МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра Прикладная математика и информатика
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Исследование и реализация алгоритмов интеллектуального анализа сетевого трафика»

Обучающийся Р.Р. Санчилеев
(Инициалы Фамилия) (личная подпись)

Руководитель д.т.н., доцент, С.В. Мкртычев
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант к.п.н., доцент, О.Н. Брега
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы – «Исследование и реализация алгоритмов интеллектуального анализа сетевого трафика».

Данная работа посвящена исследованию и реализации алгоритмов интеллектуального анализа сетевого трафика. В работе проводится обзор существующих методов анализа трафика, а также исследуется применение методов машинного обучения и искусственного интеллекта для выявления аномалий, классификации данных и прогнозирования сетевой активности.

Целью данного исследования является исследование и реализация алгоритмов интеллектуального анализа сетевого трафика с целью повышения эффективности управления сетевыми ресурсами и обеспечения безопасности передачи данных.

Объектом исследования является сетевой трафик, передаваемый по различным сетевым протоколам.

Предметом исследования являются методы и алгоритмы анализа этого трафика с использованием методов машинного обучения.

Актуальность данного исследования обусловлена необходимостью эффективного управления сетевыми ресурсами, защиты информации от кибератак и выявления аномалий в сетевом трафике, что становится все более важным в условиях быстрого развития информационных технологий и увеличения объемов передаваемых данных.

Данная работа состоит из введения, трех разделов, заключения и списка используемой литературы.

Работа включает 50 страницу текста, 15 рисунков, 2 таблицы и 23 источника.

Abstract

The topic of the bachelor's thesis is "Research and implementation of algorithms for intelligent analysis of network traffic".

This work is devoted to the research and implementation of algorithms for intelligent analysis of network traffic. The paper reviews existing traffic analysis methods and investigates the application of machine learning and artificial intelligence methods for anomaly detection, data classification and network activity prediction.

The purpose of this research is to investigate and implement algorithms for intelligent analysis of network traffic in order to improve the efficiency of network resource management and ensure the security of data transmission.

The object of the study is network traffic transmitted through various network protocols.

The subject of the study are methods and algorithms for analysing this traffic using machine learning methods.

The relevance of this study is due to the need for effective management of network resources, protection of information from cyberattacks and detection of anomalies in network traffic, which is becoming increasingly important in the rapid development of information technology and increasing volumes of transmitted data.

This paper consists of an introduction, three chapters, a conclusion and a list of references.

The paper includes 50 pages of text, 15 figures, 2 tables and 23 sources.

Содержание

Введение 5
1 Обзор алгоритмов анализа сетевого трафика 6
1.1 Постановка задачи и общие понятия и определения алгоритмов анализа сетевого трафика 6
1.2 Направления развития анализа сетевого трафика 8
1.3 Общая схема инфраструктурных алгоритмов анализа сетевого трафика 14
1.4 Классификация средств мониторинга и анализа 18
1.5 Исследование и выбор математической модели сетевого трафика 22
2 Обзор алгоритмов анализа сетевого трафика 28
2.1 Архитектура разрабатываемой системы 28
2.2 Моделирование анализа сетевого трафика 29
2.3 Проектирование с полным набором параметров и определение их значимости 32
2.4 Проектирование с сокращенным набором параметров и определение их значимости 35
3 Реализация и тестирование модуля интеллектуального анализа 39
3.1 Реализация и тестирование модуля интеллектуального анализа. 39
Заключение 44
Список используемой литературы и используемых источников 46
Приложение А. Реализация алгоритмов 49

Введение

В современном информационном обществе сетевой трафик играет ключевую роль в обеспечении связности и эффективности передачи данных. Однако, с ростом объемов и разнообразия трафика возникает необходимость в разработке и применении интеллектуальных алгоритмов анализа для обеспечения безопасности, оптимизации сетевых процессов и выявления аномалий. Анализ сетевого трафика стал очень важной областью исследований в области информатики. Алгоритмы анализа сетевого трафика используются для различных целей, включая мониторинг производительности, обнаружение вторжений и кибербезопасность.

Целью данного исследования является исследование и реализация алгоритмов интеллектуального анализа сетевого трафика с целью повышения эффективности управления сетевыми ресурсами и обеспечения безопасности передачи данных.Для достижения поставленной цели необходимо провести анализ существующих методов анализа сетевого трафика, их преимуществ и недостатков, разработать новые алгоритмы или усовершенствовать существующие, а также провести экспериментальное исследование для оценки их эффективности и применимости на практике.

Объектом исследования является сетевой трафик, передаваемый по различным сетевым протоколам, включая TCP/IP, UDP и другие.

Предметом исследования являются методы и алгоритмы анализа этого трафика с использованием методов машинного обучения, статистического анализа, искусственного интеллекта и других подходов.

Актуальность данного исследования обусловлена необходимостью эффективного управления сетевыми ресурсами, защиты информации от кибератак и выявления аномалий в сетевом трафике, что становится все более важным в условиях быстрого развития информационных технологий и увеличения объемов передаваемых данных.

1 Обзор алгоритмов анализа сетевого трафика

1.1 Постановка задачи и общие понятия и определения алгоритмов анализа сетевого трафика

С развитием сетевых технологий и ростом объемов передаваемой информации становится все более актуальной задача анализа сетевого трафика. «Традиционные методы анализа и мониторинга уже не всегда способны эффективно обнаруживать и предотвращать различные сетевые угрозы и аномалии. Рассмотрим построения системы обнаружения вторжений с использованием интеллектуального анализа сетевого трафика» [1]. Далее определим условия для будущей системы обнаружения вторжений и выработаем её структуру. Для принятия решений о возможных атаках предлагается применить методы индуктивного машинного обучения, в частности – искусственные нейронные сети.

Целью данной работы является исследование и реализация алгоритмов интеллектуального анализа сетевого трафика для обнаружения сетевых аномалий и угроз.

Рассмотрим далее основные понятия и определения.

Анализ сетевого трафика (NTA) – это процесс, включающий сбор и анализ данных из различных источников с целью обнаружения и идентификации сетевых вторжений, неправомерного использования и злонамеренных действий. Алгоритмы NTA можно использовать для обнаружения и диагностики аномального поведения, обнаружения неизвестных угроз и идентификации злоумышленников.

«Сетевой трафик – объём информации, передаваемой через компьютерную сеть за определённый период времени. Количество трафика измеряется как в пакетах, так и в битах, байтах и их производных: килобайт (КБ), мегабайт (МБ) и т. д» [20].

Алгоритмы анализа сетевого трафика – это алгоритмы, которые

помогают определить определенные паттерны в сетевом трафике. Они могут использоваться для обнаружения злоумышленной активности, определения злоумышленных IP–адресов, обнаружения подозрительной активности и идентификации потенциальных угроз. Основными преимуществами таких алгоритмов являются быстрое обнаружение злоумышленной активности, а также возможность определения потенциальных угроз. Однако они имеют некоторые недостатки, такие как невозможность идентификации однозначного источника злоумышленной активности, а также необходимость обработки большого объема данных [4].

Алгоритмы NTA можно разделить на две основные категории: основанные на сигнатурах и основанные на аномалиях. Алгоритмы на основе сигнатур используют предопределенный набор правил и сигнатур для обнаружения известных угроз. Эти правила и сигнатуры основаны на характеристиках конкретной угрозы и используются для выявления вредоносных действий. Алгоритмы на основе сигнатур обычно используются для обнаружения известных вирусов и червей.

Алгоритмы на основе аномалий обнаруживают действия, выходящие за рамки нормальных паттернов. Эти алгоритмы используют искусственный интеллект, машинное обучение и другие передовые технологии для обнаружения неизвестных угроз. Алгоритмы на основе аномалий используются для обнаружения новых или ранее неизвестных вредоносных действий и могут использоваться для выявления подозрительного поведения.

В дополнение к этим двум категориям существует несколько подкатегорий алгоритмов NTA, которые используются для анализа сетевого трафика. Алгоритмы анализа пакетов исследуют сетевые пакеты для обнаружения вредоносных действий. Эти алгоритмы используются для выявления вредоносного кода или действий, встроенных в пакеты. Алгоритмы анализа протокола используются для выявления подозрительных действий на основе нарушений протокола. Алгоритмы анализа потока данных применяются для выявления аномалий и подозрительной активности в

сетевом трафике. Они позволяют обнаруживать признаки несанкционированного доступа к данным или ресурсам, а также выявлять нетипичные паттерны в передаче информации [8].

Анализ потока данных основан на изучении характеристик сетевых соединений, таких как объем передаваемых данных, частота обращений, используемые протоколы и порты. Алгоритмы сравнивают наблюдаемые параметры с эталонными значениями или моделями нормального поведения, чтобы определить отклонения, которые могут указывать на вторжение или атаку.

Применение алгоритмов анализа потока данных позволяет повысить защищенность информационных систем, своевременно обнаруживая и предотвращая несанкционированные действия. Они являются важным компонентом комплексной системы обеспечения информационной безопасности. Наконец, алгоритмы NTA можно использовать в сочетании с другими методами, такими как сетевая криминалистика, приманки и системы обнаружения вторжений, для обнаружения и смягчения вредоносных действий. Это может помочь создать комплексную систему безопасности, способную эффективно и своевременно обнаруживать угрозы и реагировать на них.

Алгоритмы NTA являются важными инструментами для обнаружения вредоносных действий и защиты сетей. Они обеспечивают важный уровень безопасности, помогающий выявлять и устранять угрозы до того, как они нанесут ущерб. Таким образом, важно, чтобы сети были оснащены соответствующими алгоритмами NTA, чтобы гарантировать их защиту от злоумышленников.

1.2 Направления развития анализа сетевого трафика

«Выделяют два основных направления:

– глубина анализа индивидуальных сетевых пакетов, приводящее к

расширению модели OSI и более детальному рассмотрению передаваемых данных;

– увеличение полноты учета состояния как самого потока данных, так и связанных с ним других потоков» [2].

На рисунке 1 представлены три ключевых этапа развития.

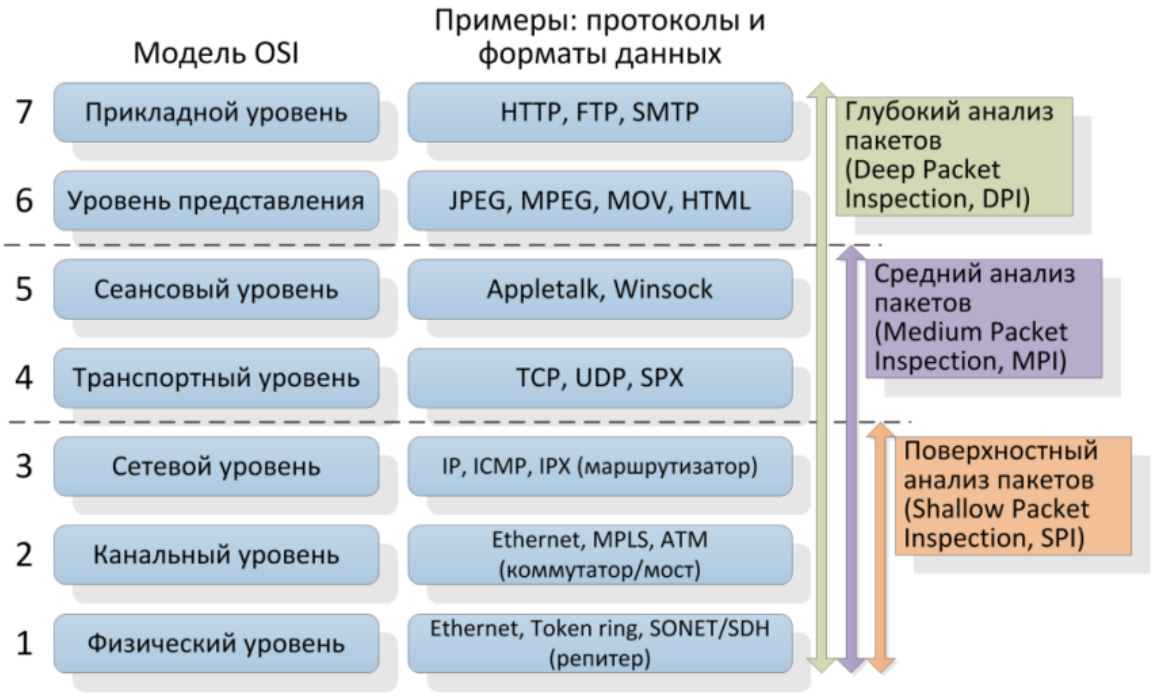


Рисунок 1 – Градации эволюции аналитики сетевого трафика, опирающиеся на глубину исследования

«Поверхностный анализ пакетов – анализ трафика, основанный на заголовках пакетов уровней L1–L3 по модели OSI, характеризуется низкими требованиями к вычислительным ресурсам, что позволяет обрабатывать большие объемы трафика. Эта технология широко распространена и используется в большинстве межсетевых экранов операционных систем, например, в Windows XP/Vista и OS X, маршрутизаторов и других сетевых устройств. Она также лежит в основе реализации сетевых списков контроля доступа на уровне IP–адресов и портов (Access Control List, ACL). Таким образом, данная технология является эффективной для разграничения доступа

извне к отдельным компьютерам (IP) и сервисам (порты) внутренней сети» [3].

«С помощью MPI можно, например, заблокировать возможность получения flash–файлов или картинок с определенных интернет сервисов (на уровне представления OSI) или заблокировать часть команд (на уровне приложения OSI) в отдельных протоколах. Набор протоколов, как правило, очень ограничен. Например, в первых версиях CheckPoint FireWall–1 (CheckPoint FW–1) поддерживались протоколы Telnet, FTP, HTTP, а в Cisco Private Internet Exchange (Cisco PIX) – FTP, HTTP, H.323, RSH, SMTP и SQLNET. Впоследствии данные наборы незначительно расширялись. Также известно, что данная технология используется в продуктах компаний McAfee и Symantec» [5].

Для решения этих проблем был разработан протокол ICAP, который позволяет прокси–серверам отправлять прохода на рисунке 2.

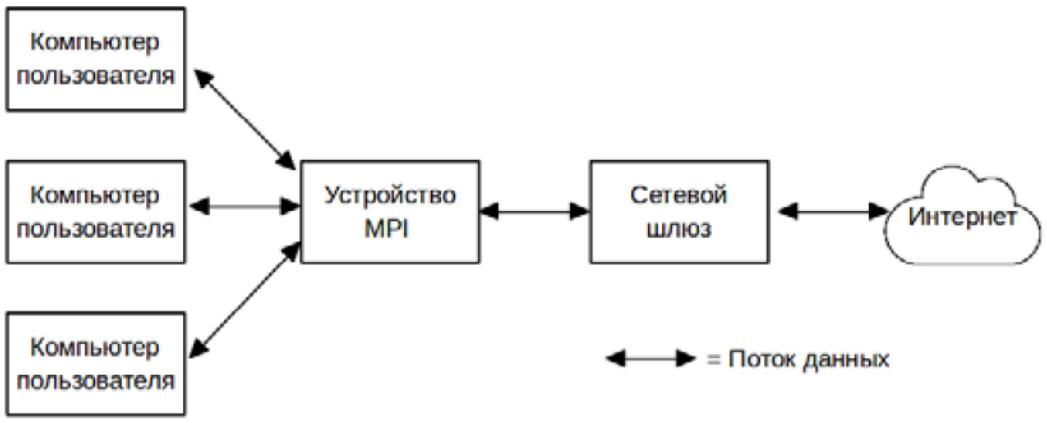


Рисунок 2 – Схема применения устройств анализа на основе технологии MPI

«Глубокий анализ пакетов иногда используется более узкий термин – DPP (Deep Packet Processing), который охватывает различные манипуляции с пакетами, такие как их модификация, фильтрация или перенаправление. В настоящее время оба термина часто рассматриваются как взаимозаменяемые. Эта технология является естественным развитием MPI. В рамках этого подхода анализатор полностью изучает содержимое каждого пакета» [6].

«Одним из важных отличий от предыдущих технологий является то, что системы на основе DPI могут принимать решения не только на основе содержания пакетов, но и на основе косвенных признаков, присущих определенным сетевым программам и протоколам. Для этого может использоваться статистический анализ, такой как анализ частоты встречаемости определенных символов, длин пакетов, расстояние между метками времени последовательных пакетов и так далее. Кроме того, по сравнению с предыдущими подходами, список применений технологии значительно расширен: от классификации и ограничения полосы до приоритизации, маркировки, кэширования и так далее» [7].

С развитием вычислительной мощности современных процессоров распространение технологии глубокого анализа пакетов (DPI) стало массовым явлением. Благодаря высокой скорости и производительности этих процессоров, теперь мы можем проводить детальный и точный анализ сетевого трафика в режиме реального времени.

В отличие от более поверхностного анализа пакетов (MPI), DPI была создана с акцентом на скорость обработки и точность идентификации различных сетевых приложений. Это обеспечивает эффективное масштабирование решений, основанных на DPI, как по пропускной способности сетевого канала, так и по числу приложений, которые можно распознать.

Важно отметить, что соответствие между разными уровнями точности классификации не всегда однозначно, как показано на рисунке 3.

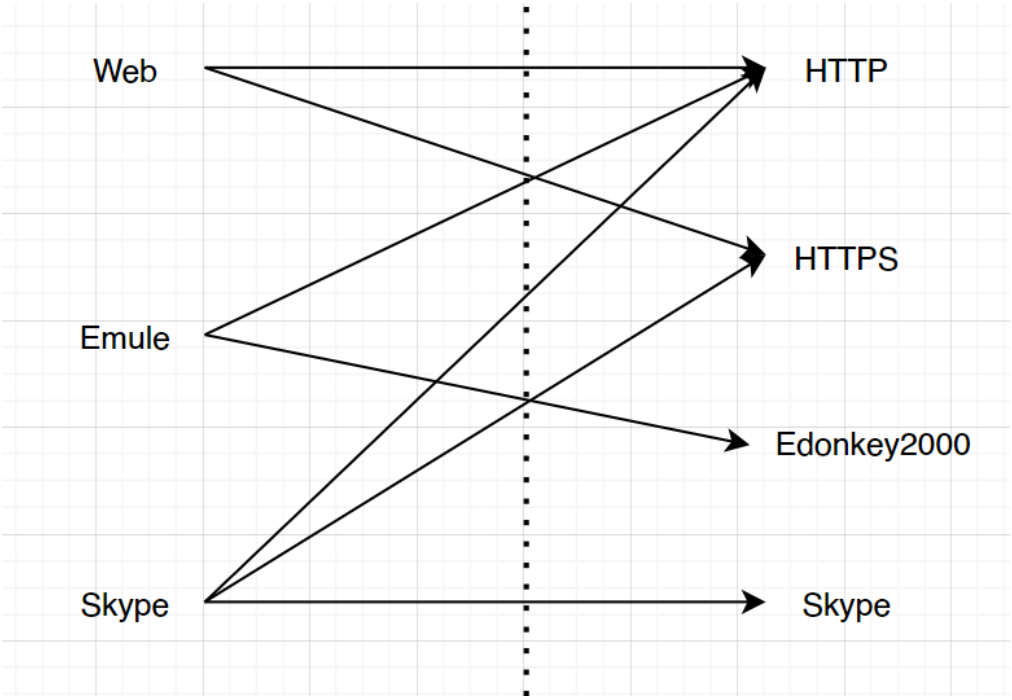


Рисунок 3 – Различие между идентификацией приложений (слева) и протоколов (справа)

«Технология глубокого анализа пакетов (DPI) является основным стандартом для анализа сетевого трафика и является критически важной для обеспечения сетевой безопасности и соблюдения законодательства. В последнее время были приняты международные стандарты, требования и рекомендации относительно реализации DPI, ее внутренней структуры и функциональных возможностей.

Хотя DPI редко используется в межсетевых экранах, за исключением некоторых, таких как Hogwash и Shield, но экраны четвертого поколения могут интегрировать функциональность DPI в процессе анализа сетевого трафика [14].

Еще одним направлением в развитии технологии анализа является учет состояния протокола (потока) в процессе обработки, известный как анализ с сохранением состояния/без сохранения состояния. Это направление особенно важно для протоколов, использующих ориентированный на установление соединения транспортный протокол, таких как TCP» [9].

Протокол EXPRESS Internet Protocol (EIP) представляет собой

инновационную концепцию транспортного протокола, основанную на особенностях UDP, обеспечивающую быструю и надежную передачу данных. Важно осознавать, что даже при использовании UDP-пакетов невозможно полностью избежать stateful анализа.

В исследовании мы рассматриваем "однонаправленный поток на транспортном уровне" как последовательность пакетов, передаваемых от определенного IP-адреса и TCP/UDP-порта к определенному IP-адресу и TCP/UDP-порту, с указанием протокола транспортного уровня (TCP/UDP). Таким образом, поток характеризуется пятью параметрами.

На рисунке 4 представлен список уровней учета состояния потока, включая:

– «анализ отдельных пакетов без учета потоков и состояний (Packet Based No State, PBNS);
– анализ пакетов в рамках потоков (Packet Based Per Flow State, PBFS);
– анализ сообщений в рамках потока (Message Based Per Flow State, MBFS), включая сборку IP–фрагментов в IP–пакеты (IP–нормализация) и сборку TCP–сегментов в TCP–сеансы (TCP–нормализация);
– анализ сообщений в рамках протокола (Message Based Per Protocol State, MBPS), где учитывается состояние автомата протокола, включая возможность принятия определенного типа сообщений. Пример автомата состояний протокола HTTP представлен на рисунке 4, где вершины отображают состояния, а рёбра представляют условия перехода, возможные действия при приёме/отправке сообщений и обработку сообщений, а также таймауты» [10].

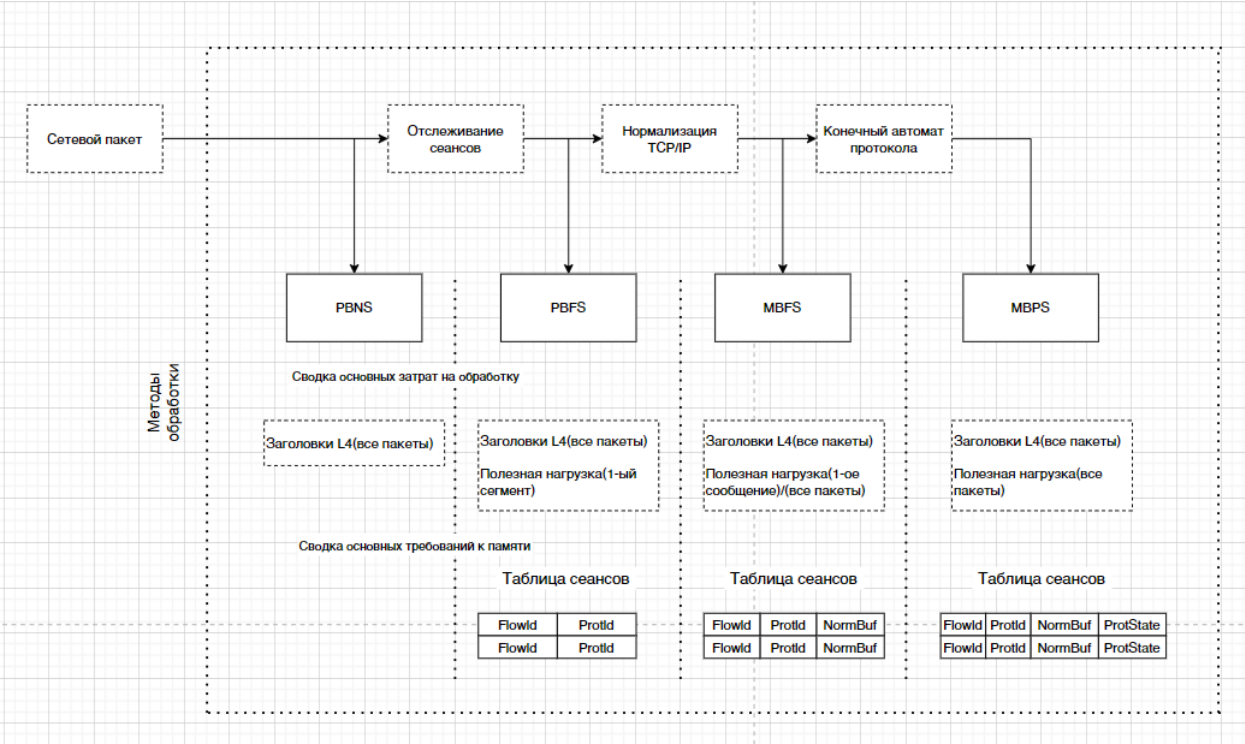


Рисунок 4 – Градации полноты учёта состояния потока

«Основные реализации DPI–технологии обычно ориентированы на stateless–анализ, где оценка выполняется на уровне отдельных пакетов без сохранения состояния между анализом нескольких пакетов одного сетевого потока. Этот уровень точности обычно достаточен для многих практических применений и позволяет эффективно использовать ресурсы. Однако существуют задачи, требующие более высокой точности. В качестве примеров можно упомянуть две технологии, основанные на statefull–подходе: инспекцию пакетов с сохранением состояния (SPI) и глубокий анализ содержимого (DCI)»[11].

1.3 Общая схема инфраструктурных алгоритмов анализа сетевого трафика

Исследование сетевого трафика – это словно танцевальная симфония, где каждое движение несет в себе свою собственную гармонию. Оно

начинается с изящного "захвата" пакетов, которые тихо плывут по потоку контролируемого сетевого соединения, формируя ансамбль данных, готовых к тщательному анализу.

Но чтобы достичь идеальной грации в точности и скорости анализа, и не изнурять ресурсы, приходится подбирать танцевальные приемы с умом. Например, есть такой трюк, как "слайсинг", когда мы рассматриваем только малую часть содержимого пакетов – это как раз для того, чтобы быстро и точно классифицировать потоки по протоколам. А ещё есть вариант с «эмплированием», когда мы выбираем лишь избранные пакеты, исходя из определенных критериев, чтобы не перегрузить себя излишними данными [16].

Другим методом является сэмплирование, при котором для анализа выбирается только часть пакетов в зависимости от определенных условий. Различные стратегии выборки, такие как равномерное сэмплирование, могут использоваться для мониторинга трафика, однако они могут приводить к недооценке среднего размера пакетов из–за предпочтения меньших пакетов.

В технических задачах, где требуется пристальное внимание к сетевому трафику, особенно в контексте обеспечения безопасности сети, ключево применение специализированных методов. Один из таких подходов – это тщательный захват пакетов без каких-либо потерь, известный как глубокий захват пакетов (DPC). Этот метод позволяет нам заполучить и сохранить все данные, проходящие через сеть, без потерь информации.

После этапа захвата пакетов следует их классификация и объединение в потоки с учетом различных параметров, таких как источник и назначение адресов. Этот процесс превращает стандартные пакеты данных в новую форму для анализа, которую мы называем сетевым потоком. Отличие подхода к анализу потоков от традиционного пакетно-ориентированного анализа заключается в том, что нам не нужно заглядывать в каждый пакет, что значительно сокращает объем данных, подлежащих обработке.

Для трансфера накопленной информации от исходной точки до места ее

анализа применяются разнообразные протоколы, такие как IPFIX, Cisco NetFlow и Juniper Jflow. Эти технологии гарантируют пересылку данных в формате, оптимальном для аналитических систем, обеспечивая высочайшую точность и эффективность в процессе анализа.

«В рамках анализа потоков существует разнообразие данных, описывающих эти потоки. Один из наиболее общих наборов данных включает в себя следующие характеристики:

– IP адреса отправителя и получателя,
– протокол транспортного уровня,
– при использовании TCP/UDP – номера портов отправителя/получателя,
– счетчики, включающие количество переданных пакетов и байт, а также время начала и окончания потока» [12].

Заметно, что, несмотря на удобство метода захвата пакетов без потерь (lossless capture) для анализа данных, он оказывается ограниченным в гибкости по сравнению с другими подходами, такими как слайсинг и сэмплинг. Например, невозможность динамической настройки объема поступающих данных, зависящего от входных потоков, становится проблемой.

В реальных сценариях часто возникает проблема большого количества кратковременных потоков, известных как "всплески" (flash flows). Это приводит к тому, что число потоков существенно меньше количества переданных пакетов. Для решения этой проблемы предлагается применять метод сэмплирования потоков.

В рамках проведенных исследований оценивается точность подхода, основанного на анализе данных потоков, а также изучается влияние временных ограничений на эту точность. Кроме того, в рамках данной работы представлен инструмент FLOW-REDUCE, который собирает полную информацию о потоке из его фрагментов, учитывая временные ограничения, на которые этот поток был разделен.

«Выполнение классификации на уровне приложения или конкретного

сетевого протокола ведёт к формированию нового объекта для анализа – сетевого потока определенного протокола или приложения (в случае VoIP, например, это SIP и RTP). После этого этапа возможна дополнительная обработка объекта, в зависимости от поставленной задачи: разбор полей протокола, сборка сессии для протоколов с установлением соединения, извлечение контента (HTML страниц, файлов различных типов, электронных писем, аудио–видео потоков и прочего), а также анализ данных приложения (рисунок 5)» [13].

Рисунок 5 – Различия типичных схем packet (слева) и flow–based (справа) анализа

Для более глубокого осмысления данной тематики приобретает важность рассмотрение дополнительного источника информации о сетевой активности, который не опирается на ранее обсуждаемые подходы, связанные с анализом пакетов и потоков данных. Этим источником является база управления информацией, или MIB. MIB – это виртуальное хранилище данных, которое служит для организации объектов в инфраструктуре сети.MIB – это центральная часть системы управления сетевой инфраструктурой, которая обеспечивает взаимодействие между различными устройствами и сервисами в сети. Она содержит информацию о различных аспектах сети, включая конфигурацию устройств, состояние сети и статистику трафика.

Включение MIB в обсуждение темы сетевого трафика позволяет получить более полное представление о сети и ее работе. Это особенно важно для анализа и мониторинга сети, так как MIB может предоставить важную информацию о состоянии сети и ее компонентов.

«Модули для сбора, хранения и обмена данными в формате MIB присутствуют в большинстве сетевых устройств, и передача данных осуществляется по протоколу SNMP. Информация, получаемая через MIB, обычно имеет небольшой объём и не привязана к конкретным протоколам. Например, можно получить общее количество пакетов и байт, прошедших через определенный сетевой интерфейс устройства. Развитие MIB и flow–based подходов частично стимулировалось глобальной дискуссией о легальности и этичности глубокого анализа трафика с точки зрения нарушения безопасности и прав на частную жизнь. Одним из результатов этой дискуссии является то, что в научных исследованиях трафик, подвергающийся глубокому анализу, часто предварительно анонимизируется специальными методами. Далее мы более детально рассмотрим отдельные этапы анализа сетевого трафика, включая методы, алгоритмы и подходы, а также обсудим их особенности и ограничения применения» [15].

1.4 Классификация средств мониторинга и анализа

«Тема классификации сетевого трафика представляет собой обширное поле исследований. Прежде чем приступить к методам его анализа, важно рассмотреть различные варианты классификации и результаты, которые могут быть получены с их помощью. Существуют три основных подхода к классификации, которые можно рассматривать как прогрессивные в направлении повышения точности: классификация по типу трафика не всегда достаточно информативна и часто требует дополнительной уточняющей классификации. В различных областях применения могут использоваться разные типы» [17].

Процесс определения протоколов на уровне приложений раскрывает возможности для точной категоризации, особенно в связи с сбором статистических данных и наблюдением за ними.

После этапа определения, основное внимание смещается к анализу протокола, включая сбор данных сессии и извлечение метаданных.

Приложение, в свою очередь, предлагает дополнительные детали для категоризации и обработки данных, улучшая аналитический уровень до конкретных действий и трактовки информации, что критично в разнообразных ситуациях, начиная от обнаружения вредоносных программ и заканчивая созданием пользовательских профилей для целевой рекламы.

Каждая задача на уровне приложения требует уникального подхода к выбору алгоритмов и параметров категоризации, что может существенно сказаться на эффективности системы (рисунок 6).

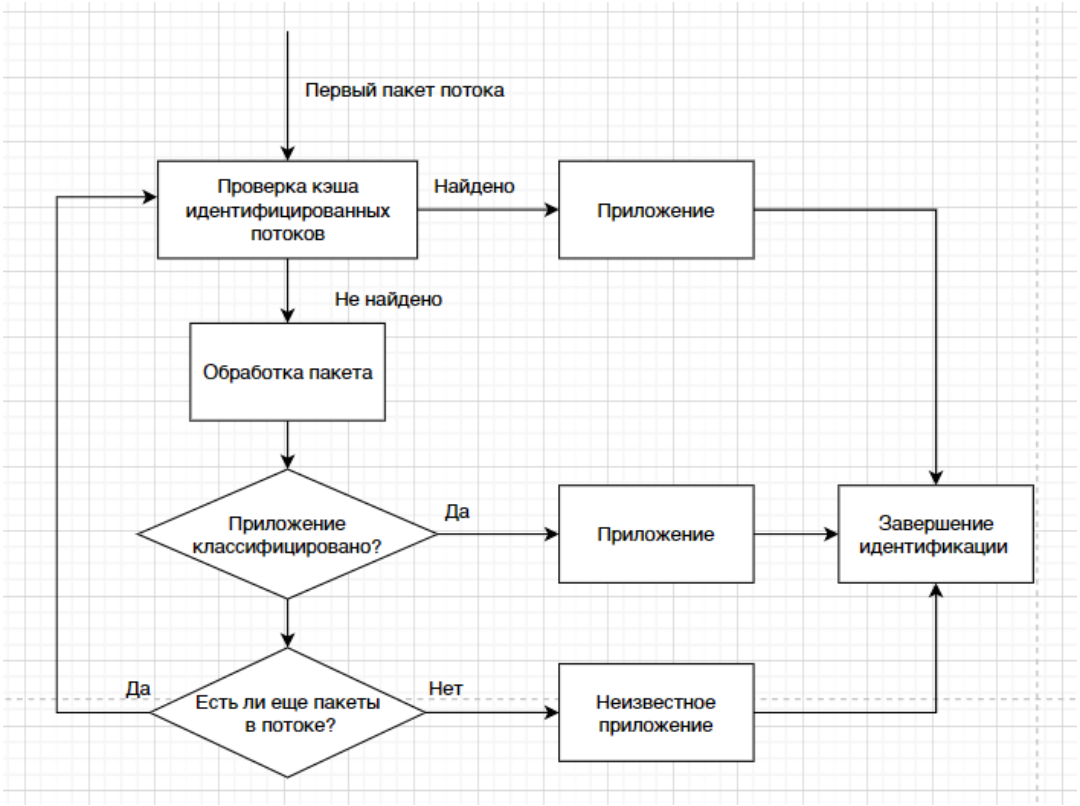


Рисунок. 6 – Схема классификации «до первого срабатывания»

«При использовании систем фильтрации по ключевым словам, этот метод становится неэффективным из–за возможности встречи разных слов в одном сетевом потоке. В таких случаях поток может попадать сразу в несколько категорий по системе классификации. Общепризнано, что первый подход более производителен, поскольку требуется анализировать меньшие объемы данных» [18].

Результаты этих экспериментов демонстрируются на рисунке 7, где неправильно классифицированные случаи обозначаются как "misclassified", а неопределенный трафик помечается как "unknown". Таким образом, выбор размера сегмента для анализа может оказать существенное влияние на достижение необходимой точности и эффективность анализа.

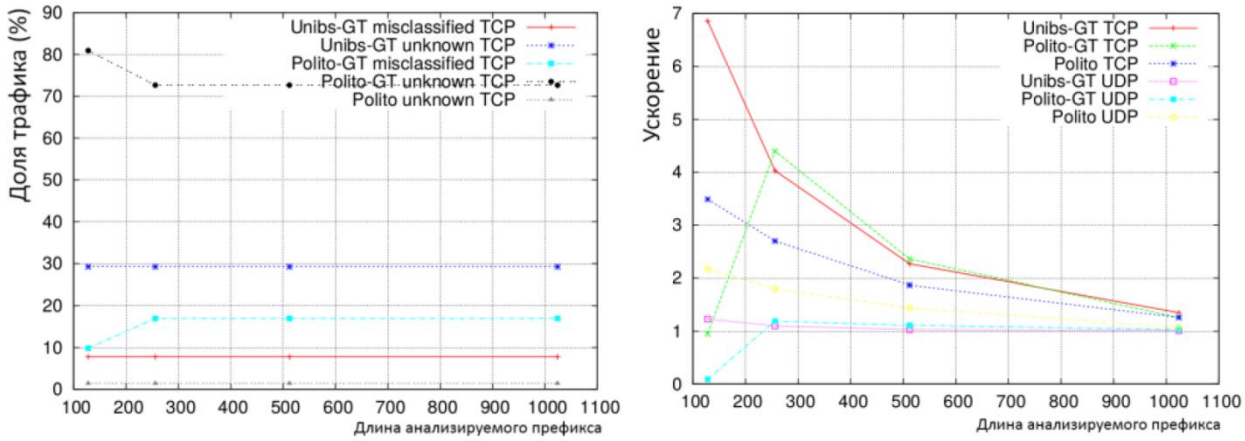


Рисунок 7 – Оценка влияния длины префикса на точность классификации (слева) и скорость (справа).

«Согласно результатам проведенных исследований, оптимизация процессов IP-дефрагментации и нормализации TCP оказывается излишней и затратной в контексте классификации сетевого трафика. Несмотря на высокие требования к вычислительным ресурсам, эти меры практически не повышают точность классификации. Причина заключается в том, что для классификации обычно используются пакеты размером не более 256 байт, в то время как минимальный размер фрагмента IP составляет не менее 576 байт. В такой

ситуации более целесообразным подходом представляется применение метода PBFS (Packet-Based Flow Sampling) вместо MBFS (Measurement-Based Flow Sampling)» [18].

PBFS – это метод, который основан на анализе пакетов, а не на измерении потоков.

Он позволяет более эффективно классифицировать трафик, так как не требует дополнительных ресурсов для обработки фрагментов пакетов.

Пройдя сквозь мозаику разнообразных подходов к категоризации и их реализацию в реальных сценариях, мы направляемся к исследованию специфических алгоритмов, применяемых для классификации сетевого трафика. Оптимальный выбор алгоритма здесь зависит от конкретных потребностей и ограничений, таких как точность, скорость обработки и доступные вычислительные ресурсы.

Некоторые стратегии классификации опираются на детальные "подписи", которые основаны на состояниях протоколов.

«После процесса классификации сообщения сопоставляются с переходами, описанными в автоматах состояний протоколов, чтобы проверить последовательность этих переходов. Этот подход известен как контекстно-ориентированный анализ состояния протокола.

Как показано на рисунке 7, классификация является наиболее затратным алгоритмом при обработке сетевых пакетов.

Ранее, для увеличения производительности при ограниченных вычислительных ресурсах, совершались попытки использовать более компактные источники данных для классификации, сохраняя при этом информативность данных, сопоставимую с содержимым пакетов» [10].

Эти подходы относятся к категории "основанных на выводе" (inference-based), в отличие от подходов, использующих "подписи".

1.5 Исследование и выбор математической модели сетевого трафика

Отсутствие всеобъемлющей математической абстракции, способной в достаточной мере описать динамику сетевого потока, представляет собой один из главных вызовов в современной сетевой инженерии. Этот вызов обусловлен нарастающим объемом передаваемой информации и множеством применяемых механизмов для ее передачи, включая разнообразные протоколы и техники маршрутизации.

Это означает, что для анализа и управления сетевым трафиком необходимо использовать несколько различных моделей, что может привести к сложности и неэффективности анализа.

В связи с этим, в последние годы было разработано несколько новых математических моделей, которые позволяют описывать структуру сетевого трафика более точно. Некоторые из этих моделей основаны на использовании статистических методов, таких как методы машинного обучения, для анализа и прогнозирования трафика. Другие модели используют теории вероятности и статистики для описания трафика.

В целом, отсутствие универсальной математической модели для описания сетевого трафика является одной из основных проблем в области сетевых технологий. Однако, с помощью разработки новых моделей и методов анализа, можно улучшить эффективность и точность анализа сетевого трафика.

«Одной из наиболее широко используемых моделей является классическая модель Пуассона. В этой модели поток данных интерпретируется как независимая случайная величина, чей характер экспоненциально меняется в течение времени сеанса (1):

$$f(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}, \quad (1)$$

где $f(t)$ – плотность распределения; λ – средняя интенсивность потока

за сеанс;

n= 1,2,3… – число потоков трафика (событий) за период времени t.

Эта модель представляет собой достаточно простой подход и может быть использована для анализа трафика с невысоким объемом данных. Однако с увеличением возможностей сетей передачи данных и необходимостью обработки разнообразных типов данных, таких как аудио, видео и другие, стало ясно, что модель Пуассона не обладает достаточной адекватностью для точного описания происходящих процессов» [19].

Математики постоянно стремятся улучшить модель распределения Пуассона, в результате чего появилось несколько уникальных моделей, учитывающих нерегулярности в структуре трафика, которые не могут быть описаны стандартной моделью Пуассона.

Например:

– «модель ON/OFF рассматривает трафик как пуассоновский процесс с перерывами. Её особенность в том, что она учитывает промежутки бездействия системы, когда передача данных приостановлена, что дает более точное описание процесса коммутации пакетов;
– модель, базирующаяся на распределении Пуассона и модулированная марковским процессом, учитывает не только промежутки бездействия системы, но и количество активных пользователей в данный момент времени. Здесь поток данных от каждого пользователя рассматривается как прерывистый пуассоновский процесс, а общий поток данных как марковский процесс» [26].

Эти концепции углубляются в анализе взаимосвязи между пакетами данных, что расширяет наше понимание процесса коммутации пакетов и способствует более точному прогнозированию трафика. Они также учитывают воздействие различных факторов на трафик, включая активное количество пользователей и текущее состояние системы, что способствует более глубокому анализу сетевой динамики.

«Отличительной чертой самоподобных процессов является наличие "тяжелых хвостов" в их распределениях и медленно убывающей зависимости автокорреляционной функции (АКФ). Такие распределения хорошо описываются моделями Парето и Вейбулла. Однако, хотя обе модели обладают степенной зависимостью функции плотности от скорости передачи данных, их форма несколько различается.

– для модели Парето $f(t) = \beta\alpha^{\beta}t^{-\beta-1}$;
– для распределения Вейбулла $f(t) = \alpha\beta^{-\alpha}t^{\alpha-1}e^{-(t/\beta)\alpha}$, где α, β – коэффициенты уравнений.

Одной из наиболее распространенных математических моделей для описания самоподобных процессов является классическое броуновское движение (БД), а также его вариация – модель фрактального броуновского движения (ФБД). В этих моделях поток событий интерпретируется как случайная величина, зависящая не только от времени, но и от предыдущих значений потока (2).

$$X_t = X_{t-1} + \varepsilon_t, \quad (2)$$

где X_t – значение случайной величины в момент времени t;

ε_t – белый шум» [20].

«Плотность распределения приращений такой величины подчиняется закону Гаусса, для которого математическое ожидание равно нулю, а дисперсия (3):

$$\sigma^2(X_2 - X_1) = C(t_2 - t_1)^{2H}, \quad (3)$$

где X_1 X_2 – значения случайной величины в моменты времени t_1 и t_2;

C – некоторая положительная константа;

H – параметр Херста.

Параметр Херста позволяет оценивать степень самоподобия трафика.

При 0,5 ≤ H ≤ 1,0 процесс является строго самоподобным и описывается моделью ФБД, при H = 0,5 – моделью классического БД, а при 0 ≤ H < 0,5 имеет стохастический характер» [25].

Модель скользящего среднего основана на представлении о том, что текущее значение функции является средним из нескольких предыдущих значений.Это позволяет моделировать процесс, который имеет тенденцию к изменению в зависимости от времени.

Обе модели временных рядов могут быть использованы для анализа самоподобных процессов, таких как трафик в сети. Они позволяют учитывать зависимость между значениями функции в различных моментах времени и прогнозировать будущие значения функции на основе прошлых данных (4).

$$X_t = C + \sum_{i=1}^{p} \alpha_i X_{t-i} + \varepsilon_t, \quad (4)$$

где p – порядок модели AR;

α_i – коэффициенты уравнения авторегрессии;

X_{t-i} – лаговый оператор.

«Модель скользящего среднего в машинном обучении предполагает, что значение функции в текущий момент времени изменяется вокруг определенного среднего значения, причем величина отклонения зависит от значений функции в предыдущие временные периоды (5):

$$X_t = C + \varepsilon_t + \sum_{i=1}^{q} \beta_i X_{t-i}, \quad (5)$$

где q – порядок модели MA;

β_i – коэффициенты уравнения скользящего среднего» [21].

«На базе моделей AR и MA сформированы два вида моделей.

Модель ARMA, в которой предыдущие значения функции влияют не только на текущее значение функции, но и на его отклонение (6):

$$X_t = C + \sum_{i=1}^{p} \alpha_i X_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \beta_i X_{t-i}, \quad (6)$$

Модель ARIMA (модель Бокса – Дженкинса, или модель интегрированной авторегрессии – скользящего среднего), которая позволяет работать с зависимостями, имеющими тренд (7):

$$X_t = C + \sum_{i=1}^{p} \alpha_i \Delta^d X_{t-i} + \varepsilon_t + \sum_{i=1}^{q} \beta_i \Delta^d \varepsilon_{t-i}, \quad (7)$$

где d – порядок модели ARIMA, характеризующий степень интегрирования;

Δ^d – оператор взятия конечной разности порядка d» [21].

«При исследовании трафика широко применяются модели временных рядов, однако точное математическое описание трафика остается задачей непростой.

Это связано с тем, что сетевой трафик содержит различные типы данных, требующие применения различных моделей для их описания, а степень самоподобия трафика может изменяться в зависимости от характера передаваемой информации» [13].

Вывод по первому разделу

В первом разделе бакалаврской работы были рассмотрены ключевые аспекты анализа сетевого трафика.

Начав с постановки задачи и рассмотрения общих понятий и определений, мы подчеркнули важность эффективного анализа для обеспечения безопасности и оптимизации работы сетевых систем.

Далее мы осветили различные направления развития анализа сетевого трафика, включая технологии и методы, используемые для обнаружения аномалий, обработки больших объемов данных и выявления угроз безопасности.

Далее представлена общая схема инфраструктурных алгоритмов анализа сетевого трафика, позволяющая понять основные этапы анализа и

взаимосвязь между ними.

Особое внимание уделено классификации средств мониторинга и анализа, что помогает определить подходящие инструменты для конкретных целей и задач исследования.

Завершая раздел, мы проанализировал различные математические модели сетевого трафика, сосредоточившись на выборе наиболее подходящей модели для дальнейшего исследования и реализации алгоритмов анализа.

Обзор и выбор математической модели сетевого трафика является важным шагом, определяющим успешность и точность проводимого анализа.

Таким образом, первый раздел работы представляет собой обширный обзор основных концепций, методов и инструментов, необходимых для понимания и дальнейшего проведения исследования алгоритмов интеллектуального анализа сетевого трафика.

2 Обзор алгоритмов анализа сетевого трафика

2.1 Архитектура разрабатываемой системы

Рассмотрим концептуальную структуру будущей системы обнаружения вторжений.

Основные элементы этой системы включают:

– модуль сбора сетевого трафика, который осуществляет захват данных с сетевых устройств на границе сети, их преобразование в нужный формат и сохранение в хранилище исходных событий;
– «хранилище исходных событий, где хранится информация о сетевом трафике, необходимая для последующего анализа на предмет сетевых атак;
– модуль интеллектуального анализа сетевого трафика, который проводит проверку трафика с применением алгоритмов машинного обучения для выявления аномалий и классификации соединений как нормальных или потенциальных атак;
– хранилище результатов, база данных для хранения выявленных аномалий, используемая для предупреждений и общего анализа безопасности системы;
– консоль управления системой, обеспечивающая интерфейс для общей настройки компонентов программного комплекса;
– система оповещения, которая информирует администратора о обнаруженных инцидентах, предоставляет информацию об аномалиях и возможность создания отчетов;
– графический интерфейс администратора, объединяющий консоль управления и систему оповещений для удобного взаимодействия пользователя с системой (рисунок 8);
– модуль интеграции, который предоставляет API для интеграции с другими системами реагирования через HTTP-запросы» [3].

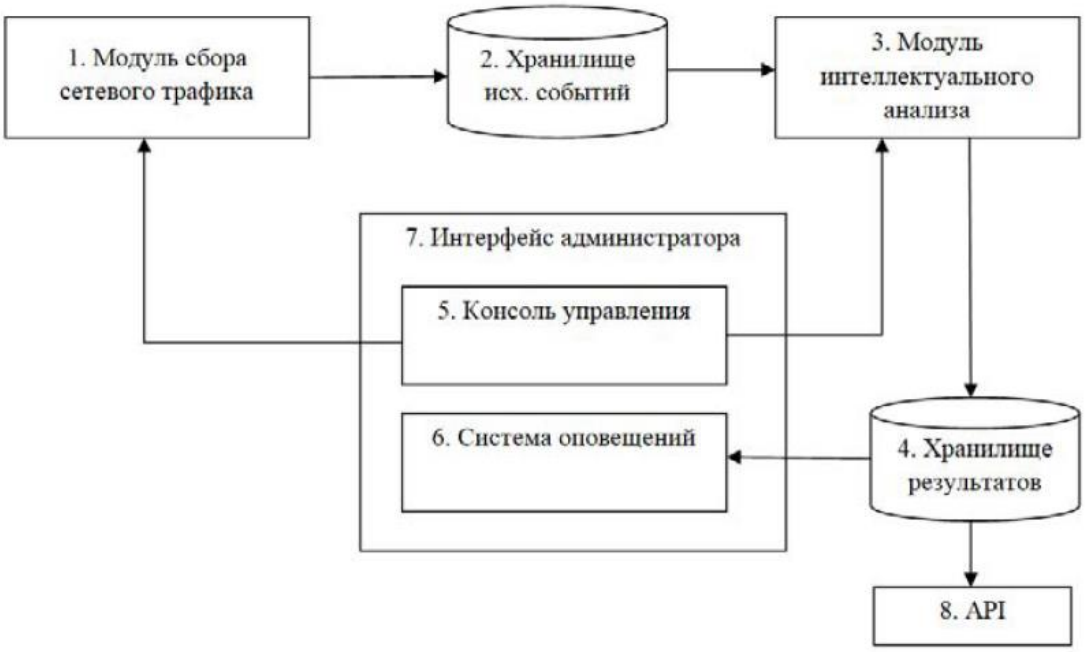


Рисунок 8 – Архитектура разрабатываемой системы обнаружения вторжений

Рассмотрим подходы к разработке одного из главных модулей системы – анализа сетевого трафика с умным подходом. Создание данного модуля, включающего в себя методики и алгоритмы обнаружения, является ключевым моментом в процессе построения интеллектуальной системы защиты от несанкционированных вторжений.

Эффективность обнаружения инцидентов напрямую зависит от функционирования данного компонента, что подчеркивает его важное значение в общей структуре системы.

2.2 Моделирование анализа сетевого трафика

«Модуль для анализа сетевого трафика и выявления атак основан на методах индуктивного машинного обучения, в частности, на использовании искусственных нейронных сетей (ИНС). Для успешного обучения, тестирования и проверки нашей нейронной сети необходимо тщательно

подготовить разнообразные примеры. Задача обнаружения сетевых атак требует выделения множества характеристик, по которым можно классифицировать атаки» [22]. Для этой цели мы используем различные наборы данных, включая результаты исследований из KDD Cup 1999. Наши исследования подтверждают, что наш подход эффективнее аналогичных методов, что открывает новые перспективы в борьбе с кибератаками.

Хотя база данных KDD CUP 1999 Data появилась в заключительном веке прошлого тысячелетия, её данные о сетевых атаках ныне кажутся далекими от актуальности. Однако в 2015 году свет увидела новая подборка данных под названием UNSW–NB15, что стала своего рода исправлением ошибок предшествующих исследований. Этот ансамбль данных, созданный научным сообществом, включает в себя более 2,5 миллиона записей.

Приняв решение о разработке инновационного аналитического модуля, мы обратились к современному набору данных, ориентированному на анализ сетевых инцидентов, включенных в рамках проекта UNSW-NB15.

Этот датасет содержит обширный набор параметров сетевого трафика – целых 47 – представленных в различных форматах: от номинальных до бинарных данных, включая числовые и временные значения.

«Каждая запись в наборе данных также содержит информацию о классификации соединения: оно может быть отнесено к нормальному трафику или одному из девяти типов сетевых атак. Однако, как показано в данных таблицы 1, выборка в базе данных UNSW–NB15 разнообразна по типам соединений: для некоторых видов атак, таких как Worms, количество доступных примеров для обучения оказывается недостаточным» [5].

Таблица 1 – Виды атак, представленные в базе UNSW–NB15

№	Вид соединения	Количество	Описание
1	Normal	2 218 761	Естественные данные транзакций

Продолжение таблицы 1

№	Вид соединения	Количество	Описание
2	Fuzzers	24 246	Попытка привести к остановке программы или сети, подавая на вход случайно созданные данные.
3	Backdoors	2 329	Метод, при котором система обеспечения безопасности обходится незаметно для получения доступа к компьютеру или его данным.
4	Analysis	2 677	Включает в себя многообразие методов проникновения в систему, отправки спама и манипулирования HTML-файлами.
5	DoS	16 353	Это злонамеренная попытка временно вывести сервер или сетевой ресурс из строя для пользователей, чаще всего путем нарушения или остановки работы хоста, подключенного к интернету.
6	Generic	215 481	Этот подход нацелен на различные типы блочного шифрования, независимо от размера блока или используемого ключа.
7	Exploits	44 525	Хакер хорошо осведомлен о проблемах безопасности в системе и активно эксплуатирует эти уязвимости для достижения своих целей.
8	Reconnaissance	13 987	Включает в себя различные виды атак, направленных на сбор информации о сети с целью разведки.
9	Worms	174	Хакер копирует свою систему, чтобы заразить другие компьютеры. Он часто использует компьютерные сети для распространения, исходя из имеющихся уязвимостей в безопасности целевой системы с целью ее захвата.
10	Shellcode	1 511	Этот кодовый фрагмент представляет собой полезную нагрузку, используемую для эксплуатации уязвимостей программного обеспечения.

Таким образом, UNSW–NB15 представляет собой современный и разнообразный набор данных, который может быть использован для создания эффективного интеллектуального аналитического модуля.

2.3 Проектирование с полным набором параметров и определение их значимости

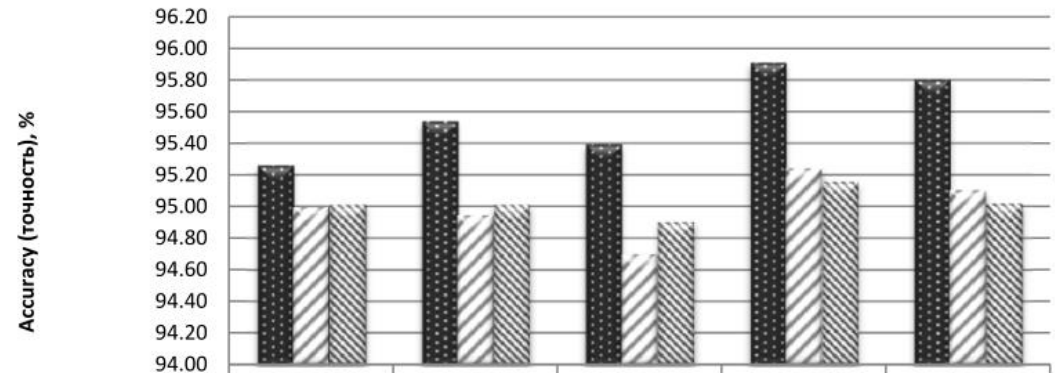
«В ходе разработки, базирующейся на наборе данных UNSW-NB1, мы использовали пакет Statistica в сочетании с инструментами Automated Neural Networks.

Количество нейронов на входном и выходном слоях было адаптировано под характеристики UNSW-NB1, в то время как для определения структуры скрытого слоя мы применили систематический метод перебора различных вариантов с минимальной ошибкой на этапах обучения, тестирования и проверки, опираясь на теоремы, такие как теоремы Арнольда-Колмогорова-Хехт-Нильсена» [12].

В результате мы создали нейронные сети с использованием пакета Statistica, способные анализировать и классифицировать сетевой трафик на основе набора данных UNSW-NB1.

Настройка нейронных сетей проводилась с учетом 45 признаков сетевого трафика и обучение проходило на разнообразных наборах входных данных.

На рисунке 9 представлены лучшие варианты нейронных сетей, обученных на входном наборе данных, содержащем 100 000 записей. Для оценки производительности модели была использована метрика Accuracy (точность), которая отражает процент правильных классификаций, сделанных классификатором на обучающем, тестовом и проверочном наборах данных.



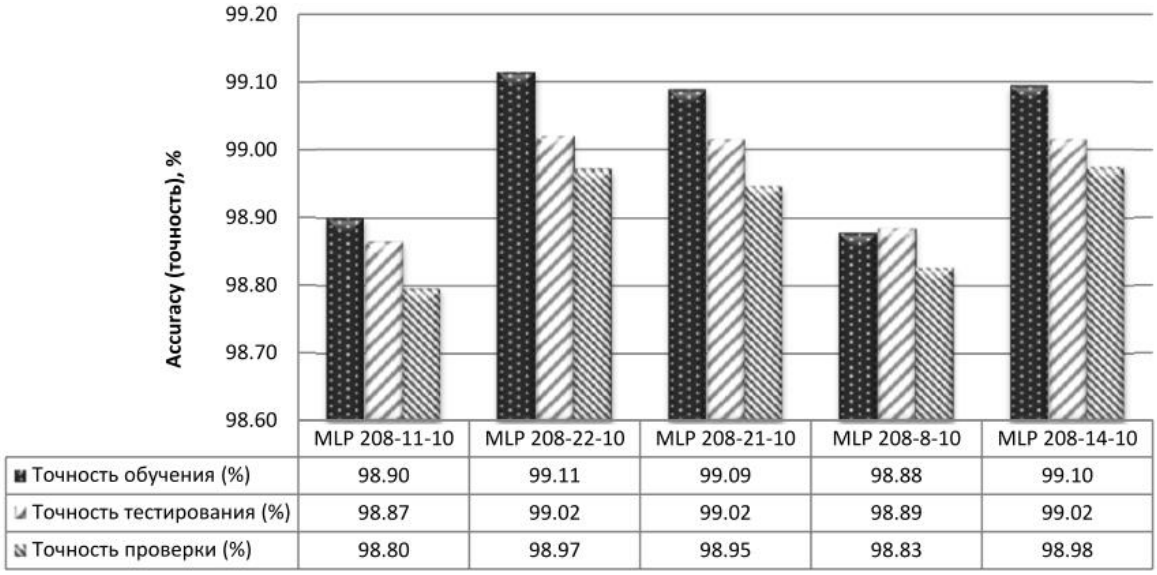
	MLP 208-17-10	MLP 208-9-10	MLP 208-12-10	MLP 208-20-10	MLP 208-16-10
Точность обучения (%)	95.26	95.54	95.40	95.92	95.81
Точность тестирования (%)	95.00	94.95	94.70	95.25	95.11
Точность проверки (%)	95.01	95.01	94.91	95.16	95.02

Рисунок 9 – Характеристики с полным набором параметров с использованием обучающего множества UNSW–NB1 мощностью 100 000 записей

«При увеличении объема входных данных в семь раз мы отметили значительное повышение эффективности процессов обучения, тестирования и проверки. Особенно впечатляющие результаты, проявленные в минимальной ошибке проверки, продемонстрировала нейронная сеть MLP 208–14–10, где числа 208, 14 и 10 отражают количество нейронов на входном, скрытом и выходном слоях соответственно.

Таким образом, увеличение объема входных данных привело к значительному улучшению эффективности обучения, тестирования и проверки нейронных сетей. Нейронная сеть MLP 208–14–10 продемонстрировала наилучшие результаты, достигнув уровня более 98,5% для каждой из пяти лучших нейронных сетей.

Каждая из пяти наилучших нейронных сетей, изображенных на рисунке 10, теперь достигает точности свыше 98,5%» [23].



	MLP 208-11-10	MLP 208-22-10	MLP 208-21-10	MLP 208-8-10	MLP 208-14-10
Точность обучения (%)	98.90	99.11	99.09	98.88	99.10
Точность тестирования (%)	98.87	99.02	99.02	98.89	99.02
Точность проверки (%)	98.80	98.97	98.95	98.83	98.98

Рисунок 10 – Характеристики с полным набором параметров с использованием обучающего множества UNSW–NB1 мощностью 700 000 записей

Данные, полученные в ходе исследования, свидетельствуют о том, что даже при высокой точности проверки нейронной сети MLP 208–14–10 обнаружение конкретных видов атак остается нетривиальной задачей.

Только 6 из 10 типов атак удается определить с точностью, превышающей 66%. Замечательно, что обнаружение нормального трафика происходит верно в подавляющем большинстве случаев (99,08%), однако точность классификации атак по их типу в среднем составляет лишь 75,96%, что говорит о динамике и сложности данной задач.

Результаты классификации атак по типам для этой сети представлены на рисунке 11.

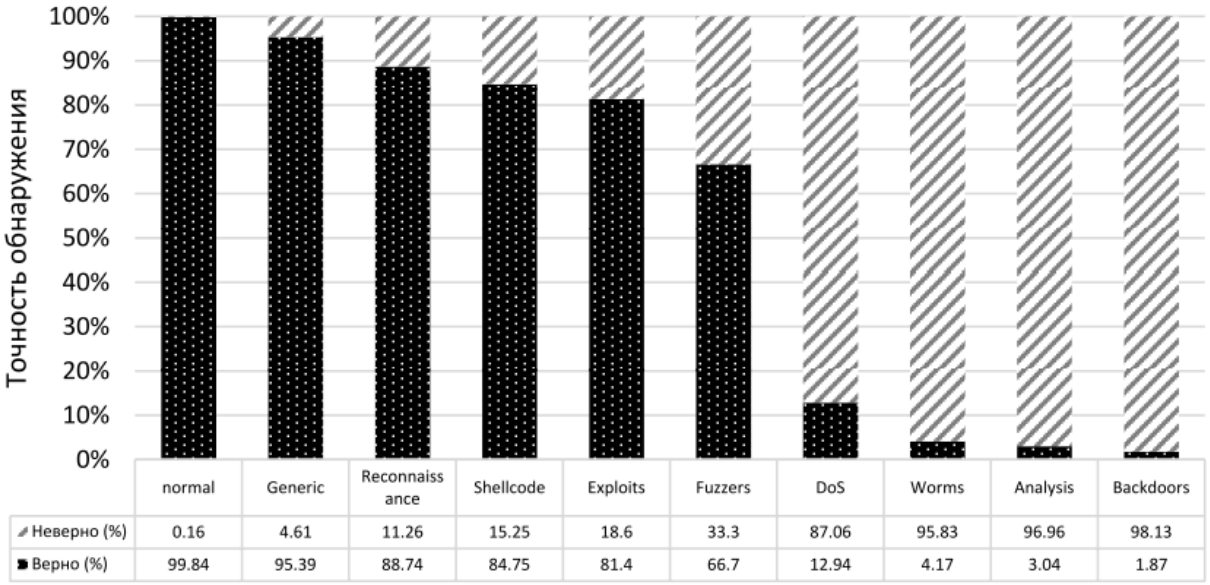


Рисунок 11 – Результаты проверки сети MLP 208–14–10, сгруппированные по видам атак

«Следующим этапом будет исследование по сокращению количества входных параметров нейронной сети, требующее анализа входных данных из набора UNSW–NB15. Методика определения важности признаков UNSW–NB15 аналогична процессу KDD Cup. Начальные шаги включают оценку информативности параметров для пяти нейронных сетей, построенных на полном и сокращенном объеме входных данных.

Затем вычисляется средняя важность каждого признака для этих сетей. Для выявления наименее значимых параметров вычисляются средние коэффициенты важности для обеих групп сетей. Далее, с использованием коэффициентов линейной корреляции, определяются группы линейно–зависимых параметров» [23].

2.4 Проектирование с сокращенным набором параметров и определение их значимости

Экспериментальные изыскания выявили определенные параметры, которые можно убрать из арсенала при разработке нейронных сетей. В

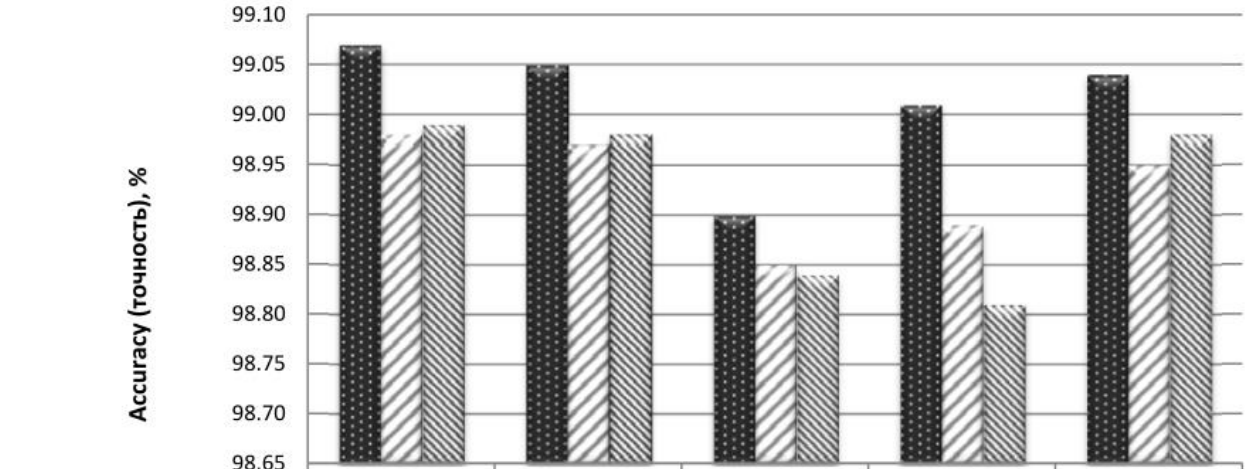
сравнении с полными наборами этих параметров, упрощенные модели сетей будут фильтровать определенные характеристики, отмеченные и описанные в таблице 2.

Таблица 2 – Наименее значимые параметры множества UNSW–NB15

Признак	Описание признака
dbytes	Объем данных, отправленных получателем обратно к отправителю.
ackdat	Интервал между приемом пакета SYN_ACK и отправкой пакета ACK в сеансе TCP.
sintpkt	Время, затраченное на доставку промежуточного пакета от отправителя (мс).
tcprtt	Интервал времени, затраченный на установку TCP–сессии с момента отправки первого сегмента SYN до приема первого пакета данных.
res_bdy_len	Реальный объем данных без сжатия, передаваемых сервером через службу HTTP.
dur	Общая продолжительность соединения
ct_ftp_cmd	Число потоков, задействованных при выполнении команд в сеансе FTP.
dloss	Пакеты получателя повторно переданы или удалены
sloss	Исходящие пакеты повторно переданы или удалены
sbytes	Объем данных, переданных от отправителя к получателю в байтах.
spkts	Число пакетов, отправленных от отправителя и полученных получателем.
ltime	Размер окна TCP, указанный получателем.
dwin	Момент завершения записи.
djit	Джиттер назначения (мс)

Мы осуществили ряд исследований, в рамках которых снизили число входных нейронов в нейронных сетях до 32. Это значительно улучшило процесс обучения и проверки сети, а также упростило ее последующее использование.

Результаты этих экспериментов представлены на рисунке 12.



	MLP 194-20-10	MLP 194-12-10	MLP 194-7-10	MLP 194-17-10	MLP 194-13-10
Точность обучения (%)	99.07	99.05	98.90	99.01	99.04
Точность тестирования (%)	98.98	98.97	98.85	98.89	98.95
Точность проверки (%)	98.99	98.98	98.84	98.81	98.98

Рисунок 12 – Характеристики с сокращенным набором параметров с использованием обучающего множества UNSW–NB1

Изучение выводов из исследования однозначно указывает на эффективное выделение наименее важных параметров, поскольку точность обучения, тестирования и проверки для всех пяти созданных нейронных сетей превысила отметку в 98,8%.

«Нейронная сеть MLP 194–20–10 (где 194, 20 и 10 – количество нейронов на входном, скрытом и выходном слое соответственно) продемонстрировала наименьшую ошибку на проверочном наборе данных. Эта модель оптимизирована по числу параметров и будет использована при разработке программного модуля для анализа сетевого трафика с использованием искусственного интеллекта» [23].

Вывод по второму разделу

Во второй разделе работы были представлены более детальные аспекты архитектуры разрабатываемой системы и ее моделирование. Мы начали с описания архитектуры системы, подробно рассматривая компоненты, взаимодействие между ними и общий принцип работы.

Далее мы представали процесс моделирования анализа сетевого трафика, включающий выбор и адаптацию соответствующих математических моделей, учет особенностей сетевого окружения и определение параметров для анализа.

Также мы рассмотрели методики проектирования с сокращенным набором параметров и определения их значимости, что помогает снизить вычислительную сложность алгоритмов и улучшить их эффективность.

В результате, второй раздел работы представляет собой важный этап в разработке и реализации алгоритмов интеллектуального анализа сетевого трафика.

Детальное описание структуры системы и подходов к моделированию способствует углубленному пониманию процесса исследования и анализа сетевого трафика, а также обосновывает выбор конкретных методов и стратегий.

3 Реализация и тестирование модуля интеллектуального анализа

3.1 Реализация и тестирование модуля интеллектуального анализа

На предыдущем этапе исследований с помощью инструмента SANN была создана модель MLP 194–20–10, которая умеет распознавать 10 видов атак с точностью около 99%. После обучения модели ее параметры можно сохранить в XML–файле, что позволяет использовать ее в других приложениях.

Эта модель послужила основой для разработки интеллектуального аналитического модуля. Для обеспечения связи между этим модулем и хранилищем данных мы разработали специализированную программу-посредника. Ее главная задача – обеспечить передачу данных между хранилищем и аналитическим модулем.

На данном этапе исследования наша цель не включала в себя реализацию хранилища данных, поэтому создание этой программы-посредника необходимо для обеспечения работы аналитического модуля, который не зависит от конкретного способа хранения событий.

Это позволит модулю анализа получать данные из различных источников и обрабатывать их независимо от метода хранения (рисунок 13).

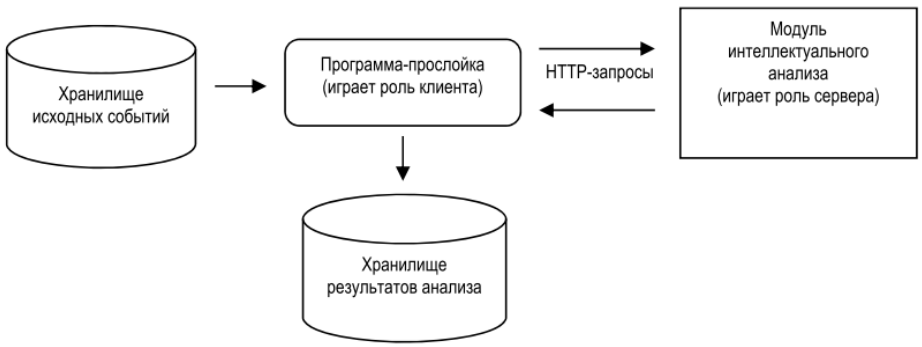


Рисунок 13 – Схема взаимодействия модуля анализа с хранилищами данных системы

Для наглядного демонстрирования функционирования модуля интеллектуального анализа сетевого трафика мы разработали структурное изображение в форме блок-схемы, которое можно увидеть на рисунке 14.

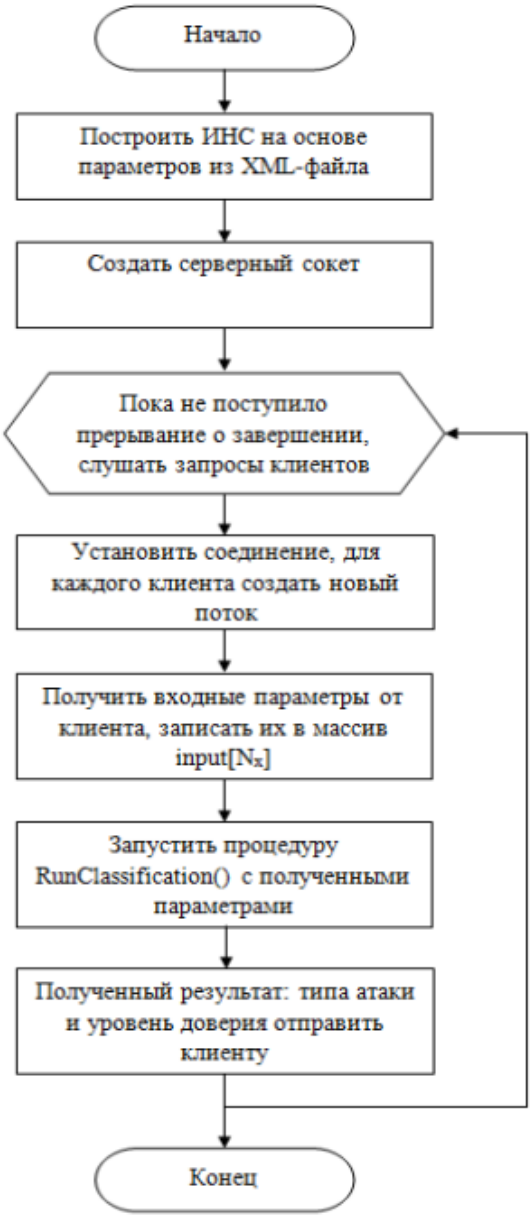


Рисунок 14 – Принцип функционирования модуля интеллектуального анализа

Наша нейронная сеть была спроектирована для анализа сетевого трафика, учитывая различные параметры, такие как тип пакета, его размер, а также время отправки и получения пакета. Она способна обнаруживать аномалии в трафике и предсказывать будущие события, что позволяет улучшить безопасность сети и уменьшить риск атак.

«Разработанное приложение прошло тестирование на множестве записей из набора данных UNSW–NB15, которое не использовалось при создании нейросетевой модели для обучения и тестирования. Это означает, что при тестировании результаты анализа известны для каждой записи из UNSW–NB15, но нейронная сеть, на которой основан модуль анализа, не обучалась на этих данных, поэтому они считаются "неизвестными" для нее» [24].

Производили эксперименты с приложением, используя выборку из 10 000 записей.

Сводные результаты тестирования представлены на рисунке 15.

Показатели	**Вид атаки**										
	Fuzzers	Analysis	Backdoors	DoS	Exploits	Generic	Reconnaissance	Shellcode	Worms	Normal	
Общее кол-во записей	504	437	263	531	542	752	165	209	14	6583	**Precision (точность),%**
Fuzzers	354	208	114	126	53	4	2	2	0	7	**40,69**
Analysis	0	18	0	0	2	0	0	0	0	2	**81,82**
Backdoors	1	7	7	1	1	0	0	2	0	0	**36,84**
DoS	3	4	6	68	8	3	1	0	1	4	**69,39**
Exploits	51	158	129	323	440	23	12	18	7	3	**37,80**
Generic	0	0	1	11	11	717	0	6	3	0	**95,73**
Reconnaissance	2	0	1	1	13	2	146	0	0	1	**87,95**
Shellcode	1	0	5	0	4	1	0	179	0	0	**94,21**
Worms	0	0	0	0	0	0	0	0	1	0	**100,00**
Normal	92	42	0	1	10	2	4	2	2	6566	**97,69**
Recall (полнота), %	**70,24**	**4,12**	**2,66**	**12,81**	**81,18**	**95,35**	**88,48**	**85,65**	**7,14**	**99,74**	

Рисунок 15 – Результаты тестирования на множестве

В начале таблицы отражен тип атаки, подаваемой на вход.

Первый столбец содержит описание формы атаки.

Анализируя рисунок 15, можно заметить, что «нейронная сеть и связанный с ней модуль интеллектуального анализа успешно определяют легальные соединения без атак (тип normal) с высокой точностью.

Однако, возникают трудности с классификацией некоторых видов атак: мы можем точно определить 6 из 9 видов атак с точностью не менее 69% и 5 из 9 видов с полнотой не менее 70%.

Проблемы возникают при классификации атак Analysis, Backdoors, DoS, Worms, которые часто ошибочно классифицируются как Exploits из–за использования известных уязвимостей» [12]. Это объясняется тем, что эти атаки в основном используют уязвимости системы, то есть, Exploits.

Изучение результатов проверки выявило, что интеллектуальный анализатор сетевого трафика обладает высокой точностью в определении паттернов безопасного поведения, когда атаки отсутствуют.

Процент ошибок первого рода составляет всего 0,16%, что означает редкое ложное срабатывание на наличие атаки, когда ее на самом деле нет. Ошибка второго рода составляет 4,48%, что подразумевает некоторые пропуски в обнаружении атак при их реальном наличии.

Однако не все типы атак могут быть корректно классифицированы в случае сетевых инцидентов.

Разработанное приложение на основе MLP 194–20–10 обеспечивает высокую полноту в классификации следующих типов сетевых соединений: Fuzzers, Exploits, Generic, Reconnaissance, Shellcode и обычные соединения.

Вывод по третьему разделу

В разделе третьем нашего исследования мы разработали и проверили модуль интеллектуального анализа.

Создали структуру программного комплекса для системы обнаружения

вторжений, интегрировав в нее модуль анализа сетевого трафика.

Мы использовали упрощенные данные из базы UNSW NB–15 для обучения.

Настроили двухслойный персептрон MLP 194–20–10 с 32 входными параметрами для классификации различных типов сетевых соединений: Fuzzers,

Exploits, Generic, Reconnaissance, Shellcode и нормальные соединения.

Как результат, ложная тревога об атаке при ее отсутствии составила 0,16%, а пропуск реальной атаки – 4,48%.

Это означает, что модуль интеллектуального анализа имеет высокую точность и эффективность в обнаружении различных видов сетевых атак.

Разработанное приложение на основе MLP 194–20–10 обеспечивает высокую полноту в классификации следующих типов сетевых соединений: Fuzzers, Exploits, Generic, Reconnaissance, Shellcode и обычные соединения.

Заключение

В данной выпускной квалификационной работе исследуется актуальная проблема, связанная с исследованием и реализацией алгоритмов интеллектуального анализа сетевого трафика

В первом рахделе бакалаврской работы мы рассмотрели ключевые аспекты анализа сетевого трафика. Начнем с обозначения задачи и тщательного рассмотрения ключевых понятий и определений, акцентируя внимание на значимости эффективного анализа для обеспечения безопасности и оптимизации работы сетевых инфраструктур.

Затем мы пролили свет на разнообразные направления развития анализа сетевого трафика, охватывая технологии и методы, применяемые для обнаружения аномалий, обработки огромных объемов данных и выявления потенциальных угроз безопасности.

В последующем развернута общая схема инфраструктурных алгоритмов анализа сетевого трафика, что позволяет ясно понять этапы анализа и связь между ними. Особое внимание уделено классификации средств мониторинга и анализа, что помогает определить подходящие инструменты для конкретных целей и задач исследования.

Во втором разделе работы были представлены более детальные аспекты архитектуры разрабатываемой системы и ее моделирование. Мы начали с описания архитектуры системы, подробно рассматривая компоненты, взаимодействие между ними и общий принцип работы.

Мы представили процесс моделирования анализа сетевого трафика, включающий выбор и адаптацию соответствующих математических моделей, учет особенностей сетевого окружения и определение параметров для анализа. Также мы рассмотрели методики проектирования с сокращенным набором параметров и определения их значимости, что помогает снизить вычислительную сложность алгоритмов и улучшить их эффективность.

В третьем разделе мы реализовали и протестировали модуль

интеллектуального анализа.

Мы разработали концепцию программной архитектуры для системы мониторинга безопасности, где мы реализовали инновационный модуль анализа сетевого трафика.

В ходе проекта мы использовали сжатые данные из базы UNSW NB–15 для настройки двухслойного персептрона MLP 194–20–10, который оборудован 32 параметрами входа и способен идентифицировать разнообразные типы сетевых соединений: от Fuzzers и Exploits до Generic, Reconnaissance, Shellcode и обычных соединений.

В результате наших тестов мы установили, что уровень ложных тревог (сигнализация о наличии атаки при ее отсутствии) составил всего лишь 0,16%, а пропуск реальных атак – 4,48%. Эти результаты говорят о высокой точности классификации соединений сетевого трафика моделью персептр

Список используемой литературы и используемых источников

1. Аль–Фукаха, А., Гизани, М., Мохаммади, М., Аледхари, М., и Айяш, М. (2015). Интернет вещей: обзор технологий, протоколов и приложений. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376.

2. Вэй Ю. и Ву Ю. (2019). Обзор анализа сетевого трафика и майнинга. Журнал IEEE Communications, 57(5), 22–29.

3. Гершман А.Б., Киселева А.В. (2014). Исследование и реализация алгоритмов интеллектуального анализа сетевого трафика. Материалы 22–й Международной конференции IEEE по компьютерному анализу, стр. 295–301.

4. Емельянова Ю.Г., Талалаев А.А., Тищенко И.П., Фраленко В.П. Нейросетевая технология обнаружения сетевых атак на информационные ресурсы // Программные системы: Теория и приложения. 2011. № 3(7). С.3–15.

5. Жигулин П.В., Мальцев А.В., Мельников М.А., Подворчан Д.Э. Анализ сетевого трафика на основе нейронных сетей // Электронные средства и системы управления. 2013. №2. С.44–48.

6. Зубков Е.В. Алгоритмы и методики интеллектуального анализа событий информационной безопасности в сетях и системах телекоммуникаций: диссертация кандидата технических наук. Сибирский государственный университет телекоммуникаций и информатики, Новосибирск, 2016. 179 с.

7. Киселев А.В., Гершман А.Б. (2016). Исследование и применение искусственных нейронных сетей для анализа сетевого трафика. Материалы 24–й Международной конференции IEEE по компьютерному анализу, стр. 241–247.

8. Киселев А.В., Гершман А.Б., Соснин О.В. (2015). Исследование и внедрение интеллектуальных систем анализа сетевого трафика. Международный журнал компьютерных наук и сетевой безопасности, 15(2), стр. 60–67.

9. Мустафаев А.Г. Нейросетевая система обнаружения компьютерных

атак на основе анализа сетевого трафика // Вопросы безопасности. 2016. № 2. С.1–7. [Электронный ресурс] URL: http://e–notabene.ru/nb/article_ 18834.html (дата обращения: 03.03.2024).

10. Насир М., Аббас Х. и Али С. (2019). Интеллектуальный анализ сетевого трафика: всесторонний обзор. Системы, основанные на знаниях, 178, 1–12.

11. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4–е изд. – СПб.: Питер, 2016. 996с.

12. Суворова В.А. Разработка приложения для обнаружения и классификации атак на основе нейросетевой модели. Ломоносов – 2017: XXIV Международная научная конференция студентов, аспирантов, молодых ученых: сб. тезисов. М.: Издательский отдел факультета ВМиК МГУ, 2017. С. 117–119.

13. Сяо, Х., и Ван, Ф. (2018). Достижения в алгоритмах интеллектуального анализа сетевого трафика. Компьютеры и электротехника, 65, 19–38.

14. Тимофеев А.В., Броницкий А.А. Исследование и моделирование нейросетевого метода обнаружения и классификации сетевых атак // International Journal «Information Technologies & Knowledge» Vol.6, Number 3, 2012. С.257–265.

15. Чен Ю., Лю Г., Чжу С. и Чжоу Г. (2020). Опрос по глубокому обучению для классификации сетевого трафика. IEEE Access, 8, 161908–161920.

16. Чжан С., Ван С., Си Д. и Ли Дж. (2020). Последние достижения в анализе сетевого трафика для обнаружения подозрительного поведения. Доступ IEEE, 8, 36890–36911.

17. Чжоу Ю. и Ву Ю. (2019). Интеллектуальный анализ сетевого трафика на основе глубокого обучения: опрос. Системы, основанные на знаниях, 169, 416–427.

18. Читтори Дж., Джха Г. и Пех Л. (2020). Обзор методов анализа

сетевого трафика. Препринт arXiv arXiv: 2003.06712.

19. Шаньгин В.Ф. Информационная безопасность компьютерных систе и сетей; учеб. пособие. – М.: ИД «ФОРУМ»: ИНФРА–М, 2011. 416 с.

20. Ясницкий Л.Н. Интеллектуальные системы. М.: Лаборатория знаний, 2016. 221 с.

21. A Hardware Platform for Network Intrusion Detection and Prevention. http://www.cc.gatech.edu/home/wenke/papers/np3.pdf, дата обращения 09.03.2024.

22. Andrew M White, Srinivas Krishnan, Michael Bailey, Fabian Monrose, and Phillip Porras. Clear and Present Data: Opaque Traffic and its Security Implications for the Future. NDSS, 2013.

23. David L. Cannon. CISA Certified Information Systems Auditor Study Guide, 2nd Edition, 2008, ISBN: 978–0–470–23152–4

24. G NetFPGA. https://github.com/NetFPGA/netfpga/wiki/G, дата обращения 24.13.2024.

25. ICAP. https://tools.ietf.org/html/rfc3507, дата обращения 24.03.2024.

26. Shaly Laurence, Anuros Thomas K. Review of SRAM based architecture for TCAM. International Journal of Advanced Research Trends in Engineering and Technology (IJARTET) Vol. II, Special Issue VI, February 2015.

Приложение А

Реализация алгоритмов

```
from scapy.all import

def analyze_traffic(packet):
if IP in packet:
src_ip = packet[IP].src
dst_ip = packet[IP].dst
print(f"IP Packet: {src_ip} --> {dst_ip}")

if TCP in packet:
src_port = packet[TCP].sport
dst_port = packet[TCP].dport
print(f"TCP Packet: {src_ip}:{src_port} --> {dst_ip}:{dst_port}")

elif UDP in packet:
src_port = packet[UDP].sport
dst_port = packet[UDP].dport
print(f"UDP Packet: {src_ip}:{src_port} --> {dst_ip}:{dst_port}")

elif ICMP in packet:
icmp_type = packet[ICMP].type
icmp_code = packet[ICMP].code
print(f"ICMP Packet: {src_ip} --> {dst_ip} Type: {icmp_type} Code: {icmp_code}")

else:
print(f"Other IP Packet: {src_ip} --> {dst_ip}")
```

```
else:
print("Non-IP Packet")
def main():
sniff(prn=analyze_traffic, count=10)
if __name__ == "__main__":
main()
```