Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий - РТФ

Школа профессионального и академического образования

ДОПУСТИТЬ К ЗАЩИТЕ ПЕРЕД ГЭК

РОП 09.04.03 Медведева М.А.
_____
«_____»_____2024 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

«Внедрение моделей машинного обучения в сетевую инфраструктуру для обнаружения и классификации вредоносного программного обеспечения в среде Интернета вещей»

Научный руководитель Агбозо Э.
старший преподаватель

Научный руководитель: Балунгу Д.М.
ассистент

Студент группы
РИМ-220981 Иванов К.В.

Екатеринбург
2024

РЕФЕРАТ

Тема магистерской диссертации:
«Обнаружение и классификация вредоносного программного обеспечения в среде IoT методами глубокого обучения»
Магистерская диссертация выполнена на 63 страницах, содержит 6 таблиц, 14 рисунков, 24 использованных источников.

С развитием Интернета вещей (IoT) растет число подключенных устройств, что увеличивает риск заражения вредоносным ПО. Традиционные методы обнаружения угроз не справляются с объемом и сложностью атак в IoT-среде. Использование машинного обучения (ML) для обнаружения и классификации вредоносного ПО в сетевой инфраструктуре IoT актуально, так как позволяет более эффективно выявлять угрозы и реагировать на них. Целью данной магистерской диссертации является разработка и внедрение моделей машинного обучения в сетевую инфраструктуру для повышения эффективности обнаружения и классификации вредоносного программного обеспечения в среде Интернета вещей.

Целью диссертации является разработка и внедрение моделей ML для повышения эффективности обнаружения и классификации вредоносного ПО в IoT-среде.

Объектом исследования является сетевая инфраструктура IoT, подверженная угрозам со стороны вредоносного ПО.

Предметом исследования являются модели ML для обнаружения и классификации вредоносного ПО в IoT-сетях.

Научная новизна заключается в разработке новых подходов к использованию ML для защиты IoT-сетей от кибератак. Предложены методы для более точного и оперативного обнаружения новых видов вредоносного ПО и адаптации моделей к изменяющимся условиям сети.

Практическая значимость работы заключается в повышении безопасности IoT-устройств и сетей за счет внедрения разработанных ML-моделей. Это

позволит своевременно выявлять и предотвращать кибератаки, минимизировать ущерб и повысить доверие к IoT-технологиям в различных сферах.

Внедрение предложенных мер обладает высокой экономической эффективностью. Сокращаются затраты на устранение последствий кибератак, снижаются риски простоя оборудования и потери данных. Повышение уровня безопасности снижает расходы на страхование информационных рисков и улучшает репутацию компаний. В долгосрочной перспективе это способствует более устойчивому развитию бизнеса и снижению совокупной стоимости владения IT-инфраструктурой.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 7
1 ОБЗОР И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 9
1.1 Актуальность 9
1.1.1 Литературный обзор 9
1.1.2 Связанные работы 10
1.1.3 Наукометрические показатели 14
1.2 Методология 14
1.2.1 Методы проведения литературного обзора 14
1.2.2 Методы анализа патентной документации 16
1.2.3 Параметры анализа патентной документации 17
1.3 Результаты исследования 20
1.4 Постановка задачи управления 23
1.5 Подведение итогов исследования 30
2 ОБЗОР СУЩЕСТВУЮЩИХ ИНСТРУМЕНТОВ ОБНАРУЖЕНИЯ И КЛАССИФИКАЦИИ ВРЕДОНОСНОГО ПО В СРЕДЕ IOT 32
2.1 Традиционные системы обнаружения вредоносного ПО 32
2.1.1 Системы обнаружения вторжений (IDS) на основе сигнатур 32
2.1.2 Системы обнаружения на основе аномалий 32
2.1.3 Гибридные системы 33
2.2 Ограниченность традиционных систем обнаружения вторжений 34
2.3 Системы обнаружения вторжений на основе машинного обучения 35
2.3.1 Алгоритмы контролируемого обучения 36
2.3.2 Алгоритмы неконтролируемого обучения 37
2.4 Особенности среды IoT и её уязвимости перед вредоносным ПО 37
2.5 Особенности среды IoT и её уязвимости перед вредоносным ПО 39
2.5.1 Архитектура IoT и её компоненты 39
2.5.2 Типичные угрозы и атаки в среде IoT 40
2.5.3 Атаки на аппаратное обеспечение 41
2.5.4 Атаки на программное обеспечение 42
2.5.5 Атаки на коммуникационные каналы 42
2.5.6 Ограничения IoT-устройств и их влияние на безопасность 43
2.6 Интеграции предлагаемого инструмента 44
2.6.1 Топологии сетевой инфраструктуры предприятий 44

2.6.2 Средства для сбора данных о сетевом трафике 45
2.6.3 ПО для подготовки данных Zeek 47
2.6.4 Выбор используемой модели 47
2.6.5 Краткое описание процесса работы 48
2.6.6 Описание процесса тестирования моделей 48
2.6.7 Оцениваемые параметры 48
2.6.8 Ожидаемые результаты тестирования 49
3 ВНЕДРЕНИЕ ПРЕДЛОЖЕННОГО ПОДХОДА НА ПРЕДПРИЯТИИ 49
3.1 Введение 49
3.2 Бизнес-процессы и пользователи 50
3.2.1 Анализ текущих бизнес-процессов 51
3.2.2 Идентификация пользователей 51
3.3 Тестирование модели 52
3.3.1 Подготовка оборудования 52
3.3.2 Описание тестового стенда 52
3.3.3 Анализ входных параметров для работы модели 53
3.3.4 ПО используемые для подготовки данных 55
3.3.5 Процесс тестирования 57
3.3.6 Интеграция в инфраструктуру 59
3.3.7 Вывод по работе системы 59
3.3.8 Вывод по интеграции системы 60
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 61

ВВЕДЕНИЕ

В последние годы Интернет вещей (IoT) стал неотъемлемой частью современных информационных технологий, предоставляя возможности для автоматизации и улучшения различных аспектов жизни человека. Устройства IoT применяются в различных сферах, таких как умные дома, промышленность, здравоохранение и транспорт. Однако вместе с ростом популярности и распространения IoT возрастает и количество киберугроз, направленных на эти устройства и сети, к которым они подключены. Это делает вопросы обеспечения безопасности и защиты от вторжений особенно актуальными.

Целью данной магистерской диссертации является разработка методов и алгоритмов для систем обнаружения вторжений (IDS), которые способны эффективно защищать сети Интернета вещей. Внедрение таких систем позволит повысить уровень безопасности IoT-сетей, минимизируя риски кибератак и утечек данных.

Для достижения поставленной цели необходимо решить следующие задачи:

- Анализ существующих методик и инструментов обнаружения вторжений, применяемых в сетях IoT.
- Определение уязвимостей и угроз, характерных для IoT-устройств и сетей.
- Разработка и тестирование новых инструментов обнаружения вторжений, учитывающих специфические особенности сетей IoT.
- Внедрение и оценка эффективности предложенных решений в реальных сетях Интернета вещей.
- Сравнительный анализ предложенных методов с существующими подходами в контексте точности обнаружения и производительности.

Научная новизна данной работы заключается в разработке инновационных методик и инструментов для систем обнаружения вторжений, специально адаптированных для особенностей сетей Интернета вещей. Эти методики и

инструменты призваны учитывать разнообразие типов данных и протоколов, а также динамический характер IoT-сетей. Результаты данной работы будут способствовать повышению надежности и безопасности IoT-сетей, что имеет важное значение для дальнейшего развития и широкомасштабного внедрения технологий Интернета вещей в различных областях.

1 ОБЗОР И АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Актуальность

1.1.1 Литературный обзор

Актуальность систематического обзора литературы по проблемам обнаружения и классификации вредоносного ПО в среде IoT методами машинного и глубокого обучения доказана рядом научных работ по данной теме.

Aslan, Ö., Aktuğ, S.S., Ozkan-Okay, M., Yilmaz, A.A., Akin, E. [1] в своей работе объясняют, что злоумышленники используют уязвимости, существующие на аппаратном, программном и коммуникационном уровнях. Различные типы кибератак включают распределенный отказ в обслуживании (DDoS), фишинг, "человек посередине", пароль, удаленное управление, повышение привилегий и вредоносное ПО. Из-за атак нового поколения и методов уклонения традиционные системы защиты, такие как брандмауэры, системы обнаружения вторжений, антивирусное программное обеспечение, списки контроля доступа и т.д., больше не эффективны при обнаружении этих сложных атак. Следовательно, существует настоятельная необходимость в поиске инновационных и более осуществимых решений для предотвращения кибератак. Сначала в документе подробно объясняются основные причины кибератак. Затем в нем рассматриваются самые последние атаки, схемы атак и методы обнаружения. В-третьих, в статье рассматриваются современные технические и нетехнические решения для заблаговременного распознавания атак. Использование таких трендовых технологий, как машинное обучение, глубокое обучение, облачные платформы, большие данные и блокчейн, может стать многообещающим решением для текущих и будущих кибератак.

Ali, R., Ali, A., Iqbal, F., Hussain, M., Ullah, F. [2] также пришли к выводу, что в наши дни всё сложнее становится использовать традиционные способы обнаружения кибератак. Несколько научных исследований показали, что методы глубокого обучения достигают сравнительно большей точности и могут научиться эффективно обнаруживать и классифицировать новые образцы вредоносных программ. Они провели обширный систематический обзор

литературы. Благодаря которому выявили, что CNN, LSTM, DBN и автоэнкодеры являются наиболее часто используемыми методами глубокого обучения, которые эффективно использовались в различных сценариях применения.

Bendiab, G., Shiaeles, S., Alruban, A., Kolokotronis, N. [3] в своём докладе объяснил актуальность использования подхода на базе DL. Также они выявили несколько основных причин, по которым исследователи выбирают технику ML/DL для такой задачи. На основании этого можно сделать вывод, что вопрос использования machine learning и deep learning в обнаружении и классификации вредоносного ПО является весьма актуальной задачей в конкретных случаях разработки.

Luo, X., Li, J., Wang, W., Gao, Y., Zhao, W. [4] убеждены, что есть и некоторые проблемы. Например, учитывая шум и выбросы в существующих наборах данных о вредоносных программах, некоторые методы недостаточно надежны. Следовательно, точность классификации вредоносных программ по-прежнему нуждается в улучшении. Таким образом, создание модели обнаружения и классификации вредоносного ПО в среде IoT на базе глубокого обучения — сложная и фундаментальная задача, которая должна охватывать различные способы и методы.

Подводя итог описанному выше опыту, актуальность и важность исследования становятся очевидными.

1.1.2 Связанные работы

В настоящее время мы наблюдаем динамичный рост научной активности в области кибербезопасности с использованием методов ML. В некоторых работах описывается, как разработка моделей обнаружения и классификации вредоносного ПО в среде IoT на базе ML реализуется в отдельных компаниях. Котенко И.В., Хмыров С.С. [5] рассказали о существовании уже реально использующихся подходов, таких как: Diamond, Mitre Att&ck и другие. В работе упоминаются кибердержавы (Россия, Китай и США) и их пути борьбы с

различными видами кибератак. Также сообщили о некоторых рекомендациях по созданию моделей машинного обучения и подчеркнули «растущую потребность в инновационных методах и подходах решения данной проблемы».

Khraisat, A., Alazab, A [6] пришли к выводу, что существую большое количество различных методов, но у этих методов есть свои плюсы и минусы. Исследователи рассматривают, как существующие IoT IDS обнаруживают интрузивные атаки и обеспечивают безопасность коммуникаций в IoT. В работе также представлена классификация IoT-атак и обсуждаются будущие исследовательские задачи по противодействию таким IoT-атакам, чтобы повысить безопасность Интернета вещей. Эти цели помогают исследователям безопасности Интернета вещей объединять, сопоставлять и обобщать разрозненные исследовательские усилия. Авторы считают, что ещё есть много чего можно улучшить и развить, к такому выводу они пришли на основе обширного обзора литературы.

В своей работе Гайфулина Д.А. и Котенко И.В.[7] пытаются объяснить необходимость создания новых подходов. В контексте обнаружения и классификации вредоносного ПО в среде IoT методами глубокого обучения, эти подходы могут быть применены для анализа сетевых аномалий, что является важной частью обнаружения вредоносного ПО. Глубокое обучение может быть использовано для анализа больших объемов данных, обнаружения паттернов, которые могут указывать на вредоносные действия, и автоматического обнаружения новых угроз, которые могут быть неизвестны традиционным методам обнаружения.

Примерно такого же мнения и авторы Худхейр Ауси Рим Мохаммед, Заргарян Е.В., Заргарян Ю.А.[8, 9], которые в своих работах считают, что возможности повышения безопасности различных сетей безграничны.

В статье OR-MEIR O., NISSIM N., ELOVICI Y., ROKACH L. [10] представлен всесторонний обзор методов анализа вредоносных программ с акцентом на эволюцию сложности вредоносных программ, проблемы обнаружения вредоносного программного обеспечения и роль статического и

динамического анализа в обнаружении вредоносных программ. В нем подчеркивается важность раннего обнаружения и ограничения существующих инструментов и методов перед лицом новых и эволюционирующих вредоносных угроз. В статье также освещается потенциал машинного обучения в расширении возможностей обнаружения вредоносных программ, направленный на улучшение обнаружения, классификации и категоризации. В нем критически рассматривается текущее состояние анализа вредоносных программ, отмечается необходимость постоянного совершенствования и адаптации к быстро меняющемуся ландшафту кибербезопасности. Структура статьи направлена на информирование читателей о методологиях и достижениях в области анализа вредоносных программ с целью внести вклад в более широкое обсуждение кибербезопасности и стратегий обнаружения вредоносных программ.

Статья Ахтар М.Ш., Феng Т. [11] представляет собой исследование, посвященное обнаружению ботнетов в среде Интернета вещей (IoT) с использованием гибридного подхода глубокого обучения, включающего техники CNN-LSTM. Основная цель работы заключается в разработке надежного метода для обнаружения вредоносных атак, таких как ботнеты, что критически важно для снижения рисков безопасности на устройствах IoT. Исследование использует методы машинного обучения, включая обработку естественного языка (NLP) и смешивание нейронов CNN и LSTM для захвата локальных пространственных корреляций и обучения на основе последовательных долгосрочных зависимостей. Статья подчеркивает важность симметрии данных, которая позволяет автоматизировать извлечение высокоуровневых абстракций и представлений для улучшения процесса категоризации вредоносного ПО. Результаты показывают значительное улучшение точности категоризации по сравнению с предыдущими исследованиями, достигая уровня точности категоризации выше 0.81. Предложенный метод CNN-LSTM достигает коэффициента детерминации R2 = 99.19% на предоставленном наборе данных, что делает его одним из наиболее точных методов обнаружения вредоносного ПО среди других, таких как SVM и DT. Статья представляет собой значительный вклад в область обнаружения

вредоносного ПО с использованием глубокого обучения, подчеркивая потенциал и важность использования CNN-LSTM для снижения рисков безопасности в сфере IoT.

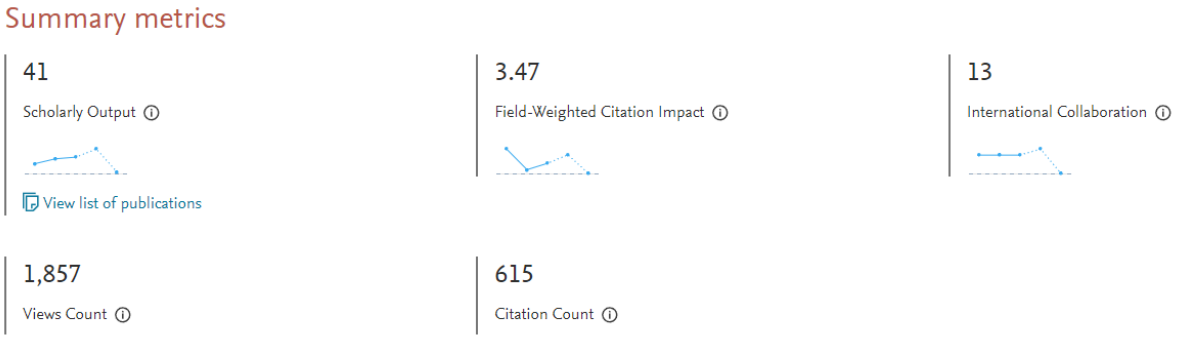
В своей статье Тоан Н.Н., Дунг Л.Т., Тхань Д.К. [12] обсуждают необходимость разработки более эффективных подходов к обнаружению вредоносного ПО на устройствах IoT с использованием моделей машинного обучения. Исследование фокусируется на использовании модели весовых коэффициентов частоты терминов в методе выбора признаков, сочетаемой с эффективной моделью машинного обучения для обнаружения вредоносного ПО на IoT устройствах на основе признаков последовательности операций. Авторы провели эксперименты на наборе данных MIPS ELF, который включает 4,511 вредоносных образцов и 4,393 безопасных программ. Результаты экспериментов показывают, что предложенный метод демонстрирует очень хорошую производительность на указанном наборе данных с точностью обнаружения и классификации 99.8% и 95.8% соответственно, при этом модели используют только 20 операций с наивысшими весовыми значениями. Эта работа представляет собой значительный вклад в область обнаружения вредоносного ПО на устройствах IoT, подчеркивая потенциал использования моделей машинного обучения для улучшения эффективности обнаружения вредоносного ПО в условиях ограниченных ресурсов.

Статья Gopali S., Siami Namin A. [13] фокусируется на обнаружении аномалий во временных рядах в контексте Интернета вещей (IoT), используя методы глубокого обучения. В частности, статья исследует производительность моделей на основе глубокого обучения, включая Bidirectional LSTM (BI-LSTM), LSTM, CNN-based Temporal Convolutional (TCN) и CuDNN-LSTM, которые представляют собой быструю реализацию LSTM, поддерживаемую CuDNN. Исследование подчеркивает важность разработки гибких и простых, но точных методов обнаружения аномалий в свете увеличения числа кибератак на сетях IoT. Результаты экспериментов показывают, что модель CuDNN-LSTM превосходит другие модели по производительности, в то время как модель на основе TCN требует меньше времени для обучения. Статья представляет собой

значительный вклад в исследование обнаружения аномалий в IoT, подчеркивая потенциал глубокого обучения для улучшения эффективности обнаружения аномалий в данных временных рядов.

1.1.3 Наукометрические показатели

Наукометрические показатели соответствующего направления исследований представлены на рисунке 1. Тенденции свидетельствуют о возрастающем интересе научного сообщества к исследованиям в этом направлении в период с 2020 по 2024.



1.2 Методология

1.2.1 Методы проведения литературного обзора

А) Критерии включения

- Оригинальные статьи и материалы конференций с описанием разрабатываемых моделей для обнаружения угроз IoT на базе глубокого обучения;

- Патентная документация, содержащая модели компонентов и подробное описание реализации обнаружения и классификации вредоносного ПО в среде IoT ;

Б) Критерий исключения

- Доклады и материалы конференций об реализации программного обеспечения без компонентов ML;

- Статьи, доступ к которым недоступен по корпоративной подписке УрФУ.

В) Вопросы исследования

- Каковы основные проблемы возникающие при внедрении DL для обнаружения вредоносного ПО?
- Какие методы глубокого обучения используются для обнаружения вредоносного ПО в среде IoT?

Потенциальный пользователь результатов: архитекторы программного обеспечения, исследователи архитектуры программного обеспечения.

Какое действие исследуется: интеграция компонентов ML в существующую или разрабатываемую архитектуру программного обеспечения по классификации и обнаружению угроз IoT-вещей.

С каким действием может быть сравнено: интеграция традиционного программного компонента в существующую или развивающуюся архитектуру.

Какой практический результат: Сокращение угроз нападения и взломов умных девайсов.

Контекст (где может быть применено): как производственная (практическая) среда, так и научная (исследовательская) среда.

Г) Поиск Процесс

Выбранные библиотеки: SciVal, Elibrary

Выбранные сроки: 2019-2024 гг.

Критерии качества: только статьи, индексированные в РИНЦ, ВАК и Scopus

Поисковый запрос :

1) **Проблемы:** (("malware") AND ("internet of things" OR "IoT") AND ("detection" OR "classification") AND ("deep learning" OR "machine learning"))
2) **Методы:** (("malware" OR "Malware Detection") AND ("internet of things" OR "IoT") AND ("detection" OR "classification") AND ("deep learning" OR "machine learning"))

Д) Процесс анализа публикаций

Шаг 0. Запрос

Шаг 1 . Проверка полнотекстовой доступности

Шаг 2 . Метаданные и анализ заголовка

Шаг 3 . Анализ аннотации

Шаг 4. Анализ результата

	SciVal	Elibrary
Шаг 0	Найдено: 41	63
Шаг 1 *	Доступно в полном тексте: 37	32
Шаг 2*	Осталось статей: 30	8
Шаг 3 *	Осталось работ : 15	5
Шаг 4 *	**Осталось документов : 7**	**5**

Е) Извлечение данных

При проведении исследования по вопросу литературного обзора №1 планируется извлечение методов классификации и обнаружения IoT-вещей на базе глубокого обучения. По вопросу №2 – извлечение проблем, а также различных подходов к решению этих проблем.

Ж) Синтез данных

Необходимо выявить спектр проблем, которые могут быть решены различными методами глубокого обучения. Необходимо создать максимально подробный список возможных моделей и выбрать наиболее эффективный метод для обнаружения и классификации угроз интернета-вещей на базе машинного обучения.

1.2.2 Методы анализа патентной документации

При анализе патентной документации используются следующие методы:

- Идентификация соответствующих патентных баз данных на основе их релевантности и доступности.
- Анализ запатентованных инструментов и методов, а также документации по зарегистрированным программам для электронных вычислительных устройств.

1.2.3 Параметры анализа патентной документации

Для достижения целей исследования необходимо извлечь следующую информацию из патентной документации:

- Описание методов обнаружения и классификации вредоносного программного обеспечения в среде IoT.
- Анализ решений и технических решений, применяемых в патентных документах.
- Выявление ключевых особенностей и тенденций развития в области безопасности IoT.

А) Поисковый запрос для патентного поиска

Для эффективного поиска патентной документации используется следующий поисковый запрос: *("threat" AND "malware" AND "internet of things" AND "IoT" AND "detection" AND "classification" AND "deep learning" AND "machine learning" AND "system" AND "algorithm" AND "security")*

Таблица 1 – Таблица анализа патентной документации

Шаг	**Действие**
Шаг 0. Запрос	Поиск патентной документации в соответствии с заданным поисковым запросом.
Шаг 1. Проверка	Проверка полнотекстовой доступности выбранных патентных документов в выбранных патентных базах данных.
Шаг 2. Анализ	Анализ содержания документов с целью выявления методов обнаружения и классификации вредоносного ПО в среде IoT.
Шаг 3. Извлечение	Извлечение данных о методах, технических решениях и особенностях, представленных в анализируемой патентной документации.

Б) Обоснование выбора патентных баз данных

Выбор конкретных патентных баз данных обусловлен их широким охватом в области кибербезопасности и доступностью релевантной информации для исследования.

В) Разъяснение критериев выбора патентных документов

При выборе патентных документов учитывается их актуальность для темы исследования, а также наличие в них информации о методах обнаружения и классификации вредоносного ПО в среде IoT.

Г) Описание патентных документов

По результатам анализа патентных документов выделены несколько технических решений, описанных в Таблице 2.

Таблица 2 – Описание патентных документов

Патент	**Дата Регистрации**	**Название**	**Описание**
WO2021087443A1	2021.05.06	INTERNET OF THINGS SECURITY ANALYTICS AND SOLUTIONS WITH DEEP LEARNING [14]	В Патенте описана реализация усиленной защиты устройств Интернета вещей (IoT) от таких угроз, как кража данных и нарушение приватности. Этот метод включает в себя мониторинг сетевого трафика IoT для идентификации событий, извлечение признаков из этих событий, обучение модели машинного обучения для их классификации и непрерывный мониторинг новых событий для обнаружения аномалий. Патент охватывает систему и методы для обнаружения аномалий в сетевом трафике IoT, используя машинное обучение для

			классификации событий на основе извлеченных признаков.
US11075934B1	2021.07.27	HYBRID NETWORK INTRUSION DETECTION SYSTEM FOR IOT ATTACKS [15]	Патент описывает систему и метод обнаружения вторжений (IDS), использующие алгоритм глубокого обучения дендритных клеток (DeepDCA), который сочетает в себе алгоритм дендритных клеток (DCA) и самонормализующуюся нейронную сеть (SNN). IDS классифицирует вторжения в сеть Интернета вещей (IoT), минимизируя ложные тревоги, и автоматизирует извлечение сигналов, улучшая производительность классификации. IDS выбирает и категоризирует признаки из набора данных IoT-Bot с использованием SNN, обеспечивая высокую точность обнаружения атак на IoT и превосходящую производительность по сравнению с классификаторами SVM, NB, KNN и MLP.
US2023328081A1	2023.10.12	SYSTEM AND METHODS FOR AUTOMATIC DETECTION OF DISTRIBUTED ATTACKS IN IOT DEVICES USING DECENTRALIZED	Патент описывает распределенные системы и методы обнаружения атак. Один из таких методов включает выполнение клиентским вычислительным устройством модели сверточной нейронной

		DEEP LEARNING [16]	сети, настроенной на обнаружение атаки в сети на клиентское вычислительное устройство; получение HTTP-запроса; извлечение универсального идентификатора ресурса (URI) из HTTP-запроса; ввод URI в модель сверточной нейронной сети; получение вывода от модели сверточной нейронной сети, классифицирующего URI как направленный на атаку в сети на клиентское вычислительное устройство; и передачу встраиваний скрытого слоя модели сверточной нейронной сети одному или нескольким серверам компьютеров, на которых размещена модель рекуррентной нейронной сети для обнаружения распределенной атаки в сети на множество клиентских вычислительных устройств.

1.3 Результаты исследования

Область обнаружения и классификации вредоносного программного обеспечения (ВПО) в среде Интернета вещей (IoT) привлекает все большее внимание исследователей в последние годы. Использование методов глубокого обучения для решения этой проблемы становится все более актуальным, учитывая сложность и масштабность среды IoT.

Множество работ посвящены анализу уязвимостей в сетях IoT и разработке методов обнаружения ВПО. Некоторые исследования фокусируются на анализе трафика в сети IoT с использованием различных методов машинного

обучения, таких как методы на основе анализа поведения, графовых моделей и алгоритмов кластеризации.

С другой стороны, существует значительное количество работ, посвященных применению глубокого обучения для решения проблемы обнаружения ВПО в среде IoT. Эти работы включают в себя использование сверточных нейронных сетей (CNN), рекуррентных нейронных сетей (RNN), а также их комбинаций для анализа данных, полученных от устройств IoT, и выявления аномалий в их поведении.

Важно отметить, что одним из основных вызовов в данной области является эффективное обнаружение ВПО при минимальном потреблении ресурсов, так как устройства IoT обычно обладают ограниченными вычислительными и энергетическими возможностями. Также существует необходимость в разработке алгоритмов, способных адаптироваться к динамическому характеру среды IoT и новым видам атак.

Работа Aslan, Ö., Aktuğ, S.S., Ozkan-Okay, M., Yilmaz, A.A., Akin, E. подчеркивает необходимость инновационных методов защиты от кибератак в условиях IoT и указывает на ограниченную эффективность традиционных систем безопасности.

Исследование Ali, R., Ali, A., Iqbal, F., Hussain, M., Ullah, F. выявляет преимущества методов глубокого обучения в обнаружении вредоносного ПО и подчеркивает значимость использования CNN, LSTM и других методов.

Работа Bendiab, G., Shiaeles, S., Alruban, A., Kolokotronis, N. аргументирует актуальность подхода на основе глубокого обучения для задач обнаружения вредоносного ПО.

Исследование Luo, X., Li, J., Wang, W., Gao, Y., Zhao, W. выделяет проблемы точности классификации вредоносных программ и необходимость улучшения методов обнаружения.

В целом, литературный обзор показывает, что глубокое обучение имеет потенциал для эффективного обнаружения и классификации ВПО в среде IoT,

однако требует дальнейших исследований и разработок для преодоления существующих вызовов и обеспечения его применимости в реальных условиях эксплуатации IoT-систем.

Исходя из проведенного обзора литературы и анализа результатов, мы делаем вывод о том, что глубокое обучение представляет собой наиболее перспективный подход для обнаружения и классификации ВПО в среде IoT. Несмотря на вызовы и ограничения, этот метод обладает высокой точностью и способностью к обнаружению сложных паттернов, что делает его наиболее подходящим для решения основной проблемы нашей работы.

1.4 Постановка задачи управления

Организационной системой, является финансовая компания, в состав которой входит отдел обеспечения информационной безопасности, эксплуатирующий систему "Обнаружения и классификация вредоносного программного обеспечения в среде IoT методами глубокого обучения".

Целью управляющего воздействия является повышение эффективности следующих параметров:

- Точность обнаружения;
- Скорость реагирования;
- Отказоустойчивость;
- Производительность системы.

Система относится к преобразованию информационного потока ресурсов. Описание процесса преобразования потока ресурсов представлено моделью черного ящика (см. Рисунок 1).

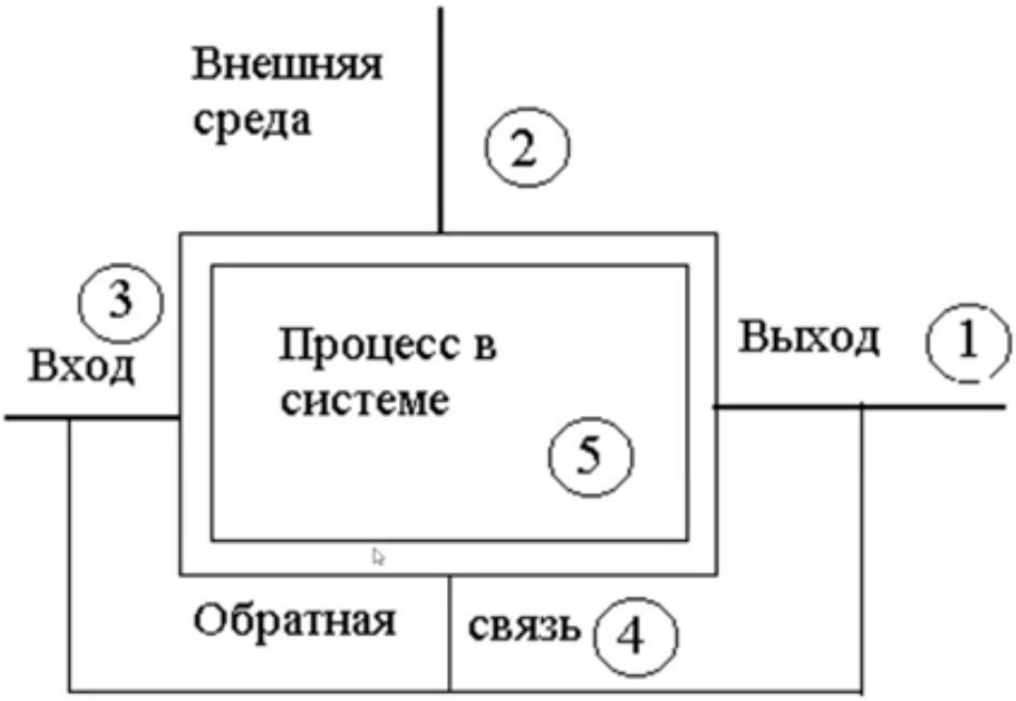


Рисунок 1 – Схема модели черного ящика

1) На выходе Система генерирует сигналы об опасности, которые могут быть использованы для принятия решений о безопасности. Это может включать в себя отчеты, оповещения или команды для обеспечения безопасности

2) Из внешней среды система получает данные об актуальных угрозах, обновления ПО.

3) На входе система получает данные о текущем состоянии подконтрольных систем

4) Из потока данных обратной связи система получает информацию о мерах реагирования, примененных к подконтрольным объектам для актуализации текущего состояния этих объектов.

5) В системе происходят процессы обработка входящих данных сетевого трафика от подконтрольных систем для генерации сигналов при обнаружении угроз подконтрольной инфраструктуре.

Надсистемой для управляемой системы является Отдел обеспечения информационной безопасности, входящий в организационную систему компании.

Так же подсистемой организационной системы, оказывающей воздействие на управляемую систему, является Отдел управления информационной инфраструктурой компании (оказывает воздействие в части управления материальными ресурсами, задействованными в работе системы).

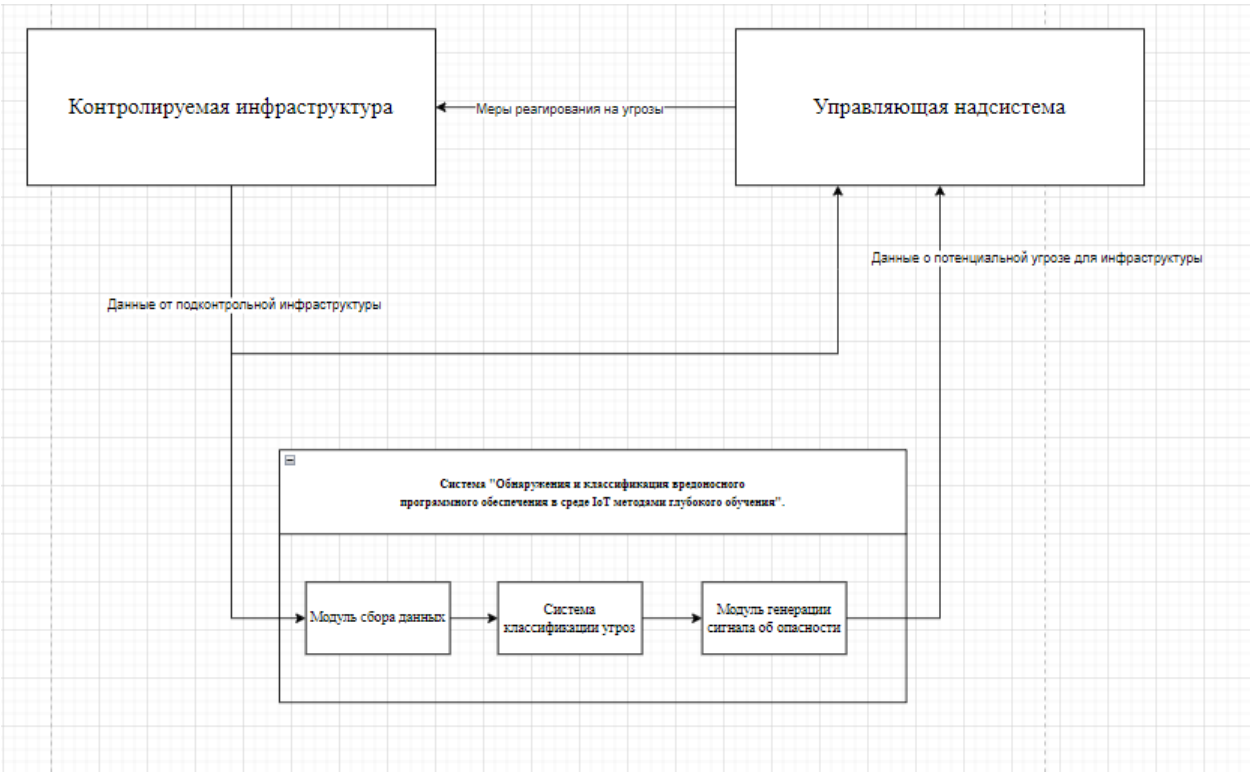
Основными функциями системы является анализ сетевого трафика, получаемого от подконтрольных IoT-устройств, включая анализ сетевых соединений и логирование действий устройств. Так же основной функцией является идентификация угроз с использованием алгоритмов машинного обучения на основе данных, полученных в процессе анализа.

Основные функции системы реализуются методами машинного обучения, которые позволяют провести категоризацию полученных данных и в случае

обнаружения угрозы для подконтрольной инфраструктуры сгенерировать сигнал об опасности.

На основе сгенерированных сигналов об опасности надсистемой принимаются меры по реагированию и устранению угроз для подконтрольной инфраструктуры.

Модель взаимодействия систем представлена на рисунке (см. Рисунок 2).



Целью функционирования системы является генерация сигнала об опасности в максимально короткий срок для обеспечения отказоустойчивости подконтрольной инфраструктуры организации.

Подробное описание подсистемы организационной системы непосредственно задействованной в управлении инновацией представлен в таблице (см. Таблица 3)

Таблица 3 – Состав управляющей подсистемы

Элемент структуры	**Функционирование**	**Влияние на показатель эффективности**
Группа реагирования на инциденты	Реагирует на сигналы об инцидентах	В целом влияет на отказоустойчивость

информационной безопасности	информационной безопасности	информационной инфраструктуры организации.
Аналитическая группа	Разработка и анализ требований для доработки системы, исследование новых технологий и методов обнаружения	Влияет на точность обнаружения, скорость реагирования.
Группа разработки	Разработка алгоритмов машинного обучения, интеграция с другими системами, создание архитектуры системы.	Влияет на все показатели эффективности через оптимизацию алгоритмов, архитектуры системы и интеграцию с другими системами.
Группа тестирования	Тестирование системы	Влияет на все показатели эффективности через отказоустойчивость системы.
Группа внедрения	Поддержка и обновление системы, обработка обратной связи.	Влияет на уровень поддержки и обновления через быстрое внедрение обновлений и улучшений

Жизненный цикл системы определен следующими этапами:

1) Замысел.
2) Разработка.
3) Производство
4) Применение
5) Поддержка.

В рамках данной работы этап прекращения применения и вывода из эксплуатации системы не рассматривается, т.к. предполагается постоянное использование системы до момента прекращения деятельности организационной системы.

Определим целевую функцию системы и множество ограничений, вызывающих нарушения работоспособности системы

$Q(x)$ – целевая функция (скорость устранения инцидентов информационной безопасности)

где x – вектор переменных (Точность обнаружения (A), скорость реагирования(S), отказоустойчивость системы обнаружения(F), производительность системы(P)), а

$D = \{D_{ag}, D_{dev}, D_{tes}, D_{dep}, D_{infr}\}$ – множество ограничений/нарушений.

D_{ag} – Ограничение в получении данных о новых угрозах и нарушения в разработке требований для доработки системы.

D_{dev} – Нарушение при разработке и доработке алгоритмов работы системы.

D_{tes} – Нарушения при тестировании новых алгоритмов и изменений системы, которые могут привести к полному отказу.

D_{dep} – Нарушения при обновлении системы, ограничения в ресурсах для масштабирования системы.

D_{infr} – Ограничения в ресурсах инфраструктуры для внесения изменений.

CF – отказоустойчивость инфраструктуры организации

$r(Q(x), CF)$ – корреляция Пирсона между скоростью устранения инцидентов информационной безопасности и отказоустойчивостью инфраструктуры организации, что влияет на эффективность работы организации в целом.

Цель состоит в том, чтобы:

$$Q(x) \text{ при } D \leq 0$$

где $x \in X = \{A, S, F, P\}$

Так же на скорость устранения инцидентов информационной безопасности влияет относительная эффективность подразделения информационной безопасности, где эффективность каждой единицы описывается следующим образом.

$$Eff_i = max \frac{y_i' \gamma}{x_i' \beta} = \frac{\sum_{s=1}^{S} y_{si} \gamma_s}{\sum_{m=1}^{M} x_{mi} \beta_m}$$

Где:

y_i – Выходной вектор блока i

x_i – Входной вектор блока i

γ – Вектор выходных весов/коэффициентов

β – Вектор входных весов/коэффициентов

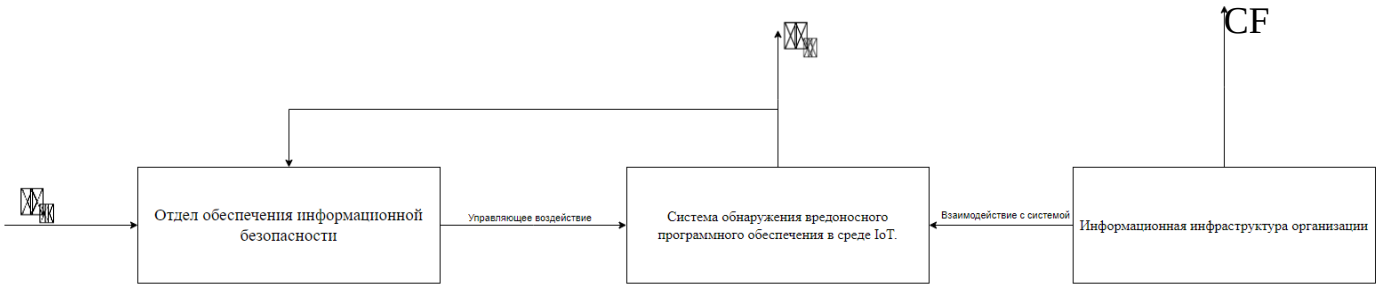
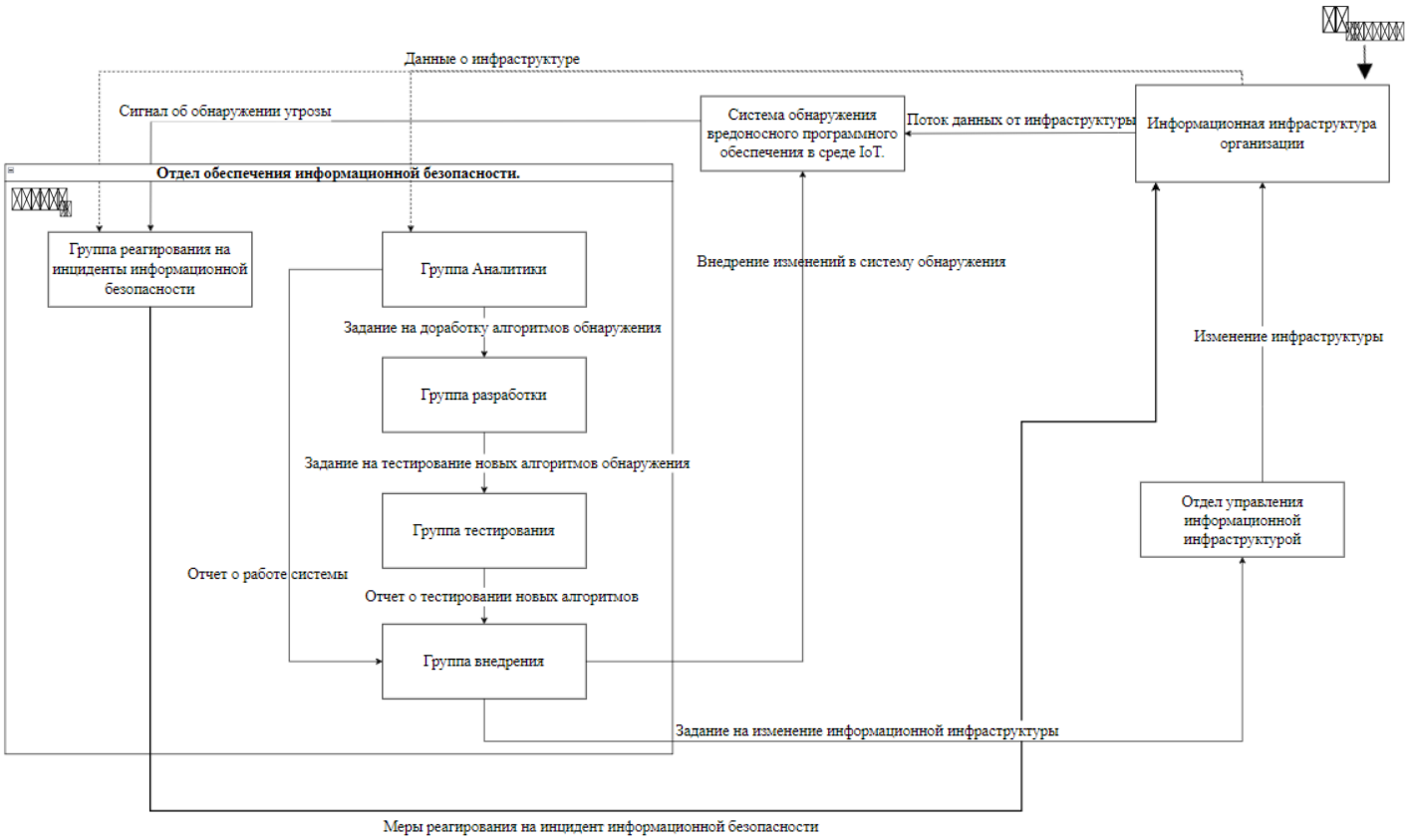


Рисунок 3 – Общая схема взаимодействия



1.5 Подведение итогов исследования

Исходя из проведенного исследования методик и инструментов обнаружения вторжений в сетях Интернета вещей сделаны следующие выводы:

1) Выполнение поставленных задач:

Изучены существующие методы обнаружения и классификации вредоносного ПО.

Проанализированы особенности среды IoT и ее уязвимости перед атаками вредоносного ПО.

Исследованы возможности применения методов глубокого обучения для обнаружения и классификации вредоносного ПО в среде IoT.

Разработаны модели глубокого обучения для обнаружения и классификации вредоносного программного обеспечения.

Проведено экспериментальное исследование разработанных моделей на реальных данных, оценена их эффективность и точность.

Предложены рекомендации по улучшению методов обнаружения и классификации вредоносного программного обеспечения в среде IoT.

2) Степень изученности объекта исследования:

Объектом исследования было вредоносное программное обеспечение, а предметом - методы интеграции и эксплуатации систем обнаружения и классификации вредоносного ПО с использованием машинного обучения в сетях Интернета вещей.

Результаты показали, что использование специализированных моделей глубокого обучения является перспективным подходом к решению проблем безопасности IoT.

3) Научная новизна и практическая значимость:

Научная новизна работы заключается в разработке специализированных моделей глубокого обучения для обнаружения и классификации вредоносного программного обеспечения в среде IoT.

Результаты данного исследования могут быть полезны для разработчиков систем безопасности IoT, способствуя созданию более надежных средств обнаружения и предотвращения атак в этой области.

4) Обоснование актуальности области исследования:

Актуальность проблемы обнаружения и классификации вредоносного ПО в среде IoT методами глубокого обучения подтверждается рядом научных исследований. Современные киберугрозы, такие как DDoS-атаки, фишинг и вредоносное ПО, требуют инновационных подходов к безопасности.

Использование технологий глубокого обучения, таких как нейронные сети и алгоритмы машинного обучения, представляет собой перспективное решение для эффективного обнаружения и классификации вредоносного программного обеспечения в среде IoT.

В целом, проведенное исследование подтвердило актуальность использования методов глубокого обучения для повышения безопасности умных устройств в среде IoT. Полученные результаты открывают новые перспективы для развития систем защиты от киберугроз, особенно в контексте быстроразвивающихся технологий интернета вещей.

2 ОБЗОР СУЩЕСТВУЮЩИХ ИНСТРУМЕНТОВ ОБНАРУЖЕНИЯ И КЛАССИФИКАЦИИ ВРЕДОНОСНОГО ПО В СРЕДЕ IOT

2.1 Традиционные системы обнаружения вредоносного ПО

2.1.1 Системы обнаружения вторжений (IDS) на основе сигнатур

Сигнатурные IDS идентифицируют угрозы путем сравнения сетевого трафика или поведения устройства с базой известных сигнатур вредоносного ПО. Примеры таких систем:

- Snort: Популярная открытая IDS, способная анализировать трафик в реальном времени и сравнивать его с базой сигнатур.
- Suricata: Высокопроизводительная IDS, которая также поддерживает работу с сигнатурами и обеспечивает расширенные функции по анализу сетевого трафика.

Преимущества:

- Высокая точность обнаружения известных угроз.
- Низкое количество ложных срабатываний.

Недостатки:

- Неэффективны против новых, неизвестных угроз (нулевого дня).
- Требуют регулярного обновления базы сигнатур.

2.1.2 Системы обнаружения на основе аномалий

Эти системы выявляют угрозы, обнаруживая отклонения от нормального поведения устройств или сетевого трафика. Примеры:

- Kismet: Беспроводная IDS, которая обнаруживает аномалии в сетевом трафике Wi-Fi.
- Zebra: IDS, использующая методы машинного обучения для анализа сетевого трафика и выявления аномалий.

Преимущества:

- Возможность обнаружения новых и неизвестных угроз.
- Способность адаптироваться к изменяющимся условиям сети.

Недостатки:

- Высокий уровень ложных срабатываний.
- Необходимость обучения и настройки моделей.

2.1.3 Гибридные системы

Гибридные IDS сочетают в себе методы обнаружения на основе сигнатур и аномалий для повышения общей эффективности. Примеры:

- AI-based Intrusion Detection Systems: Системы, использующие комбинированный подход, включая машинное обучение и базы сигнатур для улучшения точности и снижения количества ложных срабатываний.
- Zeek (Bro): Мощный сетевой анализатор, который может работать как сигнатурная IDS и система обнаружения на основе аномалий.

Преимущества:

- Высокая точность обнаружения при снижении количества ложных срабатываний.
- Универсальность и способность адаптироваться к различным типам угроз.

Недостатки:

- Высокие затраты ресурсов
- Высокая сложность эксплуатации в части обучения персонала процессам модернизации и управления системой

2.2 Ограниченность традиционных систем обнаружения вторжений

Традиционные IDS, основанные на сигнатурах, зависят от предварительно известных шаблонов атак. Эти системы:

- Эффективны против известных угроз: Они сравнивают сетевой трафик и действия устройств с базой данных сигнатур, что позволяет быстро идентифицировать известные виды вредоносного ПО и атак.
- Неэффективны против угроз нулевого дня: Новые и неизвестные угрозы, для которых нет соответствующих сигнатур в базе данных, остаются незамеченными. Это особенно критично в динамично развивающейся среде IoT, где ежедневно появляются новые уязвимости и эксплойты.

Сигнатурные IDS требуют постоянного обновления для эффективного функционирования:

- Регулярные обновления базы сигнатур: Необходимость частого обновления баз данных сигнатур для отражения новых угроз приводит к значительным затратам на поддержание актуальности системы.
- Зависимость от разработчиков: Задержки в обновлении сигнатур, зависящие от разработчиков систем безопасности, могут привести к временным окнам уязвимости.

IoT-сети характеризуются большим числом разнообразных устройств с различными функциональными и производительными характеристиками:

- Ресурсные ограничения IoT-устройств: Многие IoT-устройства обладают ограниченными вычислительными ресурсами, памяти и энергии, что затрудняет интеграцию традиционных IDS, требующих значительных ресурсов для анализа трафика и выполнения сигнатурного сканирования.
- Масштабируемость и производительность: В больших сетях IoT объем данных, подлежащих анализу, значительно возрастает, что может

привести к перегрузке традиционных IDS и снижению их производительности.

Традиционные IDS часто обладают ограниченной способностью к адаптации в условиях быстро меняющихся сетевых конфигураций и новых типов устройств:

- Сложность в адаптации к новым протоколам и стандартам: IoT-сети используют широкий спектр протоколов и стандартов связи, что требует от IDS гибкости и возможности адаптации к новым условиям.
- Статический подход: Сигнатурные IDS не способны самостоятельно адаптироваться к изменениям в поведении сети или устройств, что снижает их эффективность в условиях динамичных сетей IoT.

Традиционные IDS, основанные на сигнатурах, могут генерировать значительное количество ложных срабатываний:

- Ложные срабатывания: Высокий уровень ложных срабатываний может перегружать систему оповещений и снижать доверие к IDS.
- Требование к ручной проверке: Ложные срабатывания требуют значительных усилий по ручной проверке и анализу, что увеличивает рабочую нагрузку на специалистов по безопасности.

2.3 Системы обнаружения вторжений на основе машинного обучения

Системы на основе машинного обучения используют алгоритмы, которые обучаются на исторических данных и способны делать прогнозы или выявлять аномалии на основе анализа текущего сетевого трафика и поведения устройств. Принципы работы таких систем включают:

1) **Сбор данных:** Мониторинг и сбор данных о сетевом трафике, активности устройств и других релевантных параметров.
2) **Предварительная обработка данных:** Очистка, нормализация и трансформация данных для подготовки к анализу.

3) **Обучение моделей:** Использование обучающих данных для создания модели, способной различать нормальное и аномальное поведение.

4) **Анализ и обнаружение:** Применение обученной модели для анализа реального трафика и выявления подозрительных действий или аномалий.

2.3.1 Алгоритмы контролируемого обучения

Эти алгоритмы обучаются на размеченных данных, где известны метки нормального и аномального поведения:

- **Решающие деревья (Decision Trees)**: Древовидные структуры, которые разделяют данные на основании атрибутов.
- **Поддерживающие векторы (Support Vector Machines, SVM)**: Алгоритмы, которые находят гиперплоскость, разделяющую данные на классы.
- **Линейная и логистическая регрессия**: Алгоритмы, используемые для прогнозирования бинарных и многоклассовых исходов.

Преимущества:

- Высокая точность классификации: При правильной настройке и обучении могут достигать высокой точности предсказания.
- Интерпретируемость: Часто легко интерпретируемы и понятны специалистам в предметной области.
- Эффективность на небольших данных: Хорошо справляются с задачами на небольших выборках данных.

Недостатки:

- Требовательность к размеченным данным: Требуют большого объема размеченных данных для обучения.

- Склонность к переобучению: Могут страдать от переобучения на шумовых или несбалансированных данных.

- Неэффективность при нелинейных зависимостях: Могут быть менее эффективны при сложных нелинейных зависимостях в данных.

2.3.2 Алгоритмы неконтролируемого обучения

Эти алгоритмы не требуют размеченных данных и используют методы кластеризации и выявления аномалий:

- **Кластеризация k-средних (k-means)**: Метод, группирующий данные в k кластеров на основе их сходства.

- **Метод главных компонент (PCA)**: Алгоритм для уменьшения размерности данных и выявления аномалий.

- **Автоэнкодеры**: Нейронные сети, обучающиеся реконструировать входные данные и выявлять аномалии на основе ошибки реконструкции.

Преимущества:

- Без необходимости разметки данных: Могут быть эффективно использованы при отсутствии размеченных данных.

- Обнаружение скрытых шаблонов: Способны выявлять скрытые паттерны или кластеры в данных.

- Применимость к большим объемам данных: Хорошо масштабируются на большие наборы данных.

Недостатки:

- Трудность интерпретации: Результаты обучения могут быть менее понятны для аналитиков.

- Неоднозначность в выборе гиперпараметров: Некоторые алгоритмы требуют настройки гиперпараметров, которая может быть неоднозначной.

- Чувствительность к начальным условиям: Некоторые алгоритмы могут давать различные результаты в зависимости от начального приближения.

2.4Особенности среды IoT и её уязвимости перед вредоносным ПО

Интеграция систем обнаружения вторжений (IDS) в сети предприятия требует тщательного планирования и выполнения для обеспечения эффективной защиты от киберугроз. Современные IDS, особенно те, которые основаны на методах машинного обучения, могут значительно улучшить безопасность сети, но их успешное внедрение зависит от нескольких ключевых аспектов. Рассмотрим основные методики интеграции IDS в сети предприятия.

Определение требований и целей

Первым шагом является определение требований и целей интеграции IDS:

Анализ рисков и угроз: Идентификация наиболее значимых угроз и уязвимостей в сети предприятия.

Выбор подходящего типа IDS: Определение, будет ли использоваться сетьевая (NIDS), хостовая (HIDS) или гибридная IDS.

Определение критических активов: Идентификация наиболее важных ресурсов и сегментов сети, требующих защиты.

2. Выбор системы IDS

Выбор подходящей IDS-системы зависит от нескольких факторов:

Производительность и масштабируемость: Оценка способности IDS обрабатывать объемы сетевого трафика в реальном времени.

Совместимость с существующей инфраструктурой: Проверка совместимости IDS с текущими сетевыми устройствами и программным обеспечением.

Функциональность и возможности: Оценка функциональности IDS, включая возможности машинного обучения, анализа поведения, отчетности и интеграции с другими системами безопасности.

3. Планирование архитектуры IDS

Архитектура IDS должна быть тщательно спланирована для обеспечения максимальной эффективности:

Распределение сенсоров: Размещение сенсоров IDS в ключевых точках сети, таких как периметры, внутренние сегменты и вокруг критически важных систем.

Централизованное управление: Настройка централизованной системы управления для мониторинга и управления всеми сенсорами IDS.

Сегментация сети: Использование сетевой сегментации для изоляции критически важных ресурсов и улучшения эффективности обнаружения вторжений.

Процесс установки и настройки включает:

Развертывание сенсоров: Установка сенсоров в заранее определенных местах сети.

Конфигурация системы: Настройка параметров IDS, таких как правила обнаружения, пороговые значения для триггеров, и методы оповещения.

Интеграция с SIEM: Интеграция IDS с системами управления информацией и событиями

2.5 Особенности среды IoT и её уязвимости перед вредоносным ПО

С развитием технологий и увеличением количества подключенных устройств, проблема безопасности в среде Интернета вещей (IoT) становится все более актуальной. IoT-системы обеспечивают беспрецедентные возможности для автоматизации и улучшения различных аспектов жизни, но одновременно открывают новые векторы для кибератак. Учитывая разнообразие и масштаб IoT,

вопросы безопасности требуют особого внимания и комплексного подхода. В этом разделе рассматриваются основные аспекты безопасности IoT, включая архитектуру IoT и её компоненты, типичные угрозы и атаки, а также ограничения IoT-устройств и их влияние на безопасность.

2.5.1 Архитектура IoT и её компоненты

Интернет вещей (IoT) представляет собой сложную и многослойную сеть, состоящую из множества различных компонентов, которые взаимодействуют друг с другом для сбора, передачи и анализа данных. Основная цель IoT — создание интеллектуальных систем, способных улучшить повседневную жизнь и повысить эффективность различных процессов в промышленности, здравоохранении, сельском хозяйстве и других сферах. Архитектура IoT включает в себя следующие ключевые компоненты:

- **Устройства и сенсоры**: Это основа IoT, представляющая собой физические устройства, оснащенные сенсорами для сбора данных из окружающей среды. Примеры таких устройств включают умные термостаты, фитнес-трекеры, промышленные датчики и многие другие.
- **Gateway или шлюз**: Гейтвей служит посредником между IoT-устройствами и облаком. Он собирает данные с устройств, обрабатывает их и передает в облачные системы для дальнейшего анализа. Гейтвей также может выполнять функции фильтрации данных и обеспечения безопасности.
- **Сетевые протоколы**: Для передачи данных между устройствами и облаком используются различные сетевые протоколы, такие как Wi-Fi, Bluetooth, Zigbee и другие. Эти протоколы обеспечивают надежное и эффективное соединение между компонентами IoT-системы.
- **Облачные платформы**: Облако предоставляет инфраструктуру для хранения и обработки больших объемов данных, поступающих от IoT-

устройств. Облачные платформы также включают инструменты для аналитики, управления устройствами и разработки приложений.

- **Приложения и интерфейсы**: Это пользовательские интерфейсы и приложения, которые позволяют взаимодействовать с IoT-устройствами, анализировать данные и принимать на их основе решения. Примеры включают мобильные приложения для управления умными домами или промышленные системы мониторинга.

2.5.2 Типичные угрозы и атаки в среде IoT

С ростом популярности и распространения IoT-устройств увеличивается и количество угроз, связанных с их использованием. В IoT-среде существует множество потенциальных уязвимостей, которые могут быть использованы злоумышленниками для проведения атак. К типичным угрозам и атакам в среде IoT относятся:

- **Атаки на конфиденциальность данных**: Перехват и кража данных, передаваемых между IoT-устройствами и облачными сервисами, могут привести к утечке конфиденциальной информации, такой как персональные данные пользователей или коммерческая тайна.

- **DDoS-атаки**: Скомпрометированные IoT-устройства могут быть использованы для проведения распределённых атак отказа в обслуживании (DDoS), что может привести к недоступности критически важных сервисов и нарушению работы инфраструктуры.

- **Захват устройств**: Несанкционированный доступ к IoT-устройствам позволяет злоумышленникам управлять ими, изменять их настройки или использовать их для проведения дальнейших атак.

- **Инъекции вредоносного ПО**: Внедрение вредоносного программного обеспечения в IoT-устройства может привести к сбору данных или нарушению их работы, что особенно опасно для промышленных и медицинских систем.

2.5.3 Атаки на аппаратное обеспечение

Аппаратное обеспечение IoT-устройств также подвержено различным видам атак, которые могут нанести значительный ущерб системе. К таким атакам относятся:

- **Физический доступ**: Злоумышленники могут получить физический доступ к устройству для его модификации, кражи данных или установки вредоносных компонентов. Например, вандалы могут повредить сенсоры или камеры видеонаблюдения.
- **Сторонние каналы**: Использование электромагнитного излучения или других побочных эффектов для извлечения конфиденциальной информации. Такие атаки могут быть особенно опасны для защищённых объектов и военных систем.
- **Подмена компонентов**: Злоумышленники могут заменить оригинальные компоненты устройства на модифицированные, содержащие вредоносные функции. Это может привести к потере контроля над устройством и компрометации всей системы.

2.5.4 Атаки на программное обеспечение

Программное обеспечение IoT-устройств является одной из самых уязвимых частей системы, так как ошибки и уязвимости в коде могут быть использованы для проведения атак. К основным типам атак на программное обеспечение относятся:

- **Эксплуатация уязвимостей**: Злоумышленники могут использовать известные уязвимости в программном обеспечении для получения несанкционированного доступа к устройствам и данным. Это может

включать эксплуатацию недостатков в операционной системе или приложениях.

- **Инъекции кода**: Внедрение вредоносного кода в программное обеспечение устройства позволяет злоумышленникам выполнять несанкционированные действия, такие как кража данных, изменение настроек или отключение устройства.

- **Руткиты**: Установка руткитов для скрытия присутствия вредоносного ПО и обеспечения долгосрочного контроля над устройством. Руткиты могут быть очень сложными для обнаружения и удаления, что делает их особенно опасными.

2.5.5 Атаки на коммуникационные каналы

Коммуникационные каналы, через которые передаются данные между IoT-устройствами и облачными сервисами, также уязвимы к различным атакам. Эти атаки могут серьезно подорвать безопасность и надежность IoT-систем. Основные виды атак включают:

- **Перехват данных (Eavesdropping)**: Злоумышленники могут перехватывать данные, передаваемые по сети, что позволяет им получать конфиденциальную информацию, такую как пароли или личные данные пользователей.

- **Man-in-the-Middle атаки (MitM)**: При такой атаке злоумышленник вставляется между устройством и сервером, перехватывая и изменяя передаваемые данные. Это позволяет ему получить контроль над передаваемой информацией и даже изменить ее содержание.

- **Подмена сообщений (Replay attacks)**: Повторное использование перехваченных сообщений для выполнения несанкционированных действий. Это может привести к выполнению несанкционированных команд или предоставлению доступа к защищённым ресурсам.

2.5.6 Ограничения IoT-устройств и их влияние на безопасность

Одной из ключевых проблем обеспечения безопасности в IoT-среде являются ограничения самих устройств, которые влияют на их способность противостоять кибератакам. Эти ограничения включают:

- **Ограниченные вычислительные ресурсы**: Большинство IoT-устройств обладают ограниченными вычислительными возможностями, такими как малое количество оперативной памяти и процессорной мощности. Это ограничивает использование сложных алгоритмов шифрования и других методов защиты.
- **Низкое энергопотребление**: IoT-устройства часто разработаны для работы от батареи и имеют ограниченные энергетические ресурсы. Это требует минимизации энергопотребления, что в свою очередь ограничивает возможности реализации энергоемких защитных механизмов.
- **Низкая стоимость производства**: Стремление к снижению стоимости производства IoT-устройств приводит к сокращению затрат на обеспечение безопасности. Это может проявляться в использовании дешёвых и менее защищённых компонентов или недостаточном тестировании на уязвимости.

Эти ограничения создают дополнительные вызовы для разработчиков и пользователей IoT-систем, требуя разработки инновационных решений для обеспечения надёжной защиты данных и устройств.

2.6 Интеграции предлагаемого инструмента

В рамках данной работы предлагается интеграция комплексного инструмента обнаружения вторжений в сетях Интернета вещей промышленного предприятия с предварительным тестированием на испытательном стенде.

2.6.1 Топологии сетевой инфраструктуры предприятий

Типовые топологии сетевой инфраструктуры предприятий включают в себя шинную, кольцевую, звездообразную, древовидную и смешанную топологии.

Шинная топология предполагает, что все устройства подключаются к единой линии передачи, что делает ее простой и недорогой, но менее надежной и масштабируемой.

Кольцевая топология предполагает последовательное подключение устройств, образующих замкнутое кольцо, что делает ее более отказоустойчивой, но сложнее в настройке.

Звездообразная топология подразумевает подключение всех устройств к центральному коммутационному узлу, что обеспечивает простоту изоляции и диагностики неисправностей, и широко используется в современных локальных сетях предприятий.

Древовидная топология представляет собой иерархическую структуру, где коммутаторы и маршрутизаторы образуют древовидную схему, что обеспечивает масштабируемость и структурированность сети.

Смешанная топология представляет собой комбинацию нескольких базовых топологий, что позволяет сочетать преимущества разных схем организации сети и широко используется в корпоративных сетях средних и крупных предприятий.

Звездообразная топология чаще всего используется на предприятиях. Она подразумевает подключение всех устройств к центральному коммутационному узлу, что обеспечивает простоту изоляции и диагностики неисправностей, и широко применяется в современных локальных сетях предприятий (см. Рисунок 5)

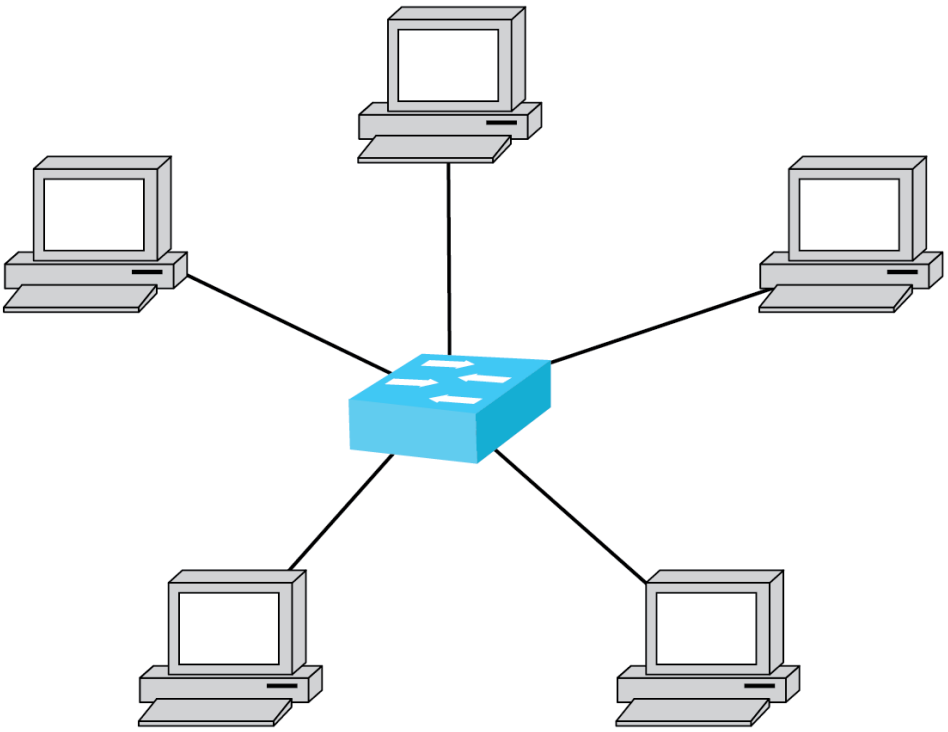


Рисунок 5 – Типовая схема топологии Звезда

2.6.2 Средства для сбора данных о сетевом трафике

Для сбора и анализа сетевого трафика на предприятиях используются различные инструменты. Это могут быть сетевые анализаторы (снифферы), такие как Wireshark, tcpdump или Netflow, которые позволяют перехватывать и детально изучать пакеты данных в сети.

Также применяются системы мониторинга и управления сетью, например, Zabbix, Nagios или SolarWinds, которые собирают статистику о работе сетевого оборудования, производительности и доступности различных сетевых сервисов. Для обнаружения вторжений и анализа угроз безопасности используются

системы предотвращения вторжений (IPS) и системы обнаружения вторжений (IDS), такие как Snort или Suricata.

Кроме того, предприятия также используют решения для централизованного сбора и хранения сетевых журналов (логов), например Elasticsearch, Logstash и Kibana. Все эти инструменты помогают ИТ-специалистам отслеживать состояние сети, выявлять проблемы и обеспечивать ее безопасность.

В рамках данной работы в качестве средства сбора данных о сетевом трафике будет использован Wireshark, т.к. данный инструмент позволяет осуществлять захват, анализ и декодирование пакетов данных практически любых сетевых протоколов. Данное ПО так же позволяет провести глубокий анализ сетевого трафика в случае обнаружения внедряемым инструментом угроз сетевой инфраструктуре (см. Рисунок 6).

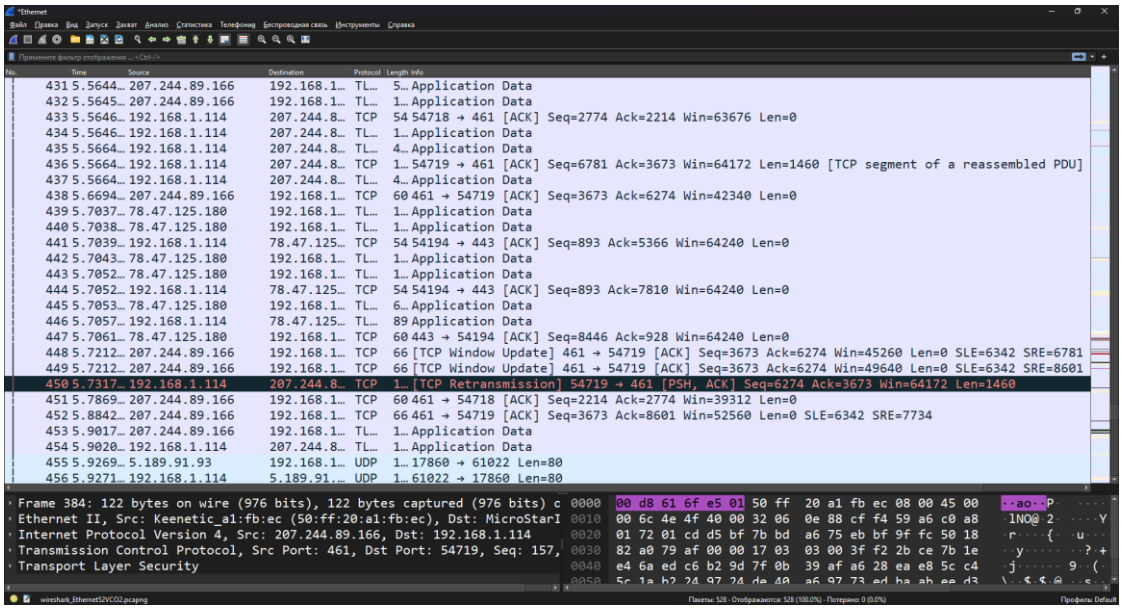


Рисунок 6 – Типовое окно ПО Wireshark

2.6.3 ПО для подготовки данных Zeek

ZEEK (ранее Bro) - это мощная платформа сетевого мониторинга и анализа, фокусирующаяся на детальном исследовании сетевой активности и выявлении угроз безопасности; она способна глубоко декодировать и

анализировать широкий спектр протоколов, использовать алгоритмы машинного обучения для обнаружения аномалий и подозрительной активности, работать с высокими скоростями сетевого трафика, предоставлять гибкий скриптовый язык для расширения функциональности, а также интегрироваться с другими системами безопасности, выступая в качестве продвинутого инструмента для комплексного обеспечения кибербезопасности.

В рамках данной работы ПО Zeek выбрано в качестве средства подготовки данных для анализа внедряемым инструментом в силу своей высокой производительности и полноты предоставляемых данных, необходимых внедряемому инструменту.

2.6.4 Выбор используемой модели

В рамках данной работы протестированы варианты внедрения инструментов на основе двух обученных моделей:

1) Сочетающая в себе PCA, ансамблевые методы и случайны лес (Rotation Forest)
2) Градиентный бустинг (XGBoost)

Данные модели выбраны в качестве основных для внедрения в инфраструктуру в силу своих высоких показателей точности и скорости обучения на больших наборах данных, который для каждой модели составили

1) 0.9922777019840934 - Rotation Forest
2) 0.994581403939118 – XGBoost

2.6.5 Краткое описание процесса работы

Система сбора данных о сетевом трафике использует зеркалирование портов на маршрутизаторах, чтобы дублировать весь трафик на рабочую станцию оператора информационной безопасности с установленным Wireshark, где формируются файлы для дальнейшего анализа. Эти данные передаются в

Zeek, где сформированный Wireshark файл обрабатывается и преобразовывается в готовый для анализа интегрированным инструментом формат. По результатам анализа выводятся результаты обнаруженных угроз, если таковые были.

2.6.6 Описание процесса тестирования моделей

Предложенные модели перед непосредственно внедрении в инфраструктуру предлагается протестировать на стенде, принципиально дублирующем сетевую инфраструктуру предприятия в части организации сетевого взаимодействия оборудования.

Предлагается симулировать получение данных с пограничного маршрутизатора промышленной сети, подготовить их к обработке моделью с помощью Wireshark и Zeek и оценить параметры работы внедряемой модели

В качестве наборов данных для тестирования предполагается использование наборов данных, полученных путем симуляции ICMP Flood DoS-атаки, Slow comprehensive scan, наборы данных использованного для обучения модели (в т.ч. различных его частей)

Для симуляции атаки на сетевую инфраструктуру будет использован стенд, непосредственно внедренный в инфраструктуру, позволяющей одновременно воспроизвести описанные выше атакующие процессы, в том числе привести к отказу атакуемого оборудования.

2.6.7 Оцениваемые параметры

В рамках тестирования предлагается оценить следующие параметры моделей:

1) Минимальная требуемая конфигурация оборудования для работы внедряемой модели
2) Общая работоспособность при различных объемах данных

3) Скорость обработки различных объемов данных при использовании минимальной конфигурации оборудования
4) Сравнение точности моделей, полученной в процессе обучения с результатами тестирования на оборудовании\

2.6.8 Ожидаемые результаты тестирования

В качестве результатов тестирования предполагается установить минимальную необходимую конфигурацию оборудования и ПО, которая в беспрерывном режиме позволит обеспечить сбор, хранение, подготовку данных к обработке, непосредственную обработку и формирование результатов работы модели.

3 ВНЕДРЕНИЕ ПРЕДЛОЖЕННОГО ПОДХОДА НА ПРЕДПРИЯТИИ

3.1 Введение

В настоящее время сфера Интернета вещей (IoT) становится все более значимой и широко применяется в различных отраслях, включая промышленность, здравоохранение, транспорт и многое другое. Однако, с увеличением числа подключенных устройств и расширением функциональности IoT систем, возрастает и риск возникновения вредоносных атак и нарушений безопасности.

Целью данной главы является рассмотрение процесса внедрения предложенного подхода по обнаружению и классификации вредоносного программного обеспечения (Malware) в среде IoT с использованием методов глубокого обучения. Основной задачей является описание этапов интеграции разработанного решения в рабочие бизнес-процессы организации и оценка его эффективности на реальных данных.

Данная глава структурирована следующим образом. В начале будет проведен анализ текущих бизнес-процессов в контексте применения системы IoT, выявлены проблемы и потребности, а также определены основные

категории пользователей. Затем будет представлен процесс интеграции предложенного подхода, включая подготовку к внедрению, этапы внедрения и мониторинга. После этого будет осуществлена экспериментальная оценка эффективности предложенного подхода на реальных данных и проведено сравнение его с альтернативными методами. В конце главы будет представлен анализ результатов, выводы и рекомендации для практического применения.

Внедрение разработанного подхода позволит повысить уровень безопасности в среде IoT, защитить конфиденциальные данные и предотвратить возможные угрозы в виде вредоносного программного обеспечения. Это, в свою очередь, способствует улучшению надежности и эффективности бизнеса, а также укреплению репутации организации.

3.2 Бизнес-процессы и пользователи

Сфера Интернета вещей (IoT) предоставляет бесконечные возможности для автоматизации и оптимизации различных бизнес-процессов в различных отраслях, начиная от промышленности и заканчивая здравоохранением. В контексте данного исследования, необходимо провести анализ существующих бизнес-процессов, где предполагается использование разработанной системы для обнаружения и классификации вредоносного программного обеспечения в среде IoT.

В данном разделе рассматривается анализ бизнес-процессов, где предполагается использование разработанной системы обнаружения и классификации вредоносного программного обеспечения в среде Интернета вещей (IoT), а также идентификация категорий пользователей, их ролей и потребностей в контексте данного исследования.

3.2.1 Анализ текущих бизнес-процессов

В рамках анализа текущих бизнес-процессов было выявлено несколько ключевых областей, где применение системы IoT является критически важным. Среди этих областей выделяются:

- **Промышленность**: Мониторинг состояния и производственных процессов на заводах, обнаружение нештатных ситуаций и аварий.
- **Здравоохранение**: Мониторинг состояния пациентов в реальном времени, оптимизация лечебных процессов и управление медицинским оборудованием.
- **Транспорт**: Оптимизация логистических процессов, мониторинг транспортных средств и обеспечение безопасности на дорогах.

3.2.2 Идентификация пользователей

После анализа бизнес-процессов были определены следующие категории пользователей системы:

- **Операторы производственных линий:** отвечают за мониторинг и управление производственным оборудованием.
- **Инженеры по обслуживанию оборудования:** занимаются техническим обслуживанием и ремонтом устройств IoT.
- **Операторы информационной безопасности**: отвечают за настройку и мониторинг системы обнаружения вредоносного программного обеспечения.
- **Аналитики данных**: занимаются анализом данных, полученных от устройств IoT, для выявления аномалий и принятия решений на основе полученной информации.

Каждая категория пользователей имеет свои уникальные роли и обязанности, а также различные потребности и ожидаемые выгоды от

использования системы обнаружения и классификации вредоносного программного обеспечения в среде IoT.

3.3 Тестирование модели

Перед непосредственным внедрением модели в сетевую инфраструктуру предприятия необходимо провести тестирование на различных конфигурациях оборудования и входных данных для оценки производительности и общей работоспособности модели.

3.3.1 Подготовка оборудования

В качестве рабочей станции оператора информационной безопасности, на которой буде осуществляется непосредственная работа модели, предлагается ПЭВМ в следующих конфигурациях (см. Таблица 2)

Таблица 4

Параметр рабочей станции	**Конфигурация 1**	**Конфигурация 2**	**Конфигурация 3**
Операционная система	Windows 11 x64	Linux Ubuntu 22.04 x64	Linux Ubuntu 22.04 x64
Центральный процессор	16-ядерный процессор 3.6 GHz	4-ядерный процессор 3.6 GHz	8-ядерный процессор 3.6 GHz
Оперативная память	32 GB DDR4 3200 MHz	4 GB DDR4 3200 MHz	8 GB DDR4 3200 MHz
Хранилище	80 GB	80 GB	80 GB
Сетевой адаптер	1 Gb/s Ethernet	1 Gb/s Ethernet	1 Gb/s Ethernet

В качестве пограничного маршрутизатора в тестовом стенде предлагается использовать Keenetic Giga (KN-1011) 1 Gb/s Ethernet, который позволит обеспечить необходимую производительность сети для проведения тестирования

3.3.2 Описание тестового стенда

Тестовый стенд представляет собой сеть из объединённых устройств, защищаемой инфраструктуры, состоящую из следующих элементов:

- Пограничный маршрутизатор;
- Защищаемые устройства;
- Рабочая станция оператора информационной безопасности

Также для проведения тестирования в стенд включена рабочая станция для имитации атаки на защищаемую инфраструктуру, установленная на ПЭВМ с характеристиками, позволяющими перегрузить защищаемое оборудование (в данном случае IoT-устройства) до состояния его неработоспособности в части приема и передачи сетевых пакетов.

Тестовый стенд также подключен к Глобальной Сети Интернет для более реалистичного сценария тестирования в части «подмешивания» стандартных сетевых пакетов в собираемые данные.

На схеме представлена расположение элементов сетевой инфраструктуры тестового стенда (атакующая станция также включена в локальную сеть стенда) (см. Рисунок 7).

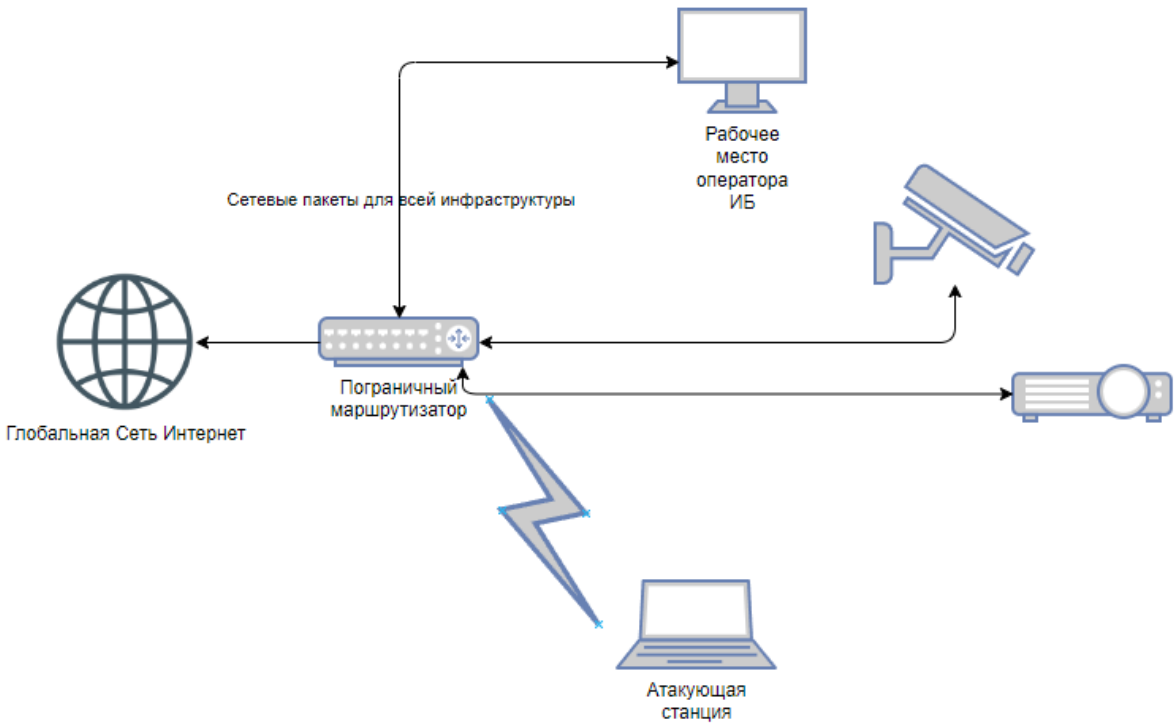


Рисунок 7 – Схема элементов сетевой инфраструктуры

3.3.3 Анализ входных параметров для работы модели

Модель принимает на входе данные в формате *csv**, в котором присутствуют все категориальные признаки, на которых она была обучена (см. Рисунок 8). Входные данные включают в себя исчерпывающую информацию обо всем сетевом трафике, проходящем через пограничный маршрутизатор (см. Таблица 5).

ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	service	duration	orig_bytes	resp_bytes	conn_state	missed_bytes	history	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes
1547066205.2400131	CSLG6c2pmfwFMOS6aa	192.168.1.194	50234.0	15.169.43.124	22.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1551381076.058894	Cpv7rs4pll4kNg5ov2	192.168.1.200	56962.0	70.78.104.175	23.0	tcp	-	3.086517	0	0	S0	0.0	S	6.0	360.0	0.0	0.0
1551383673.692914	C5rck64un3HgSk0as8	192.168.1.193	24159.0	217.115.185.40	8081.0	tcp	-	0.0005059999999999	0	0	S0	0.0	S	6.0	240.0	0.0	0.0
1551384807.804695	CDyMPd3f91ZER0wCfg	192.168.1.193	30535.0	217.208.228.177	8081.0	tcp	-	0.000505	0	0	S0	0.0	S	6.0	240.0	0.0	0.0
1551385876.999664	CyHYWj3OjIL8TmnyV7g	192.168.1.193	30535.0	217.231.172.227	8081.0	tcp	-	0.000259	0	0	S0	0.0	S	6.0	240.0	0.0	0.0
1532526266.0046651	C9SMrj3iZHTpaRFHj	192.168.100.111	52584.0	124.6.205.5	23.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1551385594.713353	C4PsGH2bWuDUUpOJt1	192.168.1.193	24159.0	217.1.224.207	8081.0	tcp	-	5e-06	0	0	S0	0.0	S	2.0	80.0	0.0	0.0
1552049029.450804	CwLpDX1dfTpTHZk9D6	192.168.1.197	63420.0	48.174.21.220	23.0	tcp	-	2e-06	0	0	S0	0.0	S	2.0	80.0	0.0	0.0
1545337129.576454	Cm6srq33WPS43zr8gc	192.168.1.197	43746.0	73.196.60.110	80.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1532527745.0062509	Czmw7G3J7wUK4KyxC1	192.168.100.111	58855.0	116.40.228.123	81.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1532526593.000497	Cj6JyD1g1fShYWxs7	192.168.100.111	34868.0	100.36.183.195	81.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1532528205.005903	CEcxkU21oAV9Zpooy4	192.168.100.111	49824.0	113.115.155.57	23.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1551380938.424754	CcfiGm16PBGQa2Uctd	192.168.1.200	39938.0	146.125.56.31	23.0	tcp	-	3.117476	0	0	S0	0.0	S	6.0	360.0	0.0	0.0
1547144851.100068	Cp0bk18OmjGQkiJec	192.168.1.198	22568.0	91.5.110.220	52869.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1551380041.538061	CVZb922IOtPuTI8Q0k	192.168.1.200	60728.0	145.142.95.56	23.0	tcp	-	3.131958	0	0	S0	0.0	S	6.0	360.0	0.0	0.0
1532525460.007534	C0zP0x1TH5xY3ga2l7	192.168.100.111	38486.0	87.243.226.58	23.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1551384579.5985029	C3yNDe3guOD7VnPUce	192.168.1.193	24159.0	197.207.87.62	8081.0	tcp	-	0.0002549999999999	0	0	S0	0.0	S	4.0	160.0	0.0	0.0
1547144889.910395	CJofXQ1GjlbhX8oLk1	192.168.1.198	22568.0	185.80.227.185	52869.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1545337062.65563	CJ3Scc3s1HR1Amwvg8	192.168.1.197	36114.0	34.169.229.142	8081.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1552048880.64429	CNGu384hR03l20ffgV9	192.168.1.197	63420.0	88.101.96.74	23.0	tcp	-	2e-06	0	0	S0	0.0	S	2.0	80.0	0.0	0.0
1547066122.854724	CNBibS2QMdZTlOkZd3	192.168.1.194	43808.0	30.136.39.48	22.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1547066099.149982	CcteD8jKb3aIO7h5	192.168.1.194	36818.0	183.173.152.228	22.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1547066054.468202	CIZc3QN8eoGHOT3d4	192.168.1.194	38402.0	143.93.123.229	22.0	tcp	-	-	-	-	S0	0.0	S	1.0	40.0	0.0	0.0
1551385390.883292	CvbDC23BveoyJWu9Lk	192.168.1.193	24159.0	217.46.221.236	8081.0	tcp	-	5e-06	0	0	S0	0.0	S	2.0	80.0	0.0	0.0
1536227448.87194	CgCDx44r4zUVpWMH2	192.168.100.111	18088.0	212.164.75.241	80.0	tcp	-	2e-06	0	0	S0	0.0	S	2.0	80.0	0.0	0.0

Рисунок 7 – Формат входных данных

Таблица 5 – Описание параметров

№ п/п	Название признака	Описание признака
1	ts	временная метка (timestamp), указывающая время отправки пакета
2	uid	уникальный идентификатор соединения
3	id.orig_h	IP-адрес отправителя
4	id.orig_p	порт отправителя
5	id.resp_h	IP-адрес получателя
6	id.resp_p	порт получателя;
7	proto	протокол сетевого уровня;
8	service	служба сетевого уровня;
9	duration	длительность соединения
10	orig_bytes	количество байт, отправленных от отправителя
11	resp_bytes	количество байт, отправленных от получателя
12	conn_state	состояние соединения (например, установлено, завершено и т.д.)
13	missed_bytes	количество пропущенных байт
14	history	история событий соединения
15	orig_pkts	количество пакетов, отправленных отправителем
16	orig_ip_bytes	количество байт, отправленных отправителем, без учета заголовков

17	resp_pkts	количество пакетов, отправленных получателем
18	resp_ip_bytes	количество байт, отправленных получателем, без учета заголовков

Все эти параметры представляют информацию о передаче данных в сети, которая может использоваться для анализа, обучения систем и модельного тестирования. Например, параметры IP-адресов и портов отправителя и получателя могут использоваться для определения того, какие устройства взаимодействовали в передаче данных. Параметры протокола и службы могут использоваться для обнаружения потенциальных атак на сеть. Параметры длительности и количества пакетов могут использоваться для определения типа соединения, его продолжительности и структуры.

3.3.4 ПО используемые для подготовки данных

Для захвата данных сетевого траффика с подконтрольной инфраструктуры предлагается использовать ПО Wireshark – инструмент для захвата и анализа сетевых пакетов, который предоставляет детальный анализ сетевого трафика. Он позволяет захватывать пакеты в реальном времени и сохранять их в различных форматах для последующего анализа. (см. Рисунок 8).

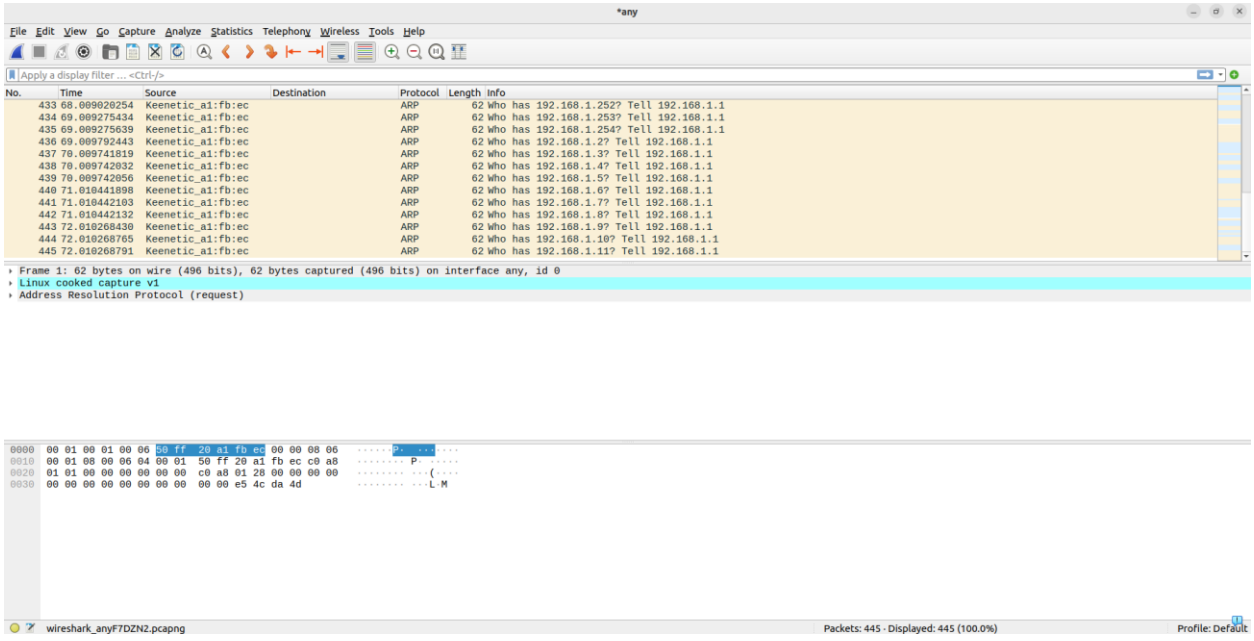


Рисунок 8 – Типовое рабочее окно Wireshark

ПО Wireshark позволяет подготовить файл-отчет обо всех захваченных в процессе мониторинга сетевых пакетах сетевой инфраструктуре в формате *.*pcap* который может быть выгружен на рабочей станицей под управлением большинства известных Операционных систем, что делает этот инструмент практически универсальным.

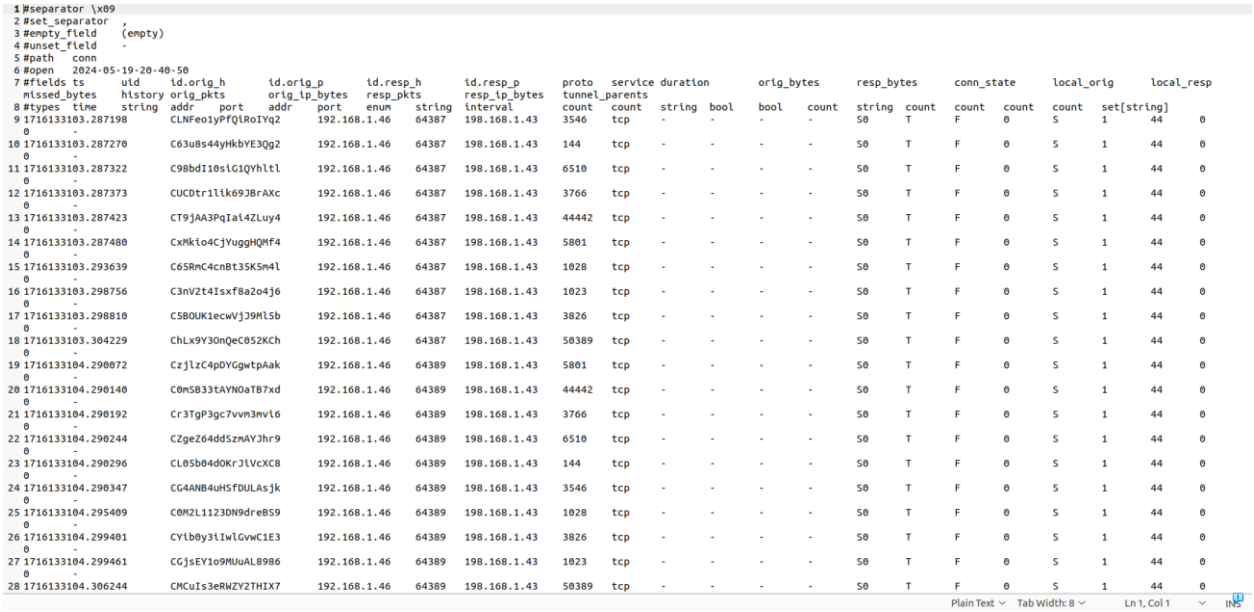
Для подготовки анализируемых данных предлагается использовать ПО Zeek – это фреймворк для анализа сетевого трафика предоставляет широкий набор возможностей для глубокого анализа сетевых событий и генерации детализированных логов. По Zeek предполагается к установке на рабочую станцию под управлением ОС Ubuntu. При анализе файла-отчета *.pcap** ПО Zeek автоматически генерирует лог-файлы, предоставляющие исчерпывающую информацию о сетевых событиях. Основные типы логов включают:

- **conn.log:** информация обо всех сетевых соединениях.
- **http.log**: данные о HTTP-трафике.
- **dns.log:** данные о DNS-запросах и ответах.
- **ssl.log:** информация о SSL/TLS соединениях.

Для использования в модели необходимо подготовить лог-файл conn.log путем преобразования его в формат .csv с использованием собственных возможностей ПО Zeek. (см. Рисунок 9)

head -n 8 conn.log | grep "^#fields" | cut -d' ' -f2- | tr '\t' ',' > conn.csv

zeek-cut < conn.log | sed '/^#/d' | tr '\t' ',' >> conn.csv



```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   conn
#open   2024-05-19-20-40-50
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   service duration        orig_bytes      resp_bytes      conn_state      local_orig      local_resp      missed_bytes    history orig_pkts       orig_ip_bytes   resp_pkts       resp_ip_bytes   tunnel_parents
#types  time    string  addr    port    addr    port    enum    string  interval        count   count   string  bool    bool    count   string  count   count   count   count   set[string]
1716133103.287198       CLNFeo1yPfQlRoIYq2      192.168.1.46    64387   198.168.1.43    3546    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.287270       C63u8s44yHkbYE3Qg2      192.168.1.46    64387   198.168.1.43    144     tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.287322       C98bdI10siG1QYhltl      192.168.1.46    64387   198.168.1.43    6510    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.287373       CUCDtr1lik69JBrAXc      192.168.1.46    64387   198.168.1.43    3766    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.287423       CT9jAA3PqIai4ZLuy4      192.168.1.46    64387   198.168.1.43    44442   tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.287480       CxMkio4CjYuggHQMf4      192.168.1.46    64387   198.168.1.43    5801    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.293639       C65RmC4cnBt3SK5m4l      192.168.1.46    64387   198.168.1.43    1028    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.298756       C3nV2t4Isxf8a2o4j6      192.168.1.46    64387   198.168.1.43    1023    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.298810       C5BOUK1ecwVjJ9Ml5b      192.168.1.46    64387   198.168.1.43    3826    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133103.304229       ChLx9Y3OnQeC052KCh      192.168.1.46    64387   198.168.1.43    50389   tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.290072       CzjlzC4pDYGgwtpAak      192.168.1.46    64389   198.168.1.43    5801    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.290140       C0mSB33tAYNOaTB7xd      192.168.1.46    64389   198.168.1.43    44442   tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.290192       Cr3TgP3gc7vvm3mvi6      192.168.1.46    64389   198.168.1.43    3766    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.290244       CZgeZ64ddSzmAYJhr9      192.168.1.46    64389   198.168.1.43    6510    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.290296       CL0Sb04dOKrJlVcXC8      192.168.1.46    64389   198.168.1.43    144     tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.290347       CG4ANB4uHSfDULAsjk      192.168.1.46    64389   198.168.1.43    3546    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.295409       C0M2L1123DN9dreBS9      192.168.1.46    64389   198.168.1.43    1028    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.299401       CYib0y3iIwlGvwC1E3      192.168.1.46    64389   198.168.1.43    3826    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.299461       CGjsEY1o9MUuAL8986      192.168.1.46    64389   198.168.1.43    1023    tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
1716133104.306244       CMCuIs3eRWZY2THIX7      192.168.1.46    64389   198.168.1.43    50389   tcp     -       -       -       -       S0      T       F       0       S       1       44      0       0       -
```

Plain Text Tab Width: 8 Ln 1, Col 1 INS

Рисунок 9 – Типовая структура файла conn.log

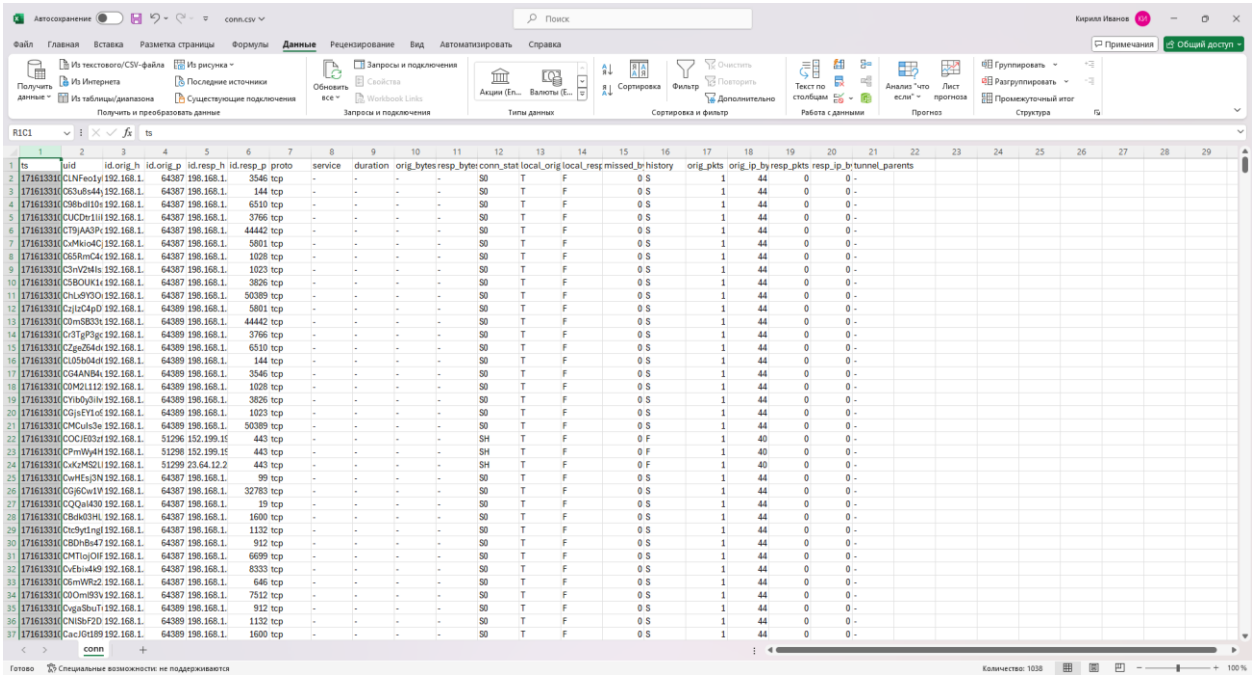


Рисунок 10 – Типовая структура подготовленного к анализу отчета

3.3.5 Процесс тестирования

После процессов подготовки оборудования и данных проведено статическое тестирование различных конфигураций системы на производительность на наборах данных различных размеров.

Порядок работы системы описан блоком кода (см. Рисунок 11), в котором обрабатываются категориальные признаки (указаны разработчиком модели) подготовленных ранее входных данных и далее выводятся результаты о обнаруженных угрозах

```python
def load_model_and_predict(input_data_path, model_path='catboost_model.cbm'):
    # Загрузка данных
    data = pd.read_csv(input_data_path, on_bad_lines='warn')
    data = data.replace(['-'], '0.0')
    data = data.replace(['0'], '0.0')
    cols_to_drop = ['uid', 'Unnamed: 0', 'label']
    data = data.drop(cols_to_drop, axis=1)

    categorical_features = ['id.orig_h', 'id.orig_p', 'id.resp_h', 'id.resp_p', 'proto',
                            'service', 'conn_state', 'local_orig',
                            'local_resp', 'history']
    data = pd.get_dummies(data, columns=categorical_features)


    data = data.fillna(0)

    numeric_features = ['duration', 'orig_bytes', 'resp_bytes']
    for col in numeric_features:
        data[col] = pd.to_numeric(data[col], errors='coerce').fillna(0)


    model = CatBoostClassifier()
    model.load_model(model_path)


    predictions = model.predict(data)

    return predictions

# Пример использования функции
input_data_path = "iot23_combined_new.csv"
predictions = load_model_and_predict(input_data_path)
print(predictions)
```

Рисунок 11 – Блок кода, описывающий порядок работы системы.

Результаты тестирования производительности для каждой конфигурации оборудования представлены в Таблице 6.

Таблица 6 – Результаты тестирования

Тестовый набор	**Конфигурация 1**	**Конфигурация 2**	**Конфигурация 3**
Тест 1 (Набор данных 800 MB)	40 сек	1 мин. 40 сек.	1 мин
Тест 1 (Набор данных 350 MB)	10 сек	40 сек.	20 сек.

По результатам тестирования можно сделать вывод, что оптимальной по производительности является *Конфигурация 3*. *Конфигурация 1* является слишком дорогостоящей для массового использования, *Конфигурация 2* сильно уступает в производительности несмотря на более низкие затраты.

3.3.6 Интеграция в инфраструктуру

В целом процесс интеграции в систему аналогичен подключению рабочей станции оператора информационной безопасности к тестовому стенду.

Для работы системы в промышленной сетевой инфраструктуре блок обработки данных был дополнен алгоритмом подготовки отчета при обнаружении угроз в проанализированных данных (см. Рисунок 12)

```
# Составление отчета
fill = PatternFill(start_color="FF0000", end_color="FF0000", fill_type="solid")

for row_idx in range(2, sheet.max_row + 1):
    if sheet.cell(row=row_idx, column=sheet.max_column).value != 1:
        for col_idx in range(1, sheet.max_column + 1):
            sheet.cell(row=row_idx, column=col_idx).fill = fill

# Сохранениние отчета
workbook.save(output_data_path)
```

Рисунок 12 – Блок подготовки отчета

3.3.7 Вывод по работе системы

Итоговой алгоритм работы системы можно описать следующим образом (см. Рисунок 13)

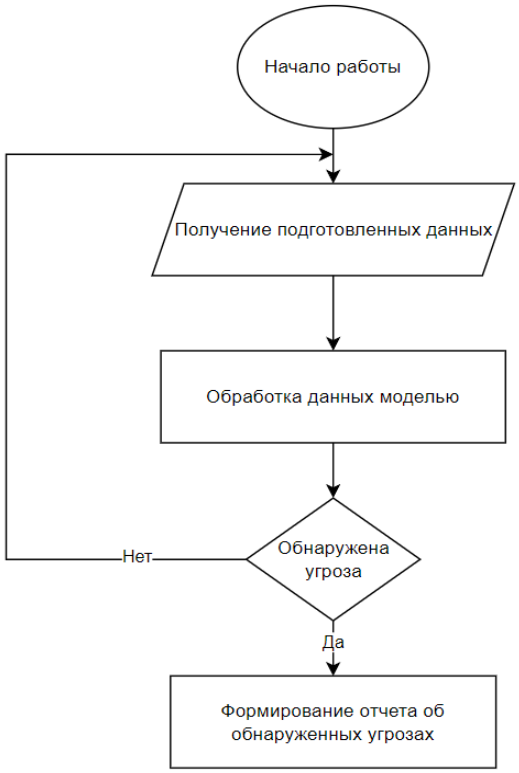


Рисунок 13 – Итоговый алгоритм

Результатом работы алгоритма и системы в целом является готовый отчет со всеми подключениями, представляющими угрозы (помечены красным) для сетевой инфраструктуры с полным описанием сетевого подключения и кратким названием обнаруженной угрозы. Итоговой отчет содержит всю информацию о подключениях, зафиксированных в момент подготовки данных (см. Рисунок 14).

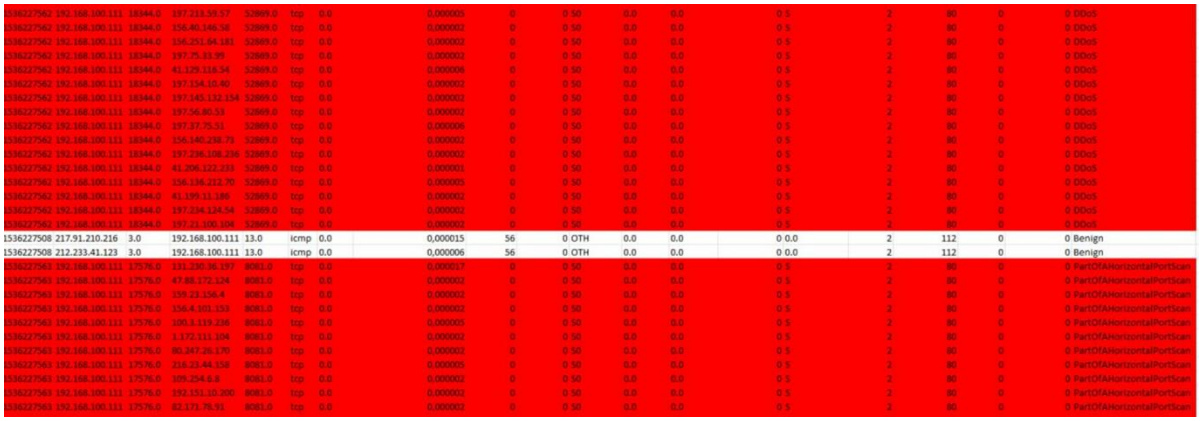


Рисунок 14 – Отчет об обнаруженных угрозах

Данный полученные из отчета используются Аналитиками информационной безопасности для исключения дальнейшей возможности атак с IP-адресов, помеченных в отчете как опасные.

3.3.8 Вывод по интеграции системы

Интегрированная система показала себе высокопроизводительной и отказоустойчивой. Масштабируемость системы в целом зависит от конфигурации оборудования, на котором она установлена с возможность разветвления потоков сетевых данных на логическом уровне между двумя и более рабочими станциями операторов информационной безопасности

Процесс дообучения и переобучения системы требует дополнительного алгоритма, но в целом обучение с нуля при появлении новых угроз устройствам интернета вещей и дополнении имеющегося дата-сета не является трудозатратным, что позволяет обучить новую модель и интегрировать ее с кратковременной остановкой системы при отсутствии такой же резервной системы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Aslan, Ö.,Aktuğ, S.S.,Ozkan-Okay, M.,Yilmaz, A.A.,Akin, E. (2023). A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions. Electronics. 2023; 12(6):1333. https://doi.org/10.3390/electronics12061333

2. Ali, R., Ali, A., Iqbal, F., Hussain, M., Ullah, F. (2022). Deep Learning Methods for Malware and Intrusion Detection: A Systematic Literature Review. Security and Communication Networks. DOI: 10.1155/2022/2959222.

3. Bendiab, G., Shiaeles, S., Alruban, A., Kolokotronis, N. (2020). IoT malware network traffic classification using visual representation and deep learning. DOI: 10.1109/NetSoft48620.2020.9165381. Published in the proceedings of the 6th IEEE Conference on Network Softwarization (NetSoft) 2020, pages 444-449. IEEE.

4. Luo, X., Li, J., Wang, W., Gao, Y., Zhao, W. (2021). Towards improving detection performance for malware with a correntropy-based deep learning method. Digit. Commun. Networks, 7(570-579). DOI: 10.1016/j.dcan.2021.02.003.

5. Котенко И.В., Хмыров С.С. (2022). Анализ моделей и методик, используемых для атрибуции нарушителей кибербезопасности при реализации целевых атак. Вопросы Кибербезопасности Номер: 4(50), страницы 52-79. eLIBRARY ID: 49326500, DOI: 10.21681/2311-3456-2022-4-52-79

6. Ansam Khraisat, Ammar Alazab (2021). A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity 4, Article number: 18. https://doi.org/10.1186/s42400-021-00077-7

7. Гайфулина Д.А., Котенко И.В. (2021). Анализ моделей глубокого обучения для задач обнаружения сетевых аномалий Интернета вещей. Информационно-управляющие системы, (1), 28-37. https://doi.org/10.31799/1684-8853-2021-1-28-37

8. Р.М. Ауси, Е.В. Заргарян, Ю.А. Заргарян (2022). Модели машинного обучения и глубокого обучения для электронной информационной безопасности в мобильных сетях. Известия ЮФУ. Технические Науки. Выпуск №3 (2022). Раздел III. Моделирование процессов и систем. DOI: 10.18522/2311-3103-2022-3-211-222

9. Р.М. Ауси, Е.В. Заргарян, Ю.А. Заргарян (2023). ГЛУБОКОЕ ОБУЧЕНИЕ МЕТОДАМ ЗАЩИТЫ ОТ АТАК. Известия ЮФУ. Технические Науки. Выпуск №3 (2023). Раздел III. Алгоритмы обработки информации. DOI: 10.18522/2311-3103-2023-2-227-239

10.Ori Or-Meir, Nir Nissim, Y.Elovici, L. Rokach (2019). Dynamic Malware Analysis in the Modern Era—A State of the Art Survey. ACM Computing Surveys, Volume 52, Issue 5, Article No.: 88pp 1–48. https://doi.org/10.1145/3329786

11.Akhtar M.Sh., Feng T. (2022). Detection of Malware by Deep Learning as CNN-LSTM Machine Learning Techniques in Real Time. Symmetry 2022, 14(11), 2308; https://doi.org/10.3390/sym14112308

12.Toan, N. N. ., Dung, L. T., & Thang, D. Q. (2022). Static Feature Selection for IoT Malware Detection. Journal of Science and Technology on Information Security, 1(15), 74-84. https://doi.org/10.54654/isj.v1i15.844

13.S. Gopali, A.S. Namin (2022). Deep Learning-Based Time-Series Analysis for Detecting Anomalies in Internet of Things. Electronics 2022, 11(19), 3205; https://doi.org/10.3390/electronics11193205

14.Патент № WO2021087443A1. INTERNET OF THINGS SECURITY ANALYTICS AND SOLUTIONS WITH DEEP LEARNING / HOLBROOK, Luke. Опубл. 2021.05.06.

15.Патент № US11075934B1. Hybrid network intrusion detection system for IoT attacks / Sahar Ahmed Aldhaheri, Daniyal Mohammed Alghazzawi Опубл. 2021.07.27.

16.Патент № US2023328081A1. SYSTEM AND METHODS FOR AUTOMATIC DETECTION OF DISTRIBUTED ATTACKS IN IOT DEVICES

USING DECENTRALIZED DEEP LEARNING / Najafirad, Peyman; De La Torre Parra, Gonzalo Опубл. 2023.10.12.

17.Зуев, В. Н. Обнаружение аномалий сетевого трафика методом глубокого обучения [Текст] / В. Н. Зуев // Программные продукты и системы. — 2021. — № 1. — С. 91-97.

18.Гурина, А. О., Гузев, О. Ю., Елисеев, В. Л. Обнаружение аномальных событий на хосте с использованием автокодировщика [Текст] / А. О. Гурина, О. Ю. Гузев, В. Л. Елисеев // International Journal of Open Information Technologies. — 2020. — № 8. — С. 26-35.

19.Канатьев, К. Н., Большаков, В. Н., Куприков, О. Д., Горошков, Д. Б., Баулин, Е. И. АНАЛИЗ УГРОЗ БЕЗОПАСНОСТИ БЕСПРОВОДНОЙ СЕТИ И РАЗРАБОТКА ОПТИМАЛЬНЫХ МЕТОДОВ ИХ ПРЕДУПРЕЖДЕНИЯ [Текст] / К. Н. Канатьев, В. Н. Большаков, О. Д. Куприков, Д. Б. Горошков, Е. И. Баулин // Инновации и инвестиции. — 2020. — № 3. — С. 116-123.

20.Гантимуров, А. П. Анализ и синтез распределенных систем хранения данных : специальность «Вычислительные машины, комплексы и компьютерные сети» : Диссертация на соискание кандидата технических наук / Гантимуров, А. П. ; ФГБОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)». — Москва, 2022. — 94 с.

21.Вапник В. Н. Статистическая теория обучения на примерах. / Вапник В. Н. — 1-е изд.. — Москва: Физматлит, 2020 — 640 с.

22.Bishop C. M. Pattern Recognition and Machine Learning. / Bishop C. M. — 1-е изд.. — Нью-Йорк: Springer, 2020 — 738 с.

23.Титов Д. Н. ОБНАРУЖЕНИЕ ВТОРЖЕНИЙ В СИСТЕМУ ИНТЕРНЕТА ВЕЩЕЙ / Титов Д. Н. // Интерэкспо Гео-Сибирь. — 2022. — № 8. — С. 118-125.

24.Татарникова, Т. М., Богданов, П. Ю. ОБНАРУЖЕНИЕ АТАК В СЕТЯХ ИНТЕРНЕТА ВЕЩЕЙ МЕТОДАМИ МАШИННОГО ОБУЧЕНИЯ / Т. М.

Татарникова, П. Ю. Богданов // Информационно-управляющие системы. — 2021. — № 6. — С. 42-52.