьев. Случайные леса решений исправляют привычку деревьев решений чрезмерно подстраиваться под обучающий набор, а также могут быть хорошо распараллелены. Однако при построении модели требуется O(K) памяти, где K — число деревьев.

*Наивные методы Байеса* — это набор алгоритмов контролируемого обучения, основанных на применении теоремы Байеса с «наивным» предположением об условной независимости между каждой парой характеристик при заданном значении переменной класса. Модель хорошо обучается и работает на небольших объемах данных, однако чувствительна к выбросам, что может снизить ее производительность.

*Методы ближайших соседей* (k-Nearest Neighbors). Принцип, лежащий в основе методов ближайших соседей, состоит в том, чтобы найти предопределенное количество обучающих выборок, ближайших по расстоянию к новой точке, и предсказать метку по ним. Метод не делает предположений о распределении данных, что делает его универсальным и применимым к разнообразным сценариям, но расчет расстояний между всеми объектами в обучающем наборе может быть ресурсоёмким при большом объеме данных.

### 1.2.2 Обучение без учителя

Одним из эффективных способов обнаружения выбросов в многомерных наборах данных является использование лесов. *Изоляционный лес* (Isolation Forest) работает на основе принципа, что аномалии более легко изолировать в структуре данных, чем нормальные точки. Алгоритм создает деревья решений, разбивая признаки случайным образом, и измеряет количество разбиений, необходимых для выделения каждой точки данных, что позволяет выявить аномалии по их относительной изоляции в структуре. Isolation Forest может столкнуться с трудностями в обнаружении выбросов в областях данных с высокой плотностью. Для достижения оптимальной производительности требуется настройка параметров алгоритма, таких как количество деревьев и глубина дерева.

*Фактор локальных выбросов* (Local Outlier Factor, LOF) – алгоритм обнаружения выбросов, который хорошо себя показывает на наборах данных средней размерности. Алгоритм вычисляет оценку, отражающую степень аномальности наблюдений, измеряя локальное отклонение плотности данной точки данных по отношению к её соседям. LOF требует тонкой настройки параметров, таких как количество соседей, а также имеет высокую вычислительную сложность, что может сделать его менее эффективным для больших наборов данных или в ресурсоемких средах.

*Автоэнкодер* – это тип искусственной нейронной сети, используемой для обучения эффективному кодированию немаркированных данных (неконтролируемое обучение). В большинстве случаев автоэнкодер обучается только на «нормальных» данных, так что после обучения он будет точно их восстанавливать, но не сможет сделать этого с незнакомыми аномальными данными. Ошибка реконструкции (ошибка между исходными данными и их низкоразмерной реконструкцией) используется в качестве оценки аномалий для обнаружения аномалий. Автоэнкодер с шумоподавлением может извлекать важные функции и уменьшать шум или влияние бесполезных признаков, однако выбор правильного типа и уровня шума для введения может быть сложной задачей и может потребовать значительных знаний в предметной области.

## 1.3 Работа с данными

Выбор подходящих данных для обучения модели системы обнаружения вторжений – важнейший шаг в обеспечении эффективности и надежности системы. Важно, чтобы набор данных содержал как обычный сетевой трафик, так и различные типы атак, чтобы модель могла изучить закономерности, связанные как с доброкачественной, так и с вредоносной деятельностью [3]. Кроме того, данные должны быть актуальными и отражать текущий спектр угроз, чтобы обеспечить актуальность модели при выявлении возникающих угроз безопасности. Тщательная предварительная обработка и маркировка данных также имеют решающее значение для обеспечения модели точной и значимой информацией для обучения и обобщения. В конечном счёте, хорошо подобранный набор данных является основой способности IDS обнаруживать и эффективно реагировать на потенциальные нарушения безопасности.

Для обнаружения вторжений IDS анализируют трафик формата PCAP. Формат PCAP (Packet Capture) представляет собой стандартный способ хранения данных, касающихся сетевого трафика. Этот формат широко используется в инструментах для анализа и мониторинга сетевого взаимодействия. Пример данных формата PCAP можно увидеть на рисунке 3.

Различные наборы данных предоставляют разное количество признаков – от основных характеристик, таких как временные метки, источники и цели сетевого взаимодействия, до более специфичных параметров, таких как коды ошибок, заголовки протоколов, и объемы передаваемых данных. Вариативность признаков в разных наборах данных играет ключевую роль в формировании обучающих выборок, требуя тщательного анализа и адаптации методов анализа данных в соответствии с особенностями каждого конкретного набора.

В результате анализа имеющихся на сегодняшний день наборов данных, были выбраны следующие: NSL-KDD и HIKARI 2022. NSL-KDD является широко используемым набором данных для задач обнаружения вторжений, представляя собой улучшенную версию оригинального KDD Cup 99. Он включает в себя разнообразные сценарии сетевой активности и предоставляет неплохой набор признаков для обучения моделей. Набор не включает избыточные записи, а количество отобранных записей из каждой группы обратно пропорционально доле записей в исходном наборе данных KDD.

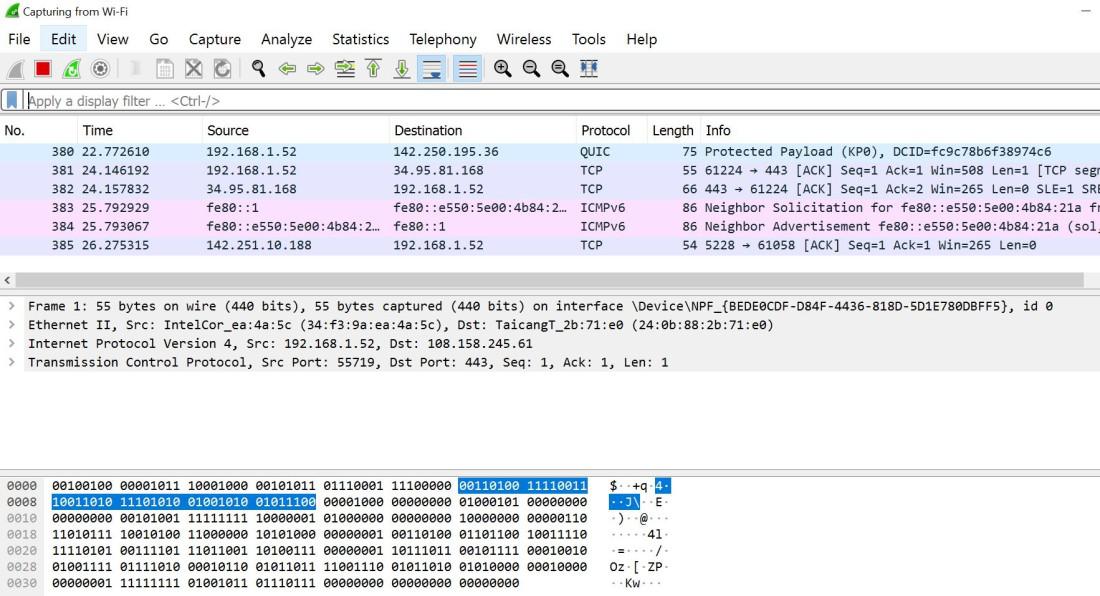


Рисунок 3 – Пример захвата сетевого трафика в утилите Wireshark

HIKARI 2022, в свою очередь, является относительно новым исследовательским набором данных, охватывающим современные сценарии сетевой активности и атак. Он основан на наборе HIKARI 2021 [4] и содержит небольшие доработки (например, удаление дубликатов). Выбор этого набора может быть обусловлен большим количеством признаков (83) и стремлением использовать актуальные данные, отражающие современные угрозы, что может повысить способность модели адаптироваться к современным видам вторжений.

Оба эти набора данных предоставляют богатый материал для обучения и тестирования моделей обнаружения вторжений, позволяя оценивать их производительность в различных сценариях и обеспечивая более реалистичное моделирование сетевых угроз.

## 1.4 Метрики оценки качества

Оценка качества модели требует использования различных метрик, которые помогают определить, насколько эффективно модель справляется с поставленными задачами. Ниже представлены возможные метрики для оценки качества моделей обнаружения вторжений.

*Матрица ошибок* (Confusion Matrix). Метрика состоит из следующих 4 параметров: True Positives (количество правильно классифицированных атак), True Negatives (количество правильно классифицированных отсутствий атак), False Positives (количество ошибочно классифицированных атак), False Negatives (количество ошибочно классифицированных отсутствий атак).

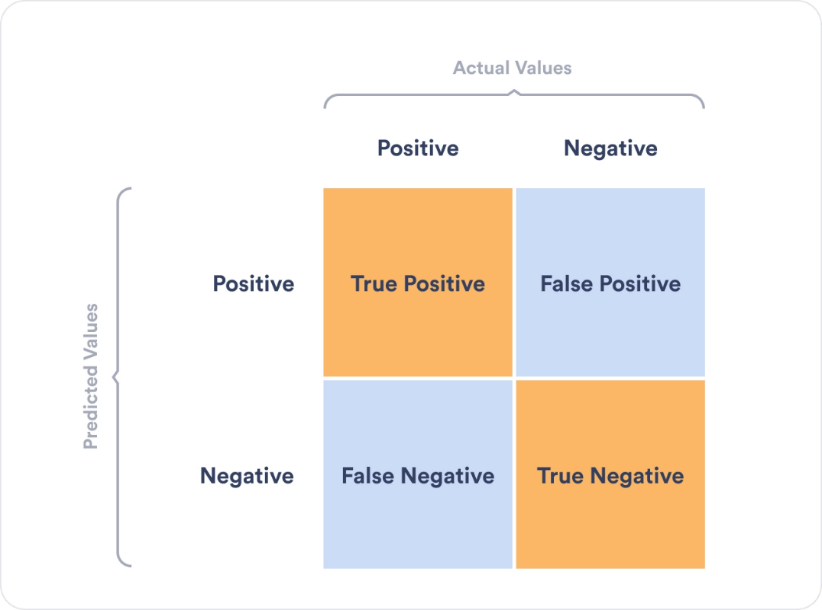


Рисунок 4 – Матрица ошибок

*Точность* (Accuracy). Общая точность классификации. Полезна в случаях, когда классы сбалансированы. Точность также используется как статистическая мера того, насколько хорошо тест двоичной классификации правильно идентифицирует или исключает состояние. То есть точность – это доля правильных предсказаний (как истинно положительных, так и истинно отрицательных) среди общего числа предсказаний. Формула для количественной оценки двоичной точности:

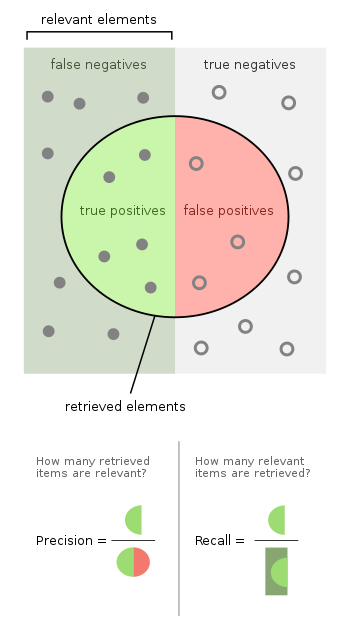


Рисунок 5 – Точность (precision) и полнота (recall)

*Точность* (Precision), также называемая положительной прогностической ценностью) – это доля правильных предсказаний среди полученных предсказаний. Формула:

*Полнота* (Recall), также известная как чувствительность – это доля правильных предсказаний, которые были получены. Формула:

*AUC-ROC* (Area Under the Receiver Operating Characteristic Curve) представляет собой метрику, используемую для оценки производительности бинарных классификационных моделей. ROC-кривая представляет собой график зависимости True Positive Rate от False Positive Rate при варьировании порога классификации. AUC-ROC вычисляется как площадь под ROC-кривой. Она может принимать значения от 0 до 1. Чем выше значение AUC-ROC, тем лучше производительность модели. Если AUC-ROC равен 0.5, это говорит о том, что модель не лучше случайного угадывания. Если AUC-ROC близко к 1, это указывает на высокую способность модели различать классы.

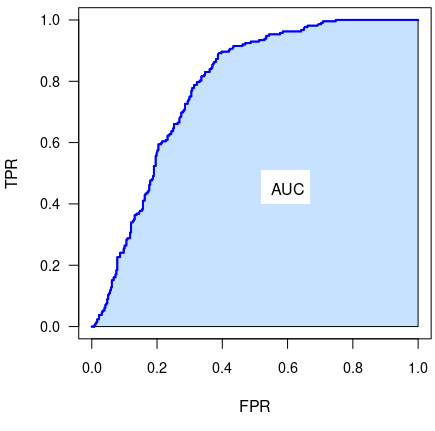


Рисунок 6 – пример кривой ROC и площади под кривой (AUC).

# 2 Проведение экспериментов

В данном разделе описываются процессы проведения экспериментов по обучению и оценке моделей обнаружения вторжений с использованием сервиса Google Colab, библиотеки scikit-learn, а также библиотеки pandas для работы с данными. Google Colab предоставляет бесплатный доступ к вычислительным ресурсам и удобную среду для запуска Python-скриптов. scikit-learn является мощной библиотекой для проведения экспериментов в области машинного обучения, которая включает в себя реализации различных алгоритмов.

Вначале было проведено обучение моделей бинарной классификации на наборе данных NSL-KDD с целью определить обобщающую способность моделей и выделить некоторых фаворитов. Некоторые из признаков в обучающем наборе NSL-KDD:

* duration – длительность соединения в секундах;
* protocol\_type – тип протокола (например, tcp, udp);
* src\_bytes – количество байт, переданных от источника к назначению;
* service – протокол передачи данных, используемый в соединении (например, http, ftp);
* flag – флаги, связанные с соединением (например, SF – успешно завершено, REJ – отклонено);
* attack – тип атаки (например, portsweep, smurf);
* dst\_host\_same\_srv\_rate – какой процент соединений к тому же сервису от различных хостов.

На рисунке 7 в качестве примера приведены первые 5 записей из набора NSL-KDD.

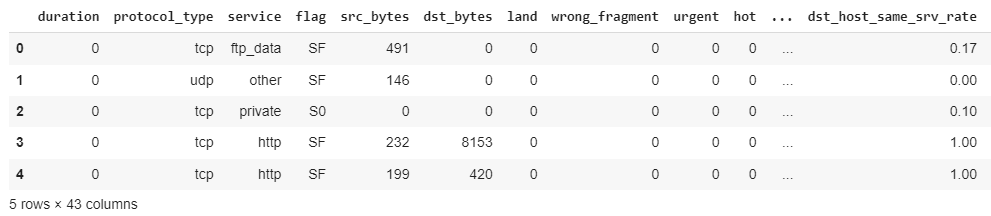


Рисунок 7 – Первые 5 записей набора NSL-KDD (столбцы урезаны по размеру вывода)

Набор данных NSL-KDD изначально разделен на 2 выборки: обучающую и тестовую, поэтому при преобразованиях мы работаем с ними обоими. После импортирования различных библиотек для работы и загрузки наборов данных при помощи функции pandas.read\_csv, начинаем подготовку данных. На рисунке 8 представлен код, преобразующий признак attack в бинарный признак binary\_attack с помощью условного выражения, а также код, преобразующий категориальные признаки в действительные числа.



Рисунок 8 – Фрагмент кода подготовки данных

При помощи метода mutual\_info\_classif библиотеки scikit-learn были выявлены оценки значимости отдельно взятых признаков на значение признака binary\_attack. На рисунке 9 изображены признаки по убыванию их влияния на признак binary\_attack.

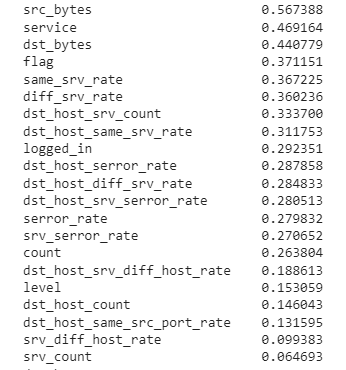


Рисунок 9 – Оценка влияния некоторых признаков из набора NSL-KDD

Как видно из рисунка, наиболее значимыми оказались признаки, отвечающие за передачу данных по сети (src\_bytes – кол-во байт переданных от источника, dst\_bytes – кол-во байт переданных от назначения и service – протокол передачи данных). Некоторые признаки из набора данных могут быть правильно интерпретированы только специалистом, так как обычно в наборах данных присутствуют очень специфичные данные. Поэтому мы выделим из набора 20 самых значимых признаков и проверим точность определения атак.

При помощи метода fit\_transform класса MinMaxScaler проводим масштабирование данных. В результате значения признаков будут находиться в диапазоне [0,1], что может улучшить процессы обучения моделей.

Теперь создадим модели при помощи стандартных классов из библиотеки scikit-learn. Для SVM рассмотрим несколько функций ядра (линейная, полиномиальная и экспоненциальная). Значение количества соседей для модели KNN выбрано перебором, как и значение глубины дерева в модели Decision Tree



Рисунок 10 – Фрагмент кода создания моделей

В цикле запускаем процесс обучения, вызывая для каждой модели общий для них всех метод fit и передавая в этот метод обучающую выборку и отдельно список значений – «правильных ответов», который мы заблаговременно выделили из обучающей выборки. После обучения для каждой модели вызываем метод predict, передавая в него всю тестовую выборку, а затем считаем значения различных метрик (accuracy, precision, recall), чтобы оценить правильность предсказания. На рисунке 11 отображены значения метрик для каждой модели.

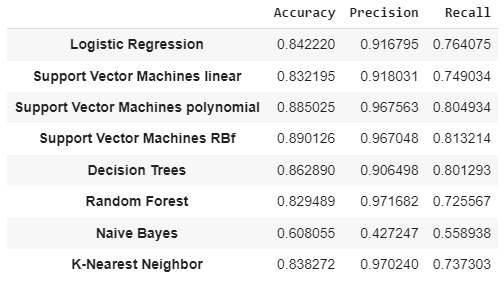


Рисунок 11 – сводка метрик для каждой модели

По самим метрикам не всегда можно точно сказать, какая модель показывает себя лучше, но даже так, мы можем заметить, что почти все модели, кроме модели Байеса, показывают неплохие результаты. Для более общей оценки воспользуемся графиком AUC-ROC.

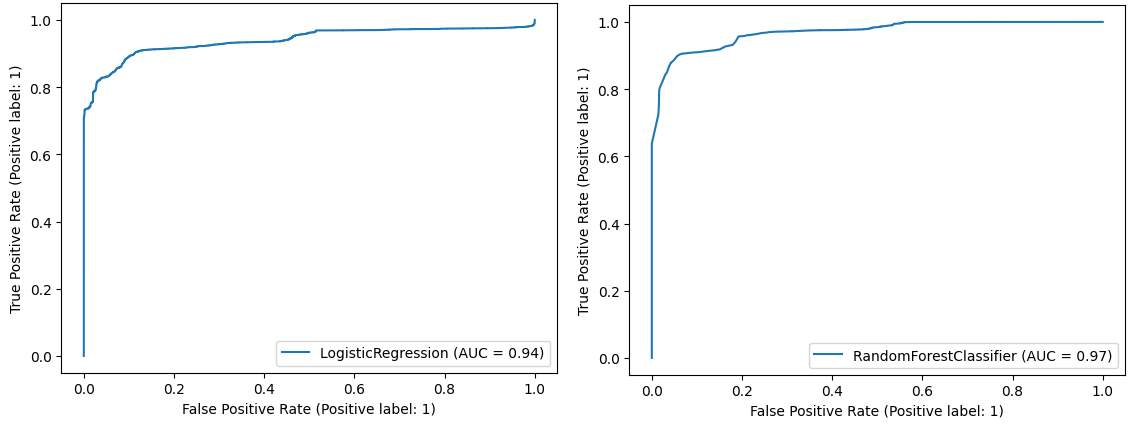


Рисунок 12 – Графики AUC-ROC для наилучших моделей

В результате оценивания, модель Random Forest получила значение 0.97, дальше по убыванию логистическая регрессия (0.94) и модели SVM (0.94). Для дальнейшего исследования были отобраны 3 лучшие модели, однако SVM хоть и является неплохой по показателям, работает довольно медленно, что может оказаться не очень эффективным. Следующая по убыванию после SVM идёт модель Decision Tree (0.87).

Далее были проведены эксперименты, целью которых было оценивание адаптируемости и обучаемости моделей. В наборе NSL-KDD представлено множество разновидностей атак, хотя некоторые из них имеют схожую природу. В связи с этим, было принято решение оценить качество обнаружения Random Forest при удалении каждого из типов атак.

Шаги при этом эксперименте были выполнены те же, что и в прошлый раз, за исключением создания словаря, в котором хранятся идентификаторы записей для каждой конкретной атаки.

Алгоритм таков: в цикле по ключам из словаря атак, удаляем из обучающего набора все записи этой атаки, обучаем модель и проверяем общее качество распознавания на тестовой выборке. Поскольку различных данных и метрик много, за качество определения было грубо принята точность (accuracy). Результаты изображены на рисунке 13. Можно заметить, что несмотря на отсутствие в обучающей выборке примеров атак какой-либо категории, за счёт того, что некоторые атаки имеют схожую природу, точность определения не всегда равна 0. Тем не менее, в большинстве случаев, для успешного определения атаки необходимо достаточно много обучающих примеров.

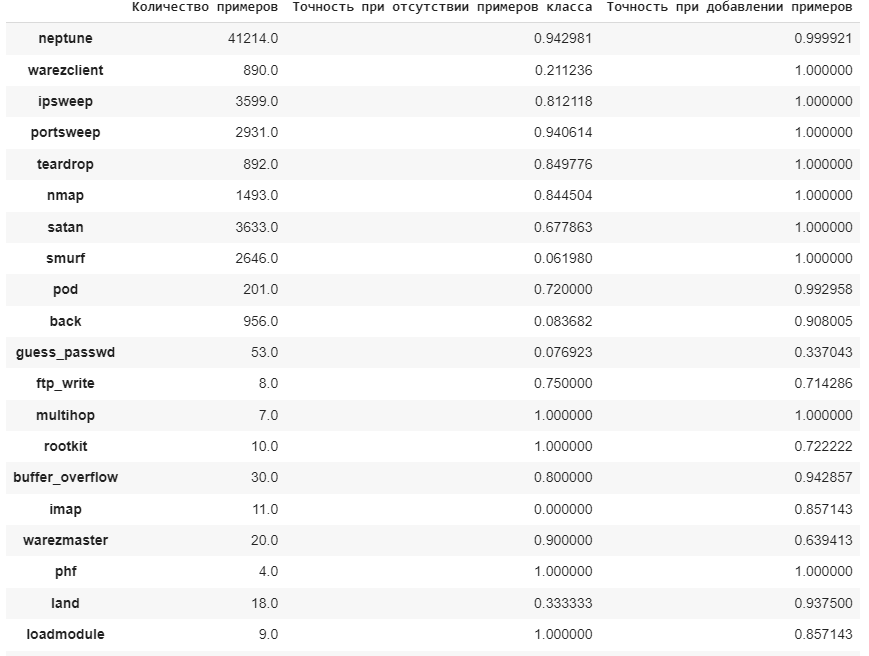


Рисунок 13 – Оценивание точности выявления конкретных атак до и после добавления в обучающую выборку

Теперь же, выполним эксперимент, направленный на выявление возможности моделей обучаться под новые атаки. В этом случае используем набор данных HIKARI 2022. В нём представлены более общие типы атак, при удалении которых модель покажет гораздо меньшую точность.

Набор HIKARI 2022 содержит 83 признака и некоторые из них должны быть обработаны перед подачей выборки на обучение. Признак «время соединения» (flow\_duration), который был представлен строкой формата «0 days 0:00:00:000000», пришлось привести к категориальному признаку в соответствии с оценкой временного интервала (long, medium, short, very short, zero), а затем к числовому признаку. Также из набора были удалены некоторые признаки со значениями, характерными лишь для конкретной сети (например, IP-адрес источника и получателя), поскольку информация о конкретных IP-адресах может быть специфичной для обучающего набора данных и не обобщаться эффективно на новые, реальные сценарии атак. Также с помощью тех же методов, что и в случае с NSL-KDD, были выявлены наиболее влиятельные признаки. Среди них нас будут интересовать 30 наиболее важных.

Последовательность действий аналогична предыдущему эксперименту, за исключением того, что в этот раз мы будем помимо оценивания эффективности обнаружения, дообучать модель на примерах «новой» атаки и проверять эффективность на той же тестовой выборке. Модель Random Forest не поддерживает механизм дообучения, так как во время обучения сравнивает случайным образом сформированные леса и выбирает из них наилучший. Проведём наш эксперимент на моделях, поддерживающих корректировку. Оценивать качество будем на метриках accuracy, precision, recall, а также auc. На рисунке 14 отображен вывод оценок (столбцы метрик с подписью 2 означают значения метрик после дообучения).

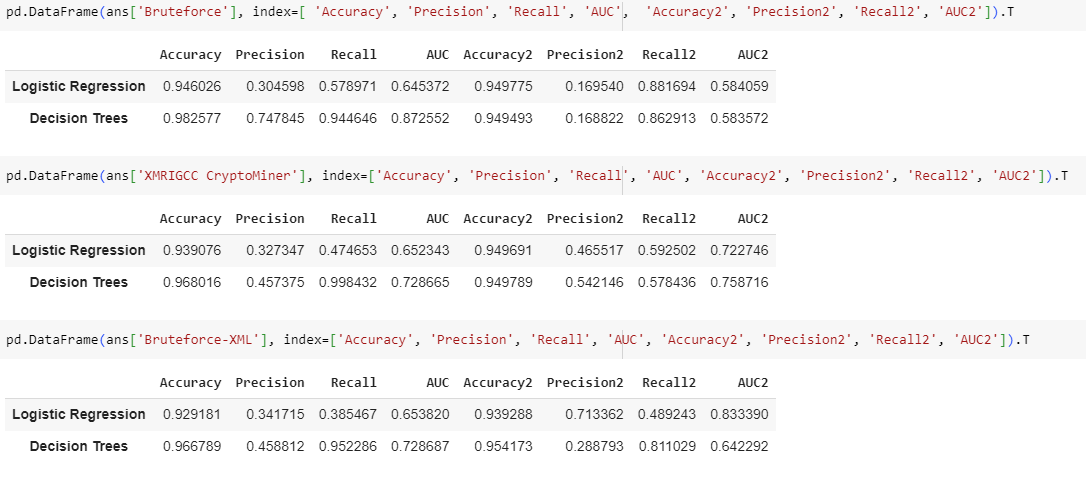


Рисунок 14 – Оценки моделей до и после дообучения

По результатам можно сказать, что в случае логистической регрессии дообучение улучшило модель, а в случае Decision Tree – ухудшило. Можно сделать вывод, что модель логистической регрессии может применяться для решения задачи обнаружения вторжений, однако не факт, что она будет стабильно улучшаться при каждой итерации дообучения. Вероятнее всего, для улучшения качества модели, лучшим выходом будет сбор и разметка новых данных, а затем полное переобучение под них. Однако надо понимать, что такие процессы при особенно частом появлении новых атак, требуют вычислительных ресурсов и времени. Некоторые методы машинного обучения поддерживают распараллеливание (например, Random Forest), следовательно, выбор модели бинарной классификации может оказаться вполне приемлемым.

Далее были рассмотрены методы обучения без учителя (Isolation Forest, автоэнкодеры), основанные на обнаружении выбросов (аномалий). Для обучения этих моделей, из обучающей выборки был вытянут и нормализован с помощью метода normalize лишь «нормальный» трафик. С помощью все той же библиотеки scikit-learn, были созданы модели, а также разбит набор данных на обучающую и тестовую выборки в соотношении 9:1. Доля обучающей выборки может играть важную роль в процессе обучения, так как модель должна наилучшим образом запомнить данные, но не учитывать излишние особенности. На рисунке 15 представлены результаты эксперимента.

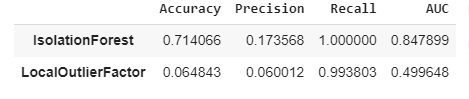


Рисунок 15 – Оценка моделей обнаружения выбросов

Эксперимент показал, что модель изоляционного леса (Isolation Forest), как и модель фактора локальных выбросов (Local Outlier Factor) имеют потенциал в определении аномалий, хотя изоляционные леса справились значительно лучше. Стоит отметить, что модель фактора локальных выбросов значительно уступает и по скорости обучения. Тем не менее, при тестировании на наборе данных NSL-KDD, модель LOF показала лучший результат (AUC = 0.73), чем изоляционный лес (AUC = 0.68). Это означает, что для применения их в профессиональном контексте необходимо тщательно подбирать параметры, учитывая особенности данных и требования конкретного сценария.

Также была исследована возможность использования автоэнкодеров. Для этого был сконструирован ряд нейронных сетей при помощи библиотеки keras и подобраны некоторые параметры (функции активации, количество нейронов в слоях и т. д.). Согласно официальной документации keras [5], были созданы несколько слоёв для кодировщика и декодировщика с функцией активации relu (она показывает себя лучше всего). Создание «бутылочного горлышка» способствует более успешному обобщению нормальных данных, так как нейросеть пытается не скопировать данные, а запомнить общие черты. Для этого размерность слоёв уменьшается к середине и увеличивается на концах (на выходе мы должны иметь слой той же размерности, что и входной). На рисунке 16 представлен код автоэнкодера.



Рисунок 16 – Код создания нейросети автоэнкодера

Главная проблема автоэнкодеров – выбор порога, при котором входные данные будут определяться как аномальные. Экспериментально был вычислен порог в 0.0004 для текущей конфигурации нейросети. Результат тестирования можно увидеть на рисунке 17.

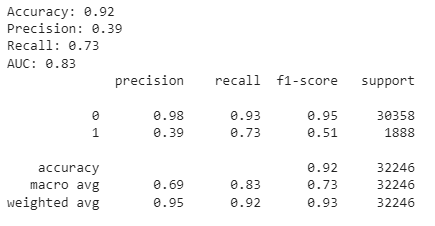


Рисунок 17 – Лучший результат применения автоэнкодера

Также был рассмотрен вариант автоэнкодера с зашумлением, где перед каждым слоем был слой Dropout, отвечающий за добавление шума к данным величиной в 0.1, однако лучший показатель AUC этой модели был всего 0.61.

Для применения моделей обнаружения выбросов может потребоваться провести дополнительные исследования в предметной области, а также тонко подбирать параметры. Это затрудняет их использование, однако в контексте обнаружения вторжений, у них есть потенциал показать наилучший результат.

# ЗАКЛЮЧЕНИЕ

За время выполнения работы были изучены аспекты использования машинного обучения на практике в решении задачи обнаружения вторжений. Проведенные исследования позволили выявить требования, предъявляемые к потенциально разрабатываемому решению, и проанализировать различные подходы к решению поставленной задачи.

Полученные результаты говорят о том, что выбор подхода к применению машинного обучения в задаче обнаружения вторжений должен быть обоснованным и зависеть от характеристик конкретной задачи и особенностей данных. Каждая из рассмотренных моделей имеет свои преимущества и ограничения, и для достижения оптимальной производительности в конкретном контексте требуется грамотный выбор и тщательная настройка модели.

Также можно подчеркнуть, что данная работа не только позволила лучше понять принципы функционирования систем обнаружения вторжений, но также предоставила практические навыки в области применения современных методов машинного обучения для решения реальных проблем в области кибербезопасности. Полученные знания и опыт в данной области послужат отличным фундаментом для дальнейших исследований и разработок в области обеспечения безопасности информационных систем.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Sung A. H., Mukkamala S. Identifying important features for intrusion detection using support vector machines and neural networks // 2003 Symposium on Applications and the Internet. - 2003. - C. 209-216
2. scikit-learn Machine Learning in Python // scikit-learn URL: <https://scikit-learn.org/stable/> (дата обращения: 10.12.2023).
3. Kwon D., Neagu R., Prem R., Ryu J. T., Kim J. Evaluating Unbalanced Network Data for Attack Detection // In Proceedings of the 2023 on Systems and Network Telemetry and Analytics (SNTA '23). - 2023. - С. 23-26. <https://doi.org/10.1145/3589012.3594898>
4. Ferriyan A., Thamrin A.H., Takeda K., Murai J. Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic // Appl. Sci. 2021, 11, 7868. <https://doi.org/10.3390/app11177868>
5. Building Autoencoders in Keras // The Keras Blog URL: <https://blog.keras.io/building-autoencoders-in-keras.html> (дата обращения: 10.12.2023).