Федеральное государственное бюджетное учреждение науки
«Санкт-Петербургский Федеральный исследовательский центр
Российской академии наук»
(СПб ФИЦ РАН)

На правах рукописи

Крибель Александр Михайлович

ВЫЯВЛЕНИЕ АНОМАЛИЙ И КЛАССИФИКАЦИЯ КОМПЬЮТЕРНЫХ АТАК В СЕТИ ПЕРЕДАЧИ ДАННЫХ НА ОСНОВЕ ПРИМЕНЕНИЯ ФРАКТАЛЬНОГО АНАЛИЗА И МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

Специальность 2.3.6 – Методы и системы защиты информации,
информационная безопасность

Диссертация на соискание ученой степени
кандидата технических наук

Научный руководитель
доктор технических наук
Лаута Олег Сергеевич

Санкт-Петербург – 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Актуальность темы диссертации. Темпы, с которыми развивается современная сфера информационных технологий, подвергает мировое сообщество целому ряду беспрецедентных угроз и уязвимостей, которые злоумышленнику открывают возможности к реализации компьютерных атак.

Компьютерные атаки представляют собой сложное комплексное воздействие на сеть, в результате которого осуществляется их компрометации и нарушается управление процессами в сети передачи данных. Зачастую этому предшествует долгая и кропотливая работа: компьютерная техническая разведка, поиск характерных уязвимостей и захват информационных активов. Воздействие компьютерных атак возможно за счет использования технологий сбора информации, малоэффективных механизмов защиты, эксплуатации устаревших сетевых служб, протоколов и операционных систем.

К категории опасных можно отнести следующие сервисы: сервисы доступа к файловой системе; *RPC*; службы каталогов; принтеры; сервисные интерфейсы систем виртуализации; *VPN*; специфичные для сетей передачи данных системы; сервисы сетевых устройств; *Telnet*; *SSH*; *RDP*; *VNC* и т.п. Кроме того, следует отметить, что недостатки защиты служебных протоколов, приводящие к перенаправлению трафика и перехвату информации о конфигурации сети, недостатки защиты протоколов *NBNS* и *LLMNR*, а также использование открытых (незащищенных) протоколов передачи данных в современных сетях передачи данных имеют высокий уровень риска. Как показывает практика, подавляющее большинство успешных компьютерных атак основаны на эксплуатации уязвимостей каких-либо ресурсов, которые не должны быть доступны на сетевом периметре.

Использование подобных уязвимостей даёт злоумышленнику возможность несанкционированно авторизовываться в системе, повышать пользовательские привилегии, изменять настройки сетевых устройств, прослушивать и

перенаправлять трафик, блокировать сетевое взаимодействие и нарушать информационный обмен в сетях передачи данных.

Кроме того, необходимо отметить, что недостатки защиты служебных протоколов, приводящие к перехвату информации о конфигурации сети и перенаправлению трафика, в современных сетях передачи данных имеют высокий уровень риска.

Воздействия компьютерных атак приводит к появлению в сетях передачи данных аномальной активности трафика [1]. Для постоянного мониторинга и обнаружения аномальной активности трафика в сетях передачи данных необходимо учитывать наличие большого количества сетевых маршрутов, на которых периодически возникают резкие колебания, задержки в передаче данных и большие потери пакетов, появление новых свойств сетевого трафика, а также необходимость обеспечения высокого качества обслуживания приложений. Все это послужило поводом для поиска новых методов обнаружения и прогнозирования компьютерных атак, к числу которых можно отнести машинное обучение и фрактальный анализ [2-3].

Наличие фрактальных свойств в сетевом трафике было обнаружено несколько десятилетий назад, когда было установлено, что на больших масштабах он обладает свойством самоподобия, то есть выглядит качественно одинаково при достаточно больших масштабах временной оси и проявляет долговременную зависимость. Господствовавшие до этого модели трафика, основанные на Марковских процессах, обладали кратковременной зависимостью. Они были заимствованы из телефонных сетей и, применительно к компьютерным сетям приводили к недооценке нагрузки.

Фрактальный анализ, как метод исследования математических множеств различной природы, базируется на идеях фрактальной геометрии, разработанной Б. Мандельбротом. Начиная с 1973 года, когда была опубликована его основополагающая работа, методы фрактального анализа нашли широкое применение в различных областях физики, химии, биологии. Главное достижение теории фракталов состоит в том, что она даёт простые способы математического

описания весьма сложных, но очень широко распространенных в природе, явлений и объектов. Фрактальный анализ является столь же фундаментальным математическим аппаратом для описания физической реальности, как дифференциальные уравнения, тригонометрия или гармонический анализ. Однако, в связи с тем, что он был открыт относительно недавно, он еще не занял подобающего места в умах ученых и инженеров [1-5].

Анализ показал, что наиболее эффективным методом классификации и прогнозирования является алгоритм LSTM (long short-term memory)-нейронной сети (LSTM). Свойство рекуррентности позволяет искусственной нейронной сети (ИНС) «обращаться» к результатам своей работы в прошлом, делать анализ предикций. Тем самым контекст решений по выработке мероприятий по защите от компьютерных атак в будущем будет зависеть не только от первичного глубокого обучения LSTM, но и её дальнейшей работы в потоке [6].

Ключевой параметр фрактального анализа – показатель Херста. Эту меру используют при анализе временных рядов. Чем больше задержка между двумя одинаковыми парами значений во временном ряду, тем меньше коэффициент Херста (или показатель скейлинга). Для нахождения этого параметра необходимо знать, стационарен ли исследуемый процесс. От этого зависит выбор алгоритма для дальнейшего вычисления скейлинга [2, 5].

Таким образом, с целью обнаружения и классификации компьютерных атак в первую очередь необходимо определить, стационарный трафик или нестационарный. Далее следует рассчитать показатель Херста (т. е. определить наличие в трафике свойства самоподобия). На заключительном этапе происходит обнаружение аномалий и с применением LSTM выработать мероприятия по защите сети передачи данных.

Важность и значимость решаемой задачи заключается в том, что на основе экспериментальных исследований возможно обосновать наилучший метод определения самоподобия для нестационарных и стационарных временных рядов, позволяющий с высокой точностью и достаточно быстро обнаруживать изменения в сетевом трафике сетей передачи данных, а также определить

структуру сети *LSTM*, позволяющую с высокой точностью и достаточно быстро прогнозировать факт воздействия компьютерных атак, на основе которого в дальнейшем могут вырабатываться проактивные мероприятия защиты.

Степень разработанности темы. В настоящее время вопросы, связанные с изучением самоподобных свойств временных рядов и их практическим применением в различных системах мониторинга, находятся в фокусе внимания многих исследователей. Фрактальные свойства исследуются во многих работах. Так, в работе *Raimundo M.S.* метод *R/S*-анализа используется для выявления закономерностей во временных рядах. В работе *Dang T.D.* моделируется *VoIP*-трафик, а также исследуются его фрактальные свойства. В работе *Sánchez-Granero M.J.* изучался не только показатель Херста, но и фрактальная размерность. В работе *Grillo D.* авторы объясняют, почему телетрафик обладает фрактальными свойствами [5, 6].

При этом следует отметить, что существует мало практических экспериментов, направленных на изучение фрактальных свойств сетевого трафика телекоммуникационных систем. Среди такого рода работ можно выделить работы *Strelkovskaya I.* и *Carvalho P*. Однако, в работе *Strelkovskaya I.* трафик рассматривается не в телекоммуникационных сетях, а в радиоволнах, передаваемых сотовыми станциями. Кроме того, исследователи приходят к выводу, что движение самоподобно, часто полагаясь исключительно на визуальные знаки *Carvalho P.; Abdalla H.; Soares A.; Solis P.; Tarchetti P*. Они ищут похожие участки на графике, выдавая их за самоподобные процессы. Однако эти работы в основном охватывают финансовый сектор и VoIP-телефонию.

В тоже время большое внимание вопросам противодействия компьютерным атакам, выявлению аномалий и классификации компьютерных атак в сетях передачи данных уделяется такими исследователями как Д.А. Губанов, И.В. Котенко, М.В. Литвиненко, Д.А. Новиков, И.Б. Саенко, А.Л. Тулупьев, Д.Ю. Турдаков, А.А. Чечулин, А.Г. Чхартишвили, *A.L. Barabasi*, *X. Zheng* и др. Работы С.А. Будникова, Ю.Л. Козирацкого, В.А. Липатникова, С.И. Макаренко, С.П.

Расторгуева, Д.В. Сахарова и др. посвящены информационному конфликту и противоборству. Вопросы информатизации процессов и оценивания эффективности информационных систем раскрываются в работах М.В. Буйневича, В.П. Заболотского, А.А. Мусаева, П.Ю. Филяка, Р.М. Юсупова [7-13]

Учитывая требования нормативно-правовых документов [14-30], а также несмотря на сделанный учеными существенный задел, проблема выявления аномалий, раннего обнаружения известных и неизвестных компьютерных атак в сетях передачи данных, их классификации не может считаться разрешенной и требует проведения новых исследований.

Таким образом, сложилось **противоречие** между возросшими деструктивными возможностями новых видов компьютерных атак на сети передачи данных, приводящих к аномальной активности трафика, и устаревшими подходами к их выявлению в сетях передачи данных, которое и предопределило выбор объекта и предмета исследования.

Разрешение указанного противоречия предопределяет необходимость изучения целого ряда теоретических и практических вопросов, удовлетворительное решение по которым до настоящего времени не получено, что обуславливает актуальность темы настоящей работы.

Цель исследования. Целью диссертационной работы является повышение эффективности выявления аномалий и классификации компьютерных атак в сетевом трафике сети передачи данных.

Научная задача заключается в разработке аналитической модели и методик выявления аномалий и классификация компьютерных атак в сетевом трафике сети передачи данных на основе применения фрактального анализа и методов машинного обучения.

Для достижения данной цели в диссертационной работе поставлены и решены следующие **частные задачи**:

1) анализ существующих моделей воздействия компьютерных атак в сети передачи данных;

2) анализ существующих алгоритмов выявления компьютерных атак, систем мониторинга и методик противодействия компьютерным атакам в сети передачи данных;

3) разработка аналитической модели выявления аномалий в сетевом трафике сети передачи данных в условиях компьютерных атак;

4) разработка методики раннего обнаружения аномалий в сетевом трафике сети передачи данных;

5) разработка методики классификации компьютерных атак в сетевом трафике сети передачи данных;

6) разработка архитектуры и программных прототипов компонентов системы раннего обнаружения и классификации компьютерных атак в сетевом трафике сети передачи данных, экспериментальная и теоретическая оценка эффективности предложенных моделей, методик и архитектуры.

Объектом исследования являются сети передачи данных в условиях компьютерных атак, а **предметом исследования** – модели, методики и алгоритмы выявления аномалий и классификации компьютерных атак в сетях передачи данных.

Научная новизна результатов исследования заключается в том, что:

разработанная аналитическая модель выявления аномалий в сетевом трафике сетей передачи данных в условиях компьютерных атак, в отличие от известных, описывает не только стационарный, но и нестационарный сетевой трафик и обосновывает выбор фрактального метода выявления аномалий в интересах дальнейшей классификации компьютерных атак в зависимости от типа трафика;

разработанная методика раннего обнаружения аномалий в сетевом трафике сетей передачи данных, в отличие от известных, способна проводить прогнозирование и обнаружение не только известных, но и неизвестных компьютерных атак на раннем этапе их проявления благодаря совместному применению методов фрактального анализа и искусственной нейронной сети *LSTM*-типа (рекуррентные нейронные сети с долгой краткосрочной памятью), в

результате чего существенно сокращается время выявления аномалий и уменьшается количество ложных срабатываний;

разработанная методика классификации компьютерных атак в сетевом трафике сетей передачи данных отличается от известных тем, что в ней обнаружение компьютерных атак производится с использованием генеративно-состязательной сети, которая производит дообучение классификатора на скрытых латентных представлениях, полученных автокодировщиком, исходя из анализа данных о функционировании сетей передачи данных;

предложенная архитектура и программные прототипы компонентов системы раннего обнаружения и классификации компьютерных атак в сетях передачи данных отличаются от известных тем, что они ориентированы на раннее обнаружение как известных, так и неизвестных компьютерных атак с минимальным количеством ложных срабатываний за счет реализации методов фрактального анализа и искусственной нейронной сети *LSTM*-типа.

Теоретическая и практическая значимость результатов исследования заключается в том, что разработанные аналитическая модель и методики представляют собой научно-методическую основу, практическая реализация которой позволяет описать различные типы трафика в сетях передачи данных, определять аномальные активности, основываясь на принципах самоподобия, и, исходя из типа трафика с применением различных методов машинного обучения, выявлять компьютерные атаки. Разработанные методики являются математической основой системы раннего обнаружения компьютерных атак, основанные на обнаружении аномалий в сетях передачи данных и принятии эффективных мероприятий по защите с применением нейронной сети автокодировщика, состоящей из ячеек с долгой краткосрочной памятью и управляющим рекуррентным блоком. При этом повышается оперативность, точность и полнота выявления аномалий в сетях передачи данных, что позволяет на практике эффективно применять разработанный подход в системах глубокой проверки сетевых пакетов в сетях передачи данных.

Методология и методы исследования. В качестве математических положений исследования использованы: фрактальный анализ; методы машинного обучения; теория и практика систем связи; теория и практика проведения тестирования на проникновение; аналитико-статистические методы.

Положения, выносимые на защиту:

1. Аналитическая модель выявления аномалий в сетевом трафике сети передачи данных в условиях компьютерных атак.

2. Методика раннего обнаружения аномалий в сетевом трафике сети передачи данных.

3. Методика классификации компьютерных атак в сетевом трафике сети передачи данных.

4. Архитектура и программные компоненты системы раннего обнаружения и классификации компьютерных атак в сетевом трафике сети передачи данных.

Соответствие диссертации паспорту научной специальности. Представленные результаты соответствуют паспорту специальности 2.3.6 – «Методы и системы защиты информации, информационная безопасность».

Степень достоверности результатов диссертационной работы подтверждается результатами вычислительных экспериментов, их сравнением с результатами других исследователей, практической апробацией разработанных модели и методик, а также одобрением основных положений диссертации на научно-технических конференциях, публикациями в ведущих рецензируемых журналах, внедрением результатов работы.

Апробация результатов. Основные научные и практические результаты работы докладывались и обсуждались на 10 научно-технических и научно-практических конференциях с 2017 по 2022 гг., к основным из которых относятся: Международная научно-практическая конференция «РусКрипто» (Московская область, 2021 и 2022); Межвузовская научно-практическая конференция «Актуальные проблемы обеспечения информационной безопасности» (Самара, 2017); Двенадцатая общероссийская молодежная научно-техническая конференция «Молодежь. Техника. Космос.» (Санкт-Петербург, 2020);

Всероссийская научно-техническая конференция «Состояние и перспективы развития современной науки по направлению «Информационная безопасность»» (Анапа, 2021 и 2020); Всероссийская научно-техническая конференция «Состояние и перспективы развития современной науки по направлению «АСУ, информационно-телекоммуникационные системы»» (Анапа, 2021); Всероссийская научно-практическая конференция РАРАН «Актуальные проблемы защиты и безопасности» (Санкт-Петербург, 2019); Международная научно-техническая и научно-методическая конференция «Актуальные проблемы инфотелекоммуникаций в науке и образовании» (Санкт-Петербург, 2019); Всеармейская научно-практическая конференция «Инновационная деятельность в Вооруженных Силах Российской Федерации», (Санкт-Петербург, 2017).

Внедрение и реализация результатов исследования. Результаты проведенного исследования нашли практическое применение в разработках, в которых автор принимал личное участие. О реализации основных результатов проведенного исследования имеются 3 акта о реализации в НИР «Корвет», «Потенциал-2018», «Свертка-СМ» (ФГКВОУ ВО «Военная орденов Жукова и Ленина Краснознаменная академия связи им. С.М. Буденного») и 1 акт о реализации в ОКР «Опорник» (ПАО «Информационные и теле-коммуникационные технологии».

Публикации по теме исследования. По тематике диссертации опубликованы семнадцать работ, среди которых шесть статей, индексируемые в международных базах данных *Web of Science* и/или *Scopus*; десять статей в рецензируемых научных изданиях, входящих в перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук, на соискание ученой степени доктора наук; одно свидетельство о государственной регистрации программы для ЭВМ.

Опубликованы статьи в следующих журналах из «перечня ВАК»: «Электросвязь»; «Известия Тульского государственного университета. Технические науки»; «Наукоемкие технологии в космических исследованиях

Земли»; «Робототехника и техническая кибернетика»; «Первая миля». Полный перечень публикаций и приравненных к ним работ представлен в приложении А диссертации.

Личный вклад соискателя. Результаты по положениям, выносимым на защиту в диссертационной работе, получены автором самостоятельно, в частности разработаны: модель выявления аномалий в сетевом трафике сетей передачи данных в условиях компьютерных атак; методики раннего обнаружения аномалий в сетевом трафике сети передачи данных и классификации компьютерных атак в сетевом трафике сети передачи данных. Разработан и зарегистрирован в установленном порядке программный компонент обнаружения аномалий сетевого трафика на основе принципов фрактального анализа данных, а также опубликованы самостоятельно и в соавторстве прочие результаты, при этом вклад соискателя в совместных публикациях был решающим.

Структура и объем диссертации. Диссертационная работа состоит из введения, четырех разделов, заключения, списка сокращений, списка литературы. Работа выполнена на 144 страницах, выполненных печатным способом, содержит 77 рисунков, 5 таблиц и 4 приложения.

Краткое содержание работы.

В первом разделе проведён анализ состояния сетей передачи данных и рассмотрены требования, предъявляемые к ней. Проведён анализ информационных воздействий злоумышленника, как одного из основных факторов, влияющих на устойчивость функционирования сетей передачи данных. Рассмотрены существующие средства и способы защиты сетей передачи данных. Проведён анализ существующих исследований, направленных на противодействие компьютерным атакам. Определены их достоинства, выделены основные недостатки, затрудняющие противодействие компьютерным атакам. Обоснована актуальность цели исследования. Предложено использование методик, основанных на методах машинного обучения для решения поставленной в исследовании цели. Итогом проведенного анализа является формальная постановка задачи и определение критериев для оценки эффективности.

Во втором разделе разработана аналитическая модель выявления аномалий в сетевом трафике сетей передачи данных в условиях компьютерных атак, предназначенная для описания сетевого трафика сразу двух типов: стационарного и нестационарного. Для проверки стационарности ряда используется обобщенный тест Дики-Фуллера. Также модель позволяет принимать и классифицировать сетевой трафик сетей передачи данных процесса, участвующий в обмене данными из сети Интернет. В зависимости от классификации сетевого трафика выбирается методика по выявлению аномалий.

С целью определения аномальной активности в сетях передачи данных применяются: принципы самоподобия для нестационарного трафика, который нарушается при возникновении аномальной активности в сети; методы машинного обучения для стационарного трафика. При экспериментальной проверке разработанной модели для нахождения показателя Херста использовались *R*/*S* анализ и метод *DFA*.

В третьем разделе разработана методика раннего обнаружения аномалий в сетевом трафике сетей передачи данных, позволяющая обнаруживать компьютерные атаки на раннем этапе их проявления с помощью методов машинного обучения для стационарного сетевого трафика сетей передачи данных и фрактального анализа для нестационарного. Методика состоит из следующих этапов: сбор сетевого трафика в сетях передачи данных; проверка на стационарность; подготовка исходных данных; фрактальный анализ для нестационарного сетевого трафика сетей передачи данных; машинное обучение для стационарного сетевого трафика сетей передачи данных. Проведен эксперимент по нахождению оптимального числа пакетов, необходимых для точного выявления аномалий с помощью фрактального анализа. Для обнаружения аномалий в стационарном трафике разработана модель автокодировщика, позволяющая эффективно выявлять компьютерные атаки «нулевого дня» в стационарном сетевом трафике, с минимальным количеством ложных срабатываний. Произведена оценка точности разработанной модели. Проведена оценка возможности раннего обнаружения компьютерных атак в сетях передачи

данных с помощью классификаторов, методов математической статистики и автокодировщика, построенного на основе нейронной сети с долгой краткосрочной памятью.

В четвертом разделе разработана методика классификации компьютерных атак в сетевом трафике сетей передачи данных, позволяющая выявлять компьютерные атаки с использованием гибридной нейронной сети, состоящей из классификатора и автокодировщика, обученного на основе данных работы функционирования сетей передачи данных, учитывающего все отклонения от ее штатной работы. В процессе работы классификатор дополнительно обучается на скрытых латентных представлениях полученных автокодировщиком, т.е. в итоге получается генеративно-состязательная сеть, в которой нейронные сети учатся друг у друга.

Также представлена архитектура и программный прототип компонентов системы раннего обнаружения и классификации компьютерных атак в сетевом трафике сетей передачи данных, которые отличаются от известных тем, что ориентированы на мгновенное обнаружение как известных, так и неизвестных компьютерных атак, их классификацию и выбор доступных контрмер с минимальным числом ложных срабатываний.

В заключении приведены основные результаты диссертационной работы и выводы, описаны возможные сценарии применения раннего обнаружения компьютерных атак в сетевом трафике сетей передачи данных, а также сформулированы перспективы области исследования.

РАЗДЕЛ 1 АНАЛИЗ СОСТОЯНИЯ СЕТЕЙ ПЕРЕДАЧИ ДАННЫХ

1.1 Анализ функционирования сети передачи данных

Сети передачи данных (СПД) – комплекс распределенной компьютерной техники, программируемых контроллеров, устройств ввода-вывода информации, соединенных между собой системой передачи данных, содержащей коммуникационное оборудование и каналы связи.

Классификация СПД:

Наибольшее распространение на сегодня получило, разделение компьютерных сетей по признаку территориального размещения. По этому признаку сети делятся на три основных класса:

1. *LAN* – локальные сети (*Local Area Networks*) – коммуникационная система, поддерживающая в пределах здания или некоторой другой ограниченной территории один или несколько высокоскоростных каналов передачи цифровой информации, предоставляемых подключенным устройствам для кратковременного монопольного использования. Территории, охватываемые ЛС, могут существенно различаться. Длина линий связи для некоторых сетей может быть не более 1000 м, другие же ЛС в состоянии обслужить целый город. Обслуживаемыми территориями могут быть как заводы, суда, самолеты, так и учреждения, университеты, колледжи. В качестве передающей среды, как правило, используются коаксиальные кабели, хотя все большее распространение получают сети на витой паре и оптоволокне, а в последнее время также стремительно развивается технология беспроводных локальных сетей, в которых используется один из трех видов излучений: широкополосные радиосигналы, маломощное излучение сверхвысоких частот (СВЧ излучение) и инфракрасные лучи. Небольшие расстояния между узлами сети, используемая передающая среда и связанная с этим малая вероятность появления ошибок в передаваемых данных позволяют поддерживать высокие скорости обмена - от 1 Мбит/с до 100 Мбит/с (в настоящее время уже есть промышленные образцы ЛС со скоростями порядка 1 Гбит/с).

2. *MAN* – городские сети (*Metropolitan Area Networks*). Городские сети, как правило, охватывают группу зданий и реализуются на оптоволоконных или широкополосных кабелях. По своим характеристикам они являются промежуточными между локальными и глобальными сетями. В последнее время в связи с прокладкой высокоскоростных и надежных оптоволоконных кабелей на городских и междугородних участках, а новые перспективные сетевые протоколы, например, *ATM* (*Asynchronous Transfer Mode* - режим асинхронной передачи), которые в перспективе могут использоваться как в локальных, так и в глобальных сетях.

3. *WAN* – глобальные сети (*Wide Area Networks*). Глобальные сети, в отличие от локальных, как правило, охватывают значительно большие территории и даже большинство регионов земного шара (примером может служить сеть Internet). В настоящее время в качестве передающей среды в глобальных сетях используются аналоговые или цифровые проводные каналы, а также спутниковые каналы связи (обычно для связи между континентами). Ограничения по скорости передачи (до 28,8 Кбит/с на аналоговых каналах и до 64 Кбит/с - на пользовательских участках цифровых каналов) и относительно низкая надежность аналоговых каналов, требующая использования на нижних уровнях протоколов средств обнаружения и исправления ошибок существенно снижают скорость обмена данными в глобальных сетях по сравнению с локальными.

Для объединения *LAN* применяются мосты, маршрутизаторы или шлюзы.

Глобальные компьютерные сети (*GAN*) объединяют *WAN*, компьютерные сети стран, материков. Построение этих сетей выполняется строго в соответствии с международными стандартами [31].

В зависимости от требований, предъявляемых к проектируемой сети, состав оборудования, используемый при монтаже, может варьироваться. На рисунке 1.1 представлена обобщенная схема узла связи (УС) СПД.

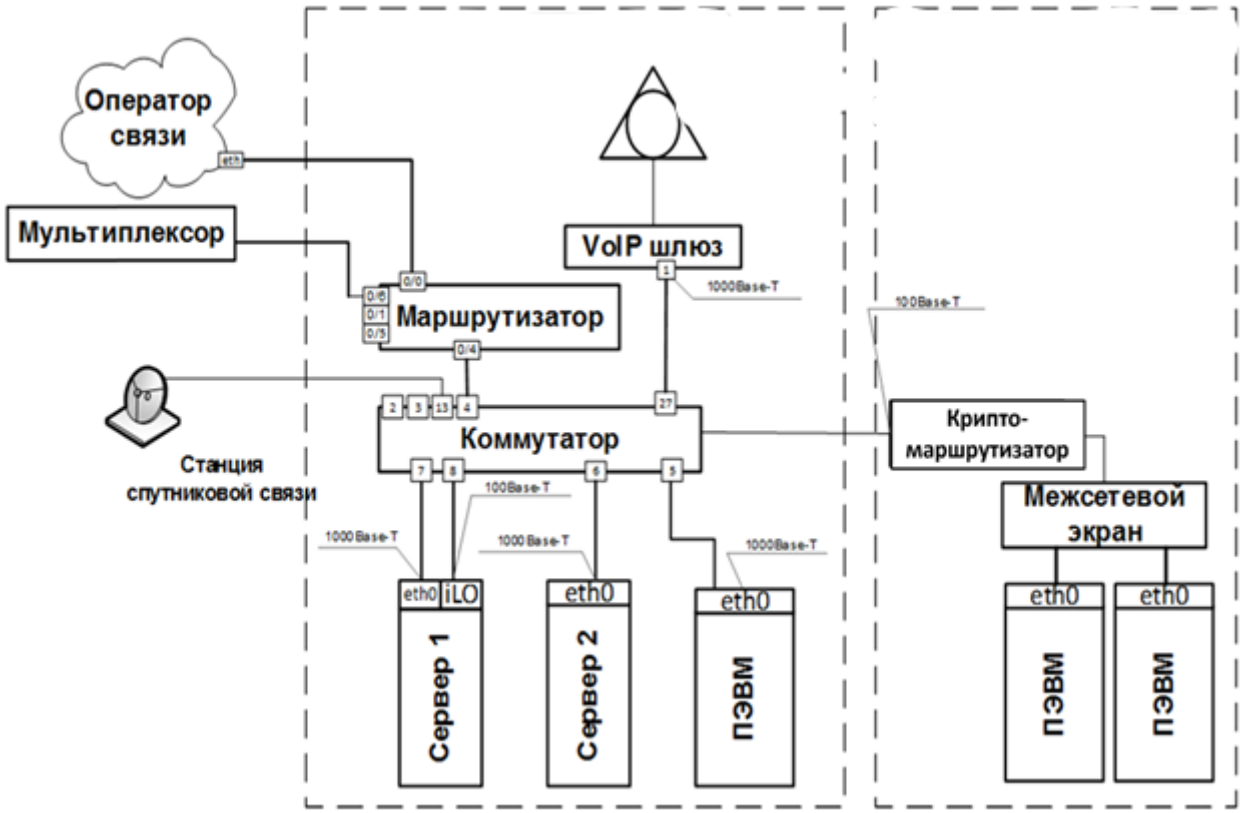

Рисунок 1.1 – Структурная схема узла связи СПД

Открытый сегмент узла связи может состоять из следующих базовых элементов:

маршрутизатор;

коммутатор;

корпоративный сервер (на нем, как правило, работает корпоративный сайт компании);

локальная вычислительная сеть (ЛВС) и автоматизированные рабочие места (АРМ) открытого сегмента;

средства *IP*-телефонии АТС (по технологии *VoIP*), сопрягаемой с внешними ресурсами (городскими и междугородными телефонными сетями).

В состав оборудования закрытого сегмента могут входить:

пограничный маршрутизатор либо межсетевой экран (если необходимо маршрутизировать трафик во внешнюю сеть, например в интернет);

криптомаршрутизатор;

коммутатор ЛВС;

ЛВС и АРМ закрытого сегмента;

средства *IP*-телефонии внутренней закрытой АТС, сопрягаемой с внешними ресурсами, но с обязательным условием криптозащиты и передачи трафика по туннелям виртуальной частной сети (*VPN*).

Таким образом, с точки зрения информационной безопасности криптомаршрутизатор и межсетевой экран или пограничный маршрутизатор обеспечивают криптографическую защиту информации на должном уровне. Однако это не даст стопроцентной защиты от компьютерных атак до тех пор, пока организация не научится минимизировать влияние человеческого фактора. Даже если перевести все узлы связи в режим тотальной автоматизации, исключив из процессов, связанных с обслуживанием СПД, человека, устранить риски, обусловленные доверительными отношениями, внутренними нарушителями, аппаратными или программными закладками и просто банальной неправильной настройкой сетевого оборудования, не представляется возможным.

1.2 Модель угроз ресурсам сети передачи данных

Современный этап развития общества характеризуется повышением роли информационной сферы, представляющей собой совокупность информации, информационной инфраструктуры и субъектов, осуществляющих сбор, формирование, распространение и использование информации. К числу наиболее заметно нарастающих угроз информационной безопасности критически важного объекта инфраструктуры относятся компьютерные атаки (КА) на СПД [32].

Современные направленные КА представляют собой сложное комплексное воздействие на сеть, в результате которого контроль ресурсов, взлом и заражение СПД не происходят «вдруг из ничего». Зачастую этому предшествует долгая и кропотливая работа в киберпространстве: разведка, поиск характерных уязвимостей и захват информационных активов. Меняющийся ландшафт угроз, частота их появления, сложность и целевой

характер атак – все это требует эволюции существующей парадигмы в сфере правил информационной безопасности.

Для минимизации потенциального ущерба важен переход к сочетанию технологий предотвращения, обнаружения и реагирования на КА, однако сегодня анализ инцидента происходит в основном «по факту», путем «латания дыр» в уязвимостях систем обеспечения информационной безопасности. Большинство государственных и частных организаций имеют средства для обнаружения известных атак, хотя, как показывает практика, данные решения не всегда спасают их от злонамеренных сетевых вторжений. Самое сложное в деле защиты конфиденциальных информационных ресурсов – остановить неизвестные атаки, специально созданные с целью обхода имеющейся защиты, использующие изменения сигнатур и шаблонов поведения. Неотъемлемой частью аналитической работы по защите СПД является разработка наиболее полной классификации КА, создающих угрозы для информационных ресурсов.

Многие известные решения по классификации компьютерных атак опираются на определенные требования, такие как:

взаимоисключаемость – категории классификации не должны пересекаться и иметь схожие значения;

исчерпываемость – классификация должна в полной мере раскрывать описываемую область исследования, максимально охватывать характеристики рассматриваемой КА;

понятность – однозначность и краткость представляемой информации;

недвусмысленность – строгое разделение представленных категорий с четко выраженной принадлежностью КА соответствующему классификатору;

полезность – возможность использования в области информационной безопасности обобщенных классификатором данных о той или иной КА;

приемлемость – построение классификации на основе анализа существующей классификации в области исследования [33, 34].

Известные исследования в области классификации КА можно разделить на две крупные группы. Первая охватывает разработку общих классификаций

по компьютерным атакам, основанных на типе атаки или понесенном ущербе. Вторая группа работ изучает определенные компьютерные атаки, давая наиболее глубокое представление о конкретной КА, используемых ею уязвимостях, эксплойтах и, как правило, последствиях вторжения в информационную систему (ИС) [35].

В общих классификациях акцент делается либо на разделение КА по методам проведения атак [36-38], либо на градацию по возможным потерям информационных активов в результате атаки [39-42].

В работе [43] проводится анализ известных классификаций компьютерных атак и представлены классификации по следующим признакам: цель, тип, модель OSI, операционная система (и ее характерные уязвимости), местоположение атакующего, тип ИС, атакуемый сервис, концентрация атаки, наличие обратной связи, условие реализации КА, тип воздействия, автоматизация, источник и количество соединений. Недостатками представленной классификации являются отсутствие определений классификаторов и недостаточное внимание к характеристикам точки воздействия, т.е. к тому, на какие ключевые узлы ИС действует та или иная КА.

В [44] приводится более 20 основных типов кибератак, используемых для внедрения в информационные ресурсы СПД; они разделены на четыре базовые категории. Рассматриваются математические модели для каждой категории, классификация предназначена в первую очередь для инженерного персонала, обеспечивающего системы безопасности современных критически важных объектов инфраструктуры СПД.

В работах [45] авторы руководствуются пятью основными классификаторами: определение природы и сущности КА, цели, воздействия КА на защищаемый информационный ресурс и мероприятиями по предотвращению угроз и их последствий. Отличительной особенностью представляемого классификатора является расширенное, многогранное описание смешанных составных сетевых атак.

Неординарный классификационный подход демонстрируют специалисты Тунисского департамента информатики [46], предложившие гибридную пятимерную модель угроз ИС, которая, во-первых, сочетает в себе классификационные методы, основанные и на технике атак, и на воздействии угроз, и, во-вторых, затрагивает возможные источники реализации угрозы, тип, мотивацию, замысел нарушителя и тип воздействия.

В работе [47] рассматриваются наиболее распространенные КА по обширному набору классификаторов и предлагаются онтологические основы для представления классификации: она отвечает требованиям полезности, взаимоисключения пунктов классификатора, понятности информации и однозначности. Однако работа [47] сосредоточена преимущественно на формализации описания сетевых КА, особенностей их подготовки и проведения.

В [48] предложена классификация методов прогнозирования и профилирования КА, таких как метод анализа логической топологии сети, метод скрытых марковских моделей, методах, основанных на нечеткой логике, методах на базе графиков сетевых атак и статистических методах анализа событий. Несмотря на многочисленный набор классификационных признаков, выделенные категории не рассматривают такие важные аспекты, как уровень эталонной модели взаимодействия открытых систем (ЭМВОС), сетевые устройства, подвергаемые воздействию в первую очередь, а также временные характеристики КА.

Классификация атак, которая может быть использована для идентификации возможных вторжений в специализированные системы мониторинга и управления данными, приведена в [49], так и в работе в [50]. Классификаторы идентифицируют атаки на основные узлы сетевого оборудования, атаки на программное обеспечение и атаки на стеки протоколов связи, выделенные отдельно в специфике *SCADA*-систем, т.е. в области применения диспетчерского контроля, управления и сбора данных.

Статьи [51,52] рассматривают классификационные признаки КА, как параметры для глубокого машинного обучения систем защиты информационных систем, базирующихся на алгоритмах искусственных нейронных сетей.

Таким образом, анализ известных работ в области классификации компьютерных атак позволяет сделать следующие выводы:

при разработке классификации КА необходимо руководствоваться наиболее общими принципами построения классификаций, способствующими правильному подбору оптимальных категорий и критериев для дальнейшей работы с созданной классификацией применительно к различным сферам обеспечения информационной безопасности;

при классификации КА предпочтение следует отдавать более общим, гибридным классификаторам. Классификации, основанные на учете ущерба, полученного информационными активами, и классификации, сосредоточенные на определенных типах атаки, не могут в достаточной мере отразить свойства и характеристики КА;

разрабатываемые классификации должны во всей полноте отвечать на следующие вопросы: «Кто совершает компьютерную атаку?», «Цель компьютерной атаки?», «Источник компьютерной атаки?», «Каков метод распространения атаки и каковы ее последствия для жертвы после инцидента?». Ответы на данные вопросы имеют большое значение для выбора стратегии защиты, предупреждения и прогнозирования атаки, ее своевременного обнаружения и противодействия проникновению в СПД.

Рассмотренные выше подходы не в полной мере отвечают всем этим требованиям.

В таблицах 1.2–1.4 и Б 1 – Б 4 приведены модели угроз безопасности ресурсам СПД [35].

Таблица 1.2 – Модель угроз маршрутизатора

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
Сканирование сетевых сервисов Обнаружение периферийных устройств Обнаружение параметров конфигурации сети	Прослушивание сети	Эксфильтрация через альтернативный протокол Эксфильтрация через канал управления C2 Эксфильтрация через альтернативную сетевую среду Эксфильтрация через альтернативную физическую среду	Внешние удаленные сервисы Метод грубой силы или полный перебор	-	Фишинговый порт Evil double	SSH FTP	Подключение через прокси Собственный криптографический протокол Многократное проксирование Многоуровневое шифрование Средства удаленного доступа	Проброс портов	Скрытие конечного адреса соединения Запасные каналы Многоступенчатые каналы
Оперативная память									
Обнаружение процессов	Hash пароля	Эксфильтрация оперативной памяти	-	Рамкиты	-	-	-	-	-
Операционная система									
Bash History Секретные ключи Обнаружение учетных записей Обнаружение файлов и каталогов Раскрытие парольной политики Обнаружение групп доступа Обнаружение информации о системе	Дампинг учетных данных Захват ввода Автоматизированный сбор	Эксплойты для получения учетных данных Автоматизированная эксфильтрация	Паблик-эксплойты 0-Day	Руткиты	-	TTY	Удаленные сервисы Связь через съемные носители	Создание учетных записей	Скрытые файлы и папки Port Knocking

Разведка			Проникновение			Выполнение		Закрепление	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
Программное обеспечение									
Секретные ключи	Дампинг учетных данных MITM	Эксфильтрация данных из ПО	Паблик-эксплойты 0-Day	Скриптинг Упаковка софта	Выполнение через доверенные утилиты разработчиков софта	-	Удаленное копирование файлов	Внедрение в программный код	Деобфускация/ дешифровка файлов или информации Обфускация файлов или информации
Аппаратная составляющая									
Обнаружение файлов с информацией о системе	Сканирование аппаратных компонентов	Эксфильтрацфия данных о системе	Аппаратные закладки, компрометация цепи поставок	Буткиты	Доверительные отношения	Через аппаратные закладки	Через буткиты	-	Резервный доступ

Таблица 1.3 – Модель угроз коммутатору

Разведка			Проникновение			Выполнение		Закрепление	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
Порты коммутатора									
Обнаружение периферийных устройств	Прослушивание сети	Эксфильтрация через альтернативный протокол Эксфильтрация через канал управления C2 Эксфильтрация через альтернативную сетевую среду Эксфильтрация через альтернативную физическую среду	DOS-атака на канальном уровне	-	-	Распространенные порты	Подключение через прокси Собственный криптографический протокол	-	Подмена MAC-адреса
Аппаратная составляющая									
-	Снифинг ARP-spoofing	-	Аппаратные закладки, компрометация цепи поставок	-	Доверительные отношения	-	-	Подмена MAC-адресов	Резервный доступ

1.2.1 Модель угроз маршрутизатору

Маршрутизатор: уровень ЭМВОС – 3; субэлементы маршрутизатора – порты сетевых служб маршрутизатора, оперативная память, операционная система, программное обеспечение, аппаратная составляющая

Маршрутизатор является одним из самых уязвимых составляющих рассматриваемого объекта: это обусловлено главным образом тем, что он содержит таблицу маршрутизации. Это дает злоумышленнику большое преимущество в плане разведки корпоративной сети изнутри. Захватив маршрутизатор, он может исследовать сеть на предмет выявления уязвимых соседей, чтобы, воспользовавшись их системами, проброситься в сеть глубже.

Существует множество способов для этого (табл. 1.2). Все зависит от конкретной ситуации и типа маршрутизирующего устройства. В большинстве случаев можно подобрать логин и пароль специальными программами или воспользоваться средством *Evil double*.

При отсутствии потенциальных жертв, обменивавшихся данными с маршрутизатором, злоумышленник может наметить новые векторы атаки, направленные на распространение и скрытие следов взлома.

Поскольку маршрутизатор обменивается информацией с другими участниками сети, при его захвате не составит труда реализовать атаку «человек посередине» (*MITM*-атаку), перехватывать трафик или просто отключить любого участника от сети, выполнить проброс портов или вынудить администратора повторно ввести учетные данные с целью их перехвата и применения в другом узле сети.

1.2.2 Модель угроз коммутатору

Коммутатор: уровень ЭМВОС – 2; субэлементы коммутатора – порты коммутатора, аппаратная составляющая

Коммутатор является самым примитивным устройством объекта СПД. В большинстве случаев он не имеет таблицы маршрутизации и работает на канальном уровне. Поэтому в таблице отсутствуют колонки, имеющие отношение

к операционной системе и программному обеспечению. Коммутатор уязвим к *ARP*-атакам, направленным на «прослушивание» сетевого трафика, а также к *DoS*-атакам, нацеленным на канал связи (табл. 1.3). Его можно использовать для получения информации об элементах (пользователях) смежных сети.

1.2.3 Модель угроз серверу

Сервер: уровень ЭМВОС – 3; элемент СПД – сервер *Windows*/*Linux*; субэлементы сервера – порты сетевых служб сервера, блок управления сервера база данных, оперативная память, операционная система.

В подавляющем большинстве случаев на сервере устанавливается операционная система *Linux* или *Windows*. В зависимости от операционной системы, а также от ее дистрибутива, конфигурация сервера, настройки сети и его предназначение могут существенно отличаться. Данная особенность сказывается на выборе варианта атаки. В таблице Б 1 приложения Б рассмотрены угрозы для обеих операционных систем. Первое, на что нужно обратить внимание, это перечень вариантов воздействий с помощью известных эксплойтов, доступных любому обывателю. Однако не следует забывать и о том, что сервер находится внутри сети. Поэтому для того, чтобы его атаковать, существуют три варианта со следующими результатами:

захвачен маршрутизатор, с помощью которого и производятся атаки;

на сервере запущен сайт, который доступен из сети интернет (самый распространенный вариант);

злоумышленник уже захватил сервер, но его права – пользовательские.

1.2.4 Модель угроз криптомаршрутизатору

Криптомаршрутизатор: уровень ЭМВОС – 4; субэлемент криптомаршрутизатора – аппаратная составляющая.

Криптомаршрутизатор имеет собственную операционную систему, недоступную простому обывателю. Из-за того, что о нем никто ничего не знает, описать атаки и выявить уязвимости в его протоколах работы весьма сложно. Слабым местом криптомаршрутизатора может оказаться неправильная настройка

протокола *IPsec*, а это шаг к дальнейшей расшифровке информации при наличии ключей шифрования от другого протокола, работающего в паре с *IPsec*. Кроме того, при отсутствии рекомендуемых настроек правил фильтрации или критических ошибках в настройках фильтров возможна реализация *DoS*-атак, направленных на нарушение работоспособности криптомашрутизатора.

В перечень уязвимостей криптомаршрутизатора также входят доверительные отношения, внутренний нарушитель и аппаратные закладки (табл. Б 2 приложения Б).

1.2.5 Модель угроз межсетевому экрану

Межсетевой экран: уровень ЭМВОС – 5; субэлементы межсетевого экрана – порты сетевых служб межсетевого экрана, оперативная память, операционная система, программное обеспечение, аппаратная составляющая.

Межсетевой экран является одним из первых уязвимых звеньев сети передачи данных, т.к. его работа основана на использовании информации разных уровней модели *tcp*/ip.

На канальном уровне применяются «*управляемые коммутаторы*», которые осуществляют фильтрацию трафика между узлами сети.

На сетевом и транспортном уровнях применяются «пакетные фильтры», которые осуществляют фильтрацию трафика на основе информации, содержащейся в заголовке пакетов.

К межсетевым экранам прикладного уровня, относятся «*файрволы веб-приложений*» (*firewall, FW*) и «*брандмауэры*», которые защищают внутреннюю пользовательскую сеть компании в составе сетей (интернет, интранет) стыка протоколов *TCP*/*IP* от случайного или целевого воздействия из внешней сети. *Firewall* включается в «разрыв» между внешней и внутренней (защищаемой) сетью.

Угрозы межсетевому экрану полностью аналогичны ранее рассматриваемым угрозам серверу (см. табл. Б 3 приложения Б), за тем исключением, что его операционная система, как правило, это *Linux*. Изделия *FW*

имеют различные программно-аппаратные платформы и наборы интерфейсов в зависимости от конкретных целей заказчика и параметров конфигурации сопрягаемого оборудования (маршрутизаторов, коммутаторов, серверов и т.д.).

Сам процесс межсетевого экранирования представляет собой совокупность многофакторных проверок дейтаграмм по их аутентификации в соответствии с критериями безопасности. Однако, помимо этого, *FW* выполняет ряд фискально-контрольных функций: *IP*-фильтрация, *NAT*-обработка, туннелирование, proxy-сервер, регистрация событий. Таким образом, *FW* представляет собой комплексный набор инструментов предупреждения и защиты внутренней локальной вычислительной сети (табл. 1.6).

1.2.6 Модель угроз ПЭВМ

ПЭВМ: уровень ЭМВОС – 6; субэлементы ПЭВМ – порты сетевых служб ПЭВМ, оперативная память, операционная система, программное обеспечение, аппаратная составляющая.

Достигнув оконечного узла сети – ПЭВМ, злоумышленник без труда сможет атаковать пользователя. Угрозы ПЭВМ полностью аналогичны ранее рассматриваемым для сервера (см. табл. 1.4).

В случае, если злоумышленнику не удастся проникнуть в закрытый сегмент сети и, всего лишь, «закрепившись» на АРМ открытого сегмента, он легко может захватить данные пользователей, среди которых, с большой долей вероятности, будут пароли администраторов сети. Не исключено, что эти пароли подойдут для элементов СПД закрытого сегмента (табл. Б 4 приложения Б). Таким образом, в приведенных выше актуальных источниках информации предложены классификации КА, которые могут служить основой для исследования свойств атак и средств противодействия им. Однако известные классификации обобщают атаки в основном для сетей связи общего пользования (ССОП) типа интернета и, как правило, недостаточно информативны для сетей критически важного объекта инфраструктуры, в которых имеются особенности защищенного исполнения, хотя предъявляемые к ним требования жестче, чем к ССОП на несколько порядков.

Представлен новый подход к классификации компьютерных атак на критически важные объекты инфраструктуры, который учитывает десять компонентов цикла угрозы: от комплексного изучения объекта воздействия до реализации атаки. Сформулированы основные тенденции реализации атак на практике – они были положены в основу предложенной классификации.

1.3 Анализ существующих средств защиты сети передачи данных

Комплексная защита СПД от КА является одним из важнейших компонентов защиты сети. **Комплексная защита СПД** - комплекс организационно-технических мероприятий (в т.ч. применение средств), направленных на осуществление защиты информации.

Средство защиты информации – техническое, криптографическое, программное или другое средство, предназначенное для защиты информации, средства в которых они реализованы, а также средства контроля эффективности защиты информации [14].

Мероприятия по защите информации – совокупность действий по разработке и практическому применению мер и средств защиты информации. Общая классификация мер и средств защиты информации представлена на рисунке 1.2.

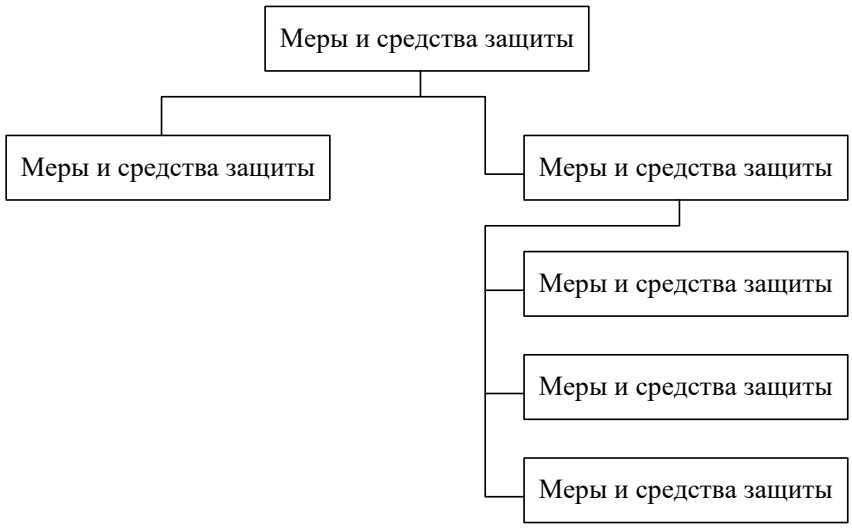

Рисунок 1.2 – Общая классификация мер и средств защиты информации

Организационные (административные) меры – это меры, регламентирующие процессы функционирования СПД, использование ее ресурсов, деятельность персонала, а также порядок взаимодействия пользователей с системой таким образом, чтобы в наибольшей степени затруднить или исключить возможность реализации КА.

Система организационных и организационно-технических мер включает в себя:

– разовые (однократно проводимые и повторяемые только при полном пересмотре принятых решений) мероприятия;

– мероприятия, проводимые при осуществлении или возникновении определенных изменений в самой защищаемой ИТКС ВН или внешней среде (проводимые по необходимости);

– многократно проводимые мероприятия (периодические или стохастические).

К организационным мерам относятся объединения методик и процедур обеспечения безопасности с действиями персонала.

Программно-технические меры защиты ИТКС ВН соответствуют аппаратному, системному и прикладному уровням системы защиты информации. В процессе ее разработки выполняются следующие группы действий (мероприятий).

На аппаратном уровне (защита технических средств (ТС) обработки информации и ТС защиты), представленном на рисунке 1.3, осуществляется:

По направлению защиты от несанкционированного доступа (НСД):

– разработка технологий обработки информации с использованием специализированных серверов с ограниченным набором функций и информационных ресурсов (серверы баз данных, серверы приложений, серверы доступа, web-серверы, почтовые серверы, файл-серверы и т.д.);

– планирование минимизации числа устройств копирования данных на внешние носители (сменных винчестеров, оптических дисков, магнитооптики, принтеров и т.д.) в пользовательских компьютерах;

– разработка схем применения ТС разграничения доступа к элементам ИТКС ВН (смарт-карты, биометрические датчики и т.п.);

– проектирование сетевой инфраструктуры с использованием специализированных устройств разделения доступа и трафика (маршрутизаторы и т.д.);

– планирование использования активных и пассивных средств защиты от побочных электромагнитных излучений (ПЭМИН) (заземление, экранирование, генераторы шума).

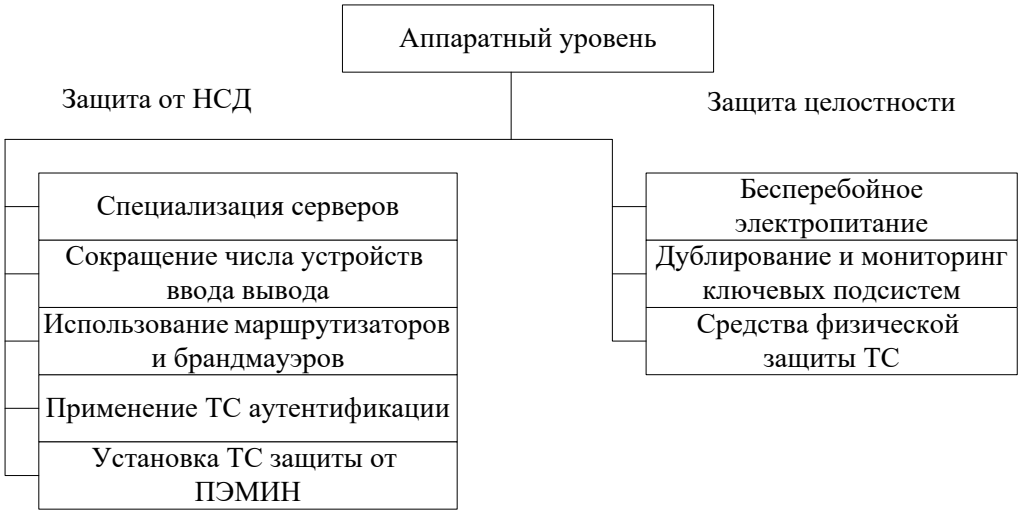

Рисунок 1.3 – Аппаратный уровень системы мер защиты

По направлению защиты целостности:

– проектирование системы бесперебойного электропитания;

– планирование установки дублирующих элементов и датчиков рабочих параметров в ключевых и наиболее уязвимых ТС (магнитные диски, источники питания в серверах, датчики температуры и т.д.);

– проектирование физических средств защиты каналов связи (стальные трубы, специальные кабели, короба и т.п.).

На системном уровне (средства защиты сетевых ОС и систем управления базами данных (СУБД), администраторские надстройки над ними), представленном на рисунке 1.4, осуществляется:

По направлению защиты от НСД:

– планирование политики аутентификации пользователей;

– разработка способов применения штатных средств разделения доступа к информационным ресурсам;

– выбор средств управления локальной вычислительной сетью и надстроек над штатными средствами защиты;

– проектирование технологии работы с базами данных без физического доступа к файлам и без регистрации на сервере баз данных (БД) («клиент-сервер»);

– планирование использования штатных средств СУБД для разделения доступа пользователей к функциям, таблицам, столбцам и т.д.;

– планирование использования шифрации записей в БД и паролей, передаваемых по сети.

Рисунок 1.4 - Системный уровень системы мер защиты

По направлению защиты целостности:

– настройка механизмов восстановления удаленных файлов, защиты файлов от удаления, защиты ядра ОС от разрушения;

– планирование разделения прав воздействия пользователей на БД (владелец, администратор, пользователь);

– настройка механизмов блокировки/отката транзакций и восстановления данных (протоколирование запросов, средства архивации данных);

– планирование регламента использования инструментария тестирования БД и исправления нарушений структуры данных;

– организация технологии информирования администратора ИТКС ВН о возникающих в процессе работы ОС и СУБД ошибках.

На прикладном уровне (инструментальное программного обеспечения (ПО) приложений, пользовательские интерфейсы, индивидуальное защитное ПО), представленном на рисунке 1.5 осуществляется:

По направлению защиты от НСД:

– использование надежной и защищенной подсистемы аутентификации пользователей;

– использование механизма контроля полномочий пользователей по агрегации информации (при сборке отчетов и т.п.);

– применение гибкой многоуровневой системы задания ограничений на доступ к конкретным электронным документам (как единицам хранения информации в КС);

– планирование и разработка разделения полномочий доступа к ветвям и объектам интерфейса и их реквизитам.

По направлению защиты целостности:

– использование программ тестирования БД на отсутствие записей-дублей, на логическую непротиворечивость, на отсутствие ошибок и т.д.;

– применение средств дублирующего протоколирования воздействий пользователей на БД;

– выбор и применение средств авторизации вводимых в БД данных, подготавливаемых электронных документов, отправляемых сообщений (фиксация даты, адреса и пользователя),

– планирование ограничений прав пользователей на удаление и коррекцию

данных в условиях многопользовательского доступа к ИТКС ВН;

– применение средств автоматизированной идентификации и контроля корректности вводимых данных [20].

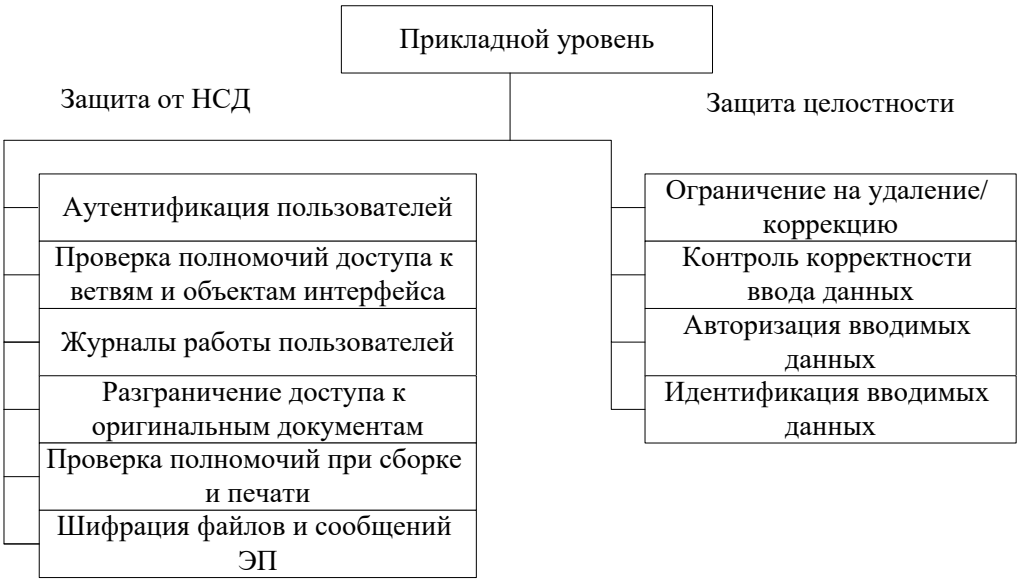

Рисунок 1.5 - Прикладной уровень системы мер защиты

На современном этапе развития теории построения (разработки) СПД вопросам безопасности уделяется недостаточное внимание. Но при этом эти вопросы ставятся в конце цикла разработки, а зачастую уже в процессе функционирования системы.

Таким образом, получается, что безопасность приходится обеспечивать за счет внесения дополнительных подсистем и модулей, которые должны учитывать структуру, возможности и другие характеристики уже сформированной системы и пытаться, каким-либо образом, подстроиться под нее. Такой подход приводит к малоэффективной и кратковременной защите, да к тому же снижает качество функционирования защищаемой системы по основным показателям. Не мало важным аспектом такого подхода являются затраты ресурсов на реализацию и постоянное «латание» защищаемой системы.

Ярким примером такого подхода могут являться системы антивирусной защиты. Применение антивирусных программ позволяет защитить компьютер от конкретного (счетного) набора вирусов на коротком промежутке времени (до создания нового вируса), после чего требуется обновление базы известных

вредоносных программ. Работа антивирусных программ в режиме реального времени отбирает у операционной системы значительный ресурс, а сканирование дискового пространства вообще отдельное деяние, во время которого не рекомендуется производить какие-либо действия (выполнение основных функций). Вопросы, связанные с защитой автоматизированных систем от КА необходимо рассматривать на этапе проектирования и строить систему со структурой, не позволяющей на нее воздействовать (в идеале) или делающей такие воздействия маловероятными и малоэффективными.

Существующие технологии обнаружения атак решают следующие задачи:

распознавание известных атак и предупреждение о них соответствующего персонала;

«понимание» зачастую непонятных источников информации об атаках;

освобождение или снижение нагрузки на персонал, отвечающий за безопасность, от текущих рутинных операций по контролю за пользователями, системами и сетями, являющимися компонентами корпоративной сети;

возможность управления средствами защиты «не экспертами» в области безопасности;

контроль всех действий субъектов корпоративной сети (пользователей, программ, процессов и т.д.).

Очень часто системы обнаружения атак выполнять функции, существенно расширяющие спектр их применения. Например:

контроль эффективности межсетевых экранов;

установка системы обнаружения атак после межсетевого экрана (внутри корпоративной сети) позволяет обнаружить атаки, пропускаемые МСЭ и, тем самым, определить недостающие правила на межсетевом экране;

контроль узлов сети с неустановленными обновлениями или узлов с устаревшим программным обеспечением;

блокирование и контроль доступа к определенным узлам *Internet*;

контроль электронной почты.

Существует большое число различных классификаций систем обнаружения атак, однако самой распространенной является классификация по принципу реализации:

host-based, то есть обнаруживающие атаки, направленные на конкретный узел сети;

network-based, то есть обнаруживающие атаки, направленные на всю сеть или сегмент сети.

Системы обнаружения атак, контролирующие отдельный компьютер, как правило, собирают и анализируют информацию из журналов регистрации операционной системы и различных приложений (*Web*-сервер, СУБД и т.д.). По такому принципу функционирует *RealSecure OS Sensor*. Однако, в последнее время стали получать распространение системы, тесно интегрированные с ядром ОС, тем самым, предоставляя более эффективный способ обнаружения нарушений политики безопасности. Причем такая интеграция может быть реализовано двояко. Во-первых, могут контролироваться все системные вызовы ОС (так работает *Entercept*) или весь входящий/исходящий сетевой трафик (так работает *RealSecure Server Sensor*). В последнем случае система обнаружения атак захватывает весь сетевой трафик напрямую с сетевой карты, минуя операционную систему, что позволяет уменьшить зависимость от нее и тем самым повысить защищенность системы обнаружения атак.

Системы обнаружения атак уровня сети собирают информацию из самой сети, то есть из сетевого трафика. Выполняться эти системы могут на обычных компьютерах (например, *RealSecure Network Sensor*), на специализированных компьютерах или интегрированы в маршрутизаторы или коммутаторы. В первых двух случаях анализируемая информация собирается посредством захвата и анализа пакетов, используя сетевые интерфейсы в беспорядочном (*promiscuous*) режиме. В последнем случае захват трафика осуществляется с шины сетевого оборудования.

Обнаружение атак требует выполнения одного из двух условий - или понимания ожидаемого поведения контролируемого объекта системы или знания

всех возможных атак и их модификаций. В первом случае используется технология обнаружения аномального поведения, а во втором случае - технология обнаружения злоумышленного поведения или злоупотреблений. Вторая технология заключается в описании атаки в виде шаблона или сигнатуры и поиска данного шаблона в контролируемом пространстве (например, сетевом трафике или журнале регистрации). Эта технология очень похожа на обнаружение вирусов (антивирусные системы являются ярким примером системы обнаружения атак), т.е. система может обнаружить все известные атаки, но она мало приспособлена для обнаружения новых, еще неизвестных, атак. Подход, реализованный в таких системах, очень прост и именно на нем основаны практически все предлагаемые сегодня на рынке системы обнаружения атак. Практически все системы обнаружения атак основаны на сигнатурном подходе.

Коммутация позволяет управлять крупномасштабными сетями, как несколькими небольшими сетевыми сегментами. В результате бывает трудно определить наилучшее место для установки системы, обнаруживающей атаки в сетевом трафике. Иногда могут помочь специальные порты (*span ports*) на коммутаторах, но не всегда. Обнаружение атак на уровне конкретного узла обеспечивает более эффективную работу в коммутируемых сетях, так как позволяет разместить системы обнаружения только на тех узлах, на которых это необходимо.

Системы сетевого уровня не требуют, чтобы на каждом хосте устанавливалось программное обеспечение системы обнаружения атак. Поскольку для контроля всей сети число мест, в которых установлены *IDS* невелико, то стоимость их эксплуатации в сети предприятия ниже, чем стоимость эксплуатации систем обнаружения атак на системном уровне. Кроме того, для контроля сетевого сегмента, необходим только один сенсор, независимо от числа узлов в данном сегменте.

Сетевой пакет, будучи ушедшим с компьютера злоумышленника, уже не может быть возвращен назад. Системы, функционирующие на сетевом уровне, используют «живой» трафик при обнаружении атак в реальном масштабе

времени. Таким образом, злоумышленник не может удалить следы своей несанкционированной деятельности. Анализируемые данные включают не только информацию о методе атаки, но и информацию, которая может помочь при идентификации злоумышленника и доказательстве в суде. Поскольку многие хакеры хорошо знакомы с механизмами системной регистрации, они знают, как манипулировать этими файлами для скрытия следов своей деятельности, снижая эффективность систем системного уровня, которым требуется эта информация для того, чтобы обнаружить атаку.

Системы, функционирующие на уровне сети, обнаруживают подозрительные события и атаки по мере того, как они происходят, и поэтому обеспечивают гораздо более быстрое уведомление и реагирование, чем системы, анализирующие журналы регистрации. Например, хакер, инициирующий сетевую атаку типа «отказ в обслуживании» на основе протокола *TCP*, может быть остановлен системой обнаружения атак сетевого уровня, посылающей *TCP*-пакет с установленным флагом *Reset* в заголовке для завершения соединения с атакующим узлом, прежде чем атака вызовет разрушения или повреждения атакуемого узла. Системы анализа журналов регистрации не распознают атаки до момента соответствующей записи в журнал и предпринимают ответные действия уже после того, как была сделана запись. К этому моменту наиболее важные системы или ресурсы уже могут быть скомпрометированы или нарушена работоспособность системы, запускающей систему обнаружения атак на уровне узла. Уведомление в реальном масштабе времени позволяет быстро среагировать в соответствии с предварительно определенными параметрами. Диапазон этих реакций изменяется от разрешения проникновения в режиме наблюдения для того, чтобы собрать информацию об атаке и атакующем, до немедленного завершения атаки.

Системы обнаружения атак, функционирующие на сетевом уровне, не зависят от операционных систем, установленных в корпоративной сети, так как они оперируют сетевым трафиком, которым обмениваются все узлы в корпоративной сети. Системе обнаружения атак все равно, какая ОС

сгенерировала тот или иной пакет, если он в соответствие со стандартами, поддерживаемыми системой обнаружения, но если они общаются между собой по протоколу *IP*, то любая из систем обнаружения атак, поддерживающая этот протокол, сможет обнаруживать атаки, направленные на эти ОС.

В [20] определены основные услуги и механизмы защиты, их размещение по уровням ЭМВОС, а именно:

1. Физический уровень:

а) услуги защиты:

конфиденциальность в режиме с установлением соединения;

конфиденциальность потока трафика.

б) механизмы защиты:

механизм полного шифрования потока данных (защита по ширине частотного спектра).

2. Канальный уровень:

а) услуги защиты:

конфиденциальность в режиме с установлением соединения;

конфиденциальность в режиме без установления соединения.

б) механизмы защиты:

механизм шифрования на основе обычных функций уровня.

3. Физический и транспортный уровень:

а) услуги защиты:

аутентификация равноправного логического объекта;

аутентификация отправителя данных;

услуга управления доступом;

конфиденциальность в режиме с установлением соединения;

конфиденциальность в режиме без установления соединения;

конфиденциальность потока трафика;

целостность данных в режиме с установлением соединения без восстановления;

целостность данных в режиме без установления соединения.

б) механизмы защиты:

аутентификации;

защиты паролями;

подписи;

шифрование;

управление доступом;

управление маршрутизацией;

целостности данных.

4. Сеансовый, представительный и прикладной уровень:

а) услуги защиты:

конфиденциальность в режиме с установлением соединения;

конфиденциальность в режиме без установления соединения;

конфиденциальность выбранных полей;

конфиденциальность потока трафика;

аутентификация равноправного логического объекта;

аутентификация отправителя данных;

целостность в режиме с установлением соединения с восстановлением;

целостность в режиме с установлением соединения без восстановления;

целостность выбранных полей в режиме с установлением соединения;

целостность в режиме без установления соединения;

целостность выбранных полей в режиме без установления соединения;

безотказность с подтверждением отправителя;

безотказность с подтверждением доставки.

б) механизмы защиты:

используются механизмы, которые могут функционировать только в соответствии с правилами кодирования данных на основе синтаксиса передачи, базирующиеся на криптографических методах.

При таком перечне механизмов защиты необходима система управления.

В [53] предложена система обеспечения кибербезопасности (рисунок 1.6), которая включает:

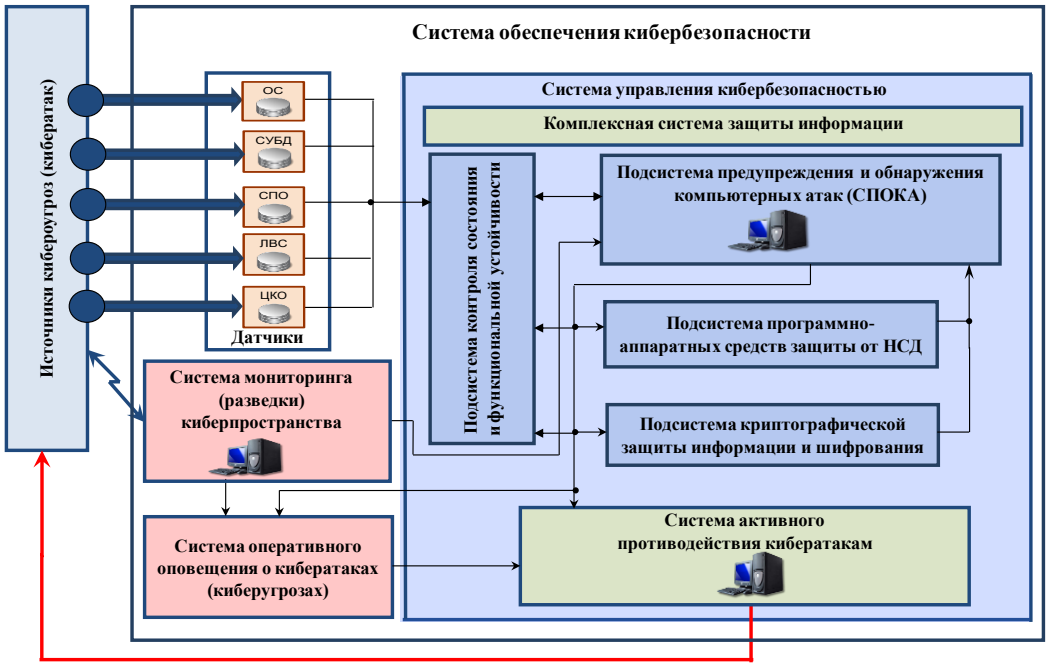

Рисунок 1.6 – Система обеспечения кибербезопасности

1. Систему мониторинга и разведки киберпространства.

Система мониторинга и разведки киберпространства – совокупность специализированных аппаратно-программных средств, предназначенных для:

оценки обстановки в киберпространстве;

систематического сбора и обработки информации о возможных угрозах кибербезопасности СПД (источники, характер, содержание, масштаб и время);

прогнозирования возможных вариантов и технологий реализации КА и потенциально опасных объектов, способных осуществлять КА на СПД;

выявления признаков КА на информационные объекты СПД;

выдачи информации о возможном воздействии КА на информационную инфраструктуру.

На систему мониторинга и разведки киберпространства возлагается функция обеспечение формирования и ведения базы данных по вскрытым (обнаруженным) различным видам и источникам угроз.

2. Систему оперативного оповещения о КА (угрозах).

Система оперативного оповещения о КА (угрозах) – совокупность взаимосвязанных программно-аппаратных и телекоммуникационных средств, предназначенная для организации своевременного доведения информации в

режиме реального времени до соответствующих субъектов управления о возможных (выявленных) КА (угрозах), их сущности и параметрах, попытках несанкционированного доступа (НСД) к информации и принятых (необходимых) мерах защиты и противодействия.

3. Систему управления кибербезопасностью.

Система управления кибербезопасностью включает в себя современные СЗИ и системы (средства) активного противодействия выявленным деструктивным атакам на информационные объекты СПД.

Системы и средства защиты информации СПД представляют собой совокупность технических, программных и программно-технических средств защиты и средств контроля ее эффективности.

В состав системы комплексной защиты информации СПД входят:

подсистема предупреждения и обнаружения компьютерных атак (СПОКА);

подсистема программно-аппаратных средств защиты от НСД;

подсистема криптографической защиты информации и шифрования;

подсистема контроля состояния и функциональной устойчивости.

Подсистема предупреждения и обнаружения компьютерных атак (СПОКА) - совокупность взаимосвязанных программно-аппаратных средств, предназначенных для прогнозирования сценариев и классификации КА, идентификации признаков вторжения и обнаружения КА, анализа уязвимостей и контроля технических и программных средств, информационной системы или сети с целью предупреждения о возможном вторжении, применения методов противодействия КА, оценки и обеспечения функциональной устойчивости функционирования СПД в условиях КА.

4. Подсистему предупреждения и обнаружения компьютерных атак (СПОКА).

Архитектура СПОКА включает:

систему обнаружения вторжений (СОВ) и КА;

подсистему накопления и извлечения накопленных знаний;

подсистему маскировки информационных объектов и критически важных сегментов СПД;

подсистему активного противодействия атакам;

подсистему создания ложных целей;

имитатор атак;

подсистему сбора и фиксации фактов и результатов КА.

5. Подсистему контроля состояния и функциональной устойчивости.

Подсистема контроля состояния и функциональной устойчивости предназначена для обеспечения непрерывного контроля состояния и функциональной устойчивости СПД и ее системы защиты с выдачей информации и рекомендаций в систему управления кибербезопасностью с принятием адекватных мер по корректировке работы СЗИ для борьбы с текущими угрозами и КА, осуществление их своевременной плановой (внеплановой) смены и включает:

систему мониторинга, киберразведки и сбора информации о состоянии функциональной устойчивости и параметрах СПД;

средства анализа и оценки количественных показателей уровня защищенности СПД и ее СЗИ от НСД;

средства подготовки и принятия решений для формирования сигналов управления средств регулирования параметров СЗИ СПД (подсистему адаптации);

средства централизованного перехода к новым настройкам СЗИ.

6. Систему активного противодействия КА.

Система активного противодействия КА включает в себя:

средства выбора оптимальной стратегии противодействия;

средства активного воздействия на процесс совершения атаки;

средства планирования и ведения упреждающих атакующих действий;

средства активного поражения критически важных информационных объектов противоборствующей стороны, в особенности в угрожаемый период и в период ведения боевых действий.

7. Подсистему программно-аппаратных средств защиты от НСД.

В состав подсистемы входят:

средства идентификации и аутентификации пользователей;

средства разграничения доступа пользователей к ресурсам СПД (программно-аппаратный комплекс СЗИ от НСД, включающие в свой состав программные средства защиты информации от НСД);

средства антивирусной защиты (антивирусный комплекс);

защищенные системы управления базами данных;

средства межсетевого экранирования;

средства формирования и проверки электронной подписи;

аппаратно-программные средства доступа к виртуальным системам («защищенное облако»);

средства усиления аутентификации на основе биометрических сканеров, токенов и радиометок;

доверенная операционная система и приложения;

средства защиты информации от утечки из автоматизированных систем (однонаправленные шлюзы);

средства предотвращения утечки информации из автоматизированных систем (*DLP*-системы);

средства контроля защищенности (сканер защищенности);

технические средства охраны средств вычислительной техники, обрабатывающих критически важную информацию;

программно-аппаратные СЗИ технологии «тонкий клиент»;

комплексы средств защиты от иностранной технической разведки и радиоэлектронной борьбы и др.

Таким образом, в настоящее время в нормативно-правовых актах определены механизмы и услуги защиты информации на всех уровнях ЭМВОС. Однако, анализ показывает, что в большинстве случаев выбор местоположения средств защиты основывается только на принципе «минимизации затрат». Данный принцип не обеспечивает выполнение требований по защищенности. Поэтому необходимо определять число, местоположение средств защиты и их взаимосвязь.

1.4 Исследования, посвященные выявлению аномалий в сетях передачи данных и управлению информационной безопасности

Вопросы, связанные с обнаружением аномалий в компьютерных сетях, интересуют специалистов в области компьютерной безопасности достаточно давно. Этот интерес обусловлен тем, что, как правило, воздействие различного типа атак на компьютерные сети приводит к появлению аномалий в сетевом трафике. При этом сами атаки могут быть любой степени сложности. В результате иных эффективных способов обнаружения компьютерных атак, чем детектирование появляющихся после их реализации аномалий, может не существовать. Поэтому многие работы непосредственно посвящены обзору *state-of-the-art* в детектировании аномалий [54-59]. В них предлагаются различные схемы классификации как самих атак, так и методов и средств их обнаружения.

Несмотря на то, что во многих случаях успешно ведут себя статистические методы обнаружения сетевых атак [60], большинство исследователей считают, что наибольшей эффективностью по обнаружению аномалий в современных компьютерных сетях, учитывая сложность, распределенность и интегрированность формируемых на их основе информационных инфраструктур, обладают методы машинного обучения [61,62].

Методы машинного обучения могут существенно повысить эффективность и снизить трудоемкость решения задач кибербезопасности в современных компьютерных сетях [63]. Так, в [64] показано, что наибольшей популярностью среди методов машинного обучения по обнаружению компьютерных атак обладают *SVM* (*Support Vector Machine*) алгоритмы. Эти методы в различных условиях обеспечивают *the accuracy* в диапазоне от 80 до 99.6%. Возможность успешного применения этих же алгоритмов для обнаружения атак в мобильных сетях продемонстрировано в [65-67]. В [68] предложено использовать методы машинного обучения для обнаружения аномалий в СПД, что позволяет выявлять специфические атаки в мобильных устройствах, направленные на систему электропитания. Возможность успешного применения методов машинного обучения для обнаружения атак в сетях Интернета вещей показана в [69].

Особенностью сетей Интернет вещей является использование в них вычислительных средств небольшой мощности. Поэтому основным требованием, предъявляемым к методам машинного обучения, используемым для обнаружения аномалий в сетях Интернета вещей, является низкая ресурсоемкость.

Одним из направлений повышения эффективности обнаружения аномалий с помощью методов машинного обучения является эффективное комбинирование различных методов. В [70,71] показано, что комбинирование методов, таких как *SVM*, деревья решений, наивный байесовский классификатор, многослойный персептрон и другие, приводит к снижению времени обучения классификаторов и времени анализа сетевого трафика, с одной стороны, и точности обнаружения аномалий – с другой.

Другим направлением повышения эффективности обнаружения аномалий является применение методов глубокого обучения, основанных на использовании различных разновидностей искусственных нейронных сетей. Так, [72,73] продемонстрировали эффективность применения комбинированных нейронных сетей для обнаружения в компьютерных сетях аномалий и выявления инцидентов безопасности. В предложенных этими работами нейронных сетях использовался автоэнкодер, позволяющий снизить размерность входных данных. Известны работы, в которых для обнаружения аномалий успешно применяются самоорганизующиеся карты или карты Кохонена [74-76]. После обучения такие сети группируют входные векторы со схожими признаками в отдельные кластеры.

Относительно новым направлением в построении и использовании искусственных нейронных сетей являются рекуррентные нейронные сети, которые способны моделировать элементы памяти [77,78]. Преимуществом таких сетей является их высокая эффективность в задачах предсказания. Одной из разновидностей рекуррентных сетей являются сети с *Long Term Short Memory* (*LTSM*), получившие популярность за счет своей простоты реализации и высокой эффективности. Так, [79] предлагает совместно использовать архитектуру *LSTM* и алгоритмы *SVM,* чтобы получить высокую производительность при обнаружении аномалий в последовательностях данных переменной длины. Использование

автоэнкодера, основного на *LTSM*, исследовано в [80], где показана его высокая эффективность для обнаружения аномалий в непериодических временных рядах. [81] предлагает основанный на *LTSM* детектор аномалий, который не требуется предварительно обучать. [82] продемонстрировала возможность успешного использования *LTSM* для обнаружения аномалий в массивных и многомерных наборах данных.

Также, в настоящее время вопросы, связанные с изучением самоподобных свойств временных рядов и их практическим применением в различных системах мониторинга, находятся в фокусе внимания многих исследователей. Фрактальные свойства исследуются во многих работах [83, 84]. Так, в работе [85] метод *R*/*S*-анализа используется для выявления закономерностей во временных рядах. В работе [86] моделируется *VoIP*-трафик, а также исследуются его фрактальные свойства. В работе [87] изучался не только показатель Херста, но и фрактальная размерность. В работе [88] авторы объясняют, почему телетрафик обладает фрактальными свойствами. Однако эти работы в основном охватывают финансовый сектор и VoIP-телефонию.

При этом следует отметить, что существует мало практических экспериментов, направленных на изучение фрактальных свойств сетевого трафика телекоммуникационных систем. Среди такого рода работ можно выделить работы [89-91]. Однако, в работе [89] трафик рассматривается не в телекоммуникационных сетях, а в радиоволнах, передаваемых сотовыми станциями. Кроме того, исследователи приходят к выводу, что движение самоподобно, часто полагаясь исключительно на визуальные знаки [90,91]. Они ищут похожие участки на графике, выдавая их за самоподобные процессы.

Одной из первых работ, в которой было обращено внимание к свойству самоподобия сетевого трафика, является работа [84]. В ней существенно изменились существующие представления о процессах, происходящих в информационно-телекоммуникационных сетях. Это представление будет более подробно рассмотрено в следующей секции. Также следует выделить ряд работ, в которых предложены математические модели, описывающие посвящены

фрактальных свойства сетевого трафика, например, [92, 93]. Однако они не были ориентированы на обнаружение кибератак, и их нельзя считать исчерпывающими. Таким образом, настоящая работа, с одной стороны, опирается на достигнутые успехи в исследовании самоподобных свойств телекоммуникационного трафика. С другой стороны, она развивает дальше известные решения в направлении создания метода, позволяющего обнаруживать аномалии сетевого трафика, вызванные воздействием кибератак.

В тоже время все угрозы безопасности СПД возможно охарактеризовать двумя параметрами: во-первых, вероятностью реализации угрозы, и, во-вторых, потенциальным ущербом для СПД (организации, предприятия). Использование этих параметров для выбора модели угроз ресурсам СПД позволяет находить «золотую середину» при построении системы защиты, осуществлять выборку методик управления сетями, принимать решения по минимизации рисков. На сегодняшний день существует огромное количество разнообразных и весьма распространенных методик управления системами безопасности СПД, которые в свою очередь делятся на три основные группы [91, 93].

Первая группа [94 – 98] обобщает методики, исходя из количественного критерия. За меру ранжирования моделей угроз (критерий) принимается допустимый уровень возможного ущерба от информационно-технического воздействия на ресурсы СПД и оценка профит-фактора от инвестиций в защитные меры. Количественные методики следуют требованиям *ISO* 27001 и 27002, *NIST*, а также *COBIT IV*. Хотя данные методики учитывают предопределённый риск-аппетит, они не рассматривают вариативность построения системы защиты СПД. Также к недостаткам данных методик можно отнести сложность их исполнения и высокий уровень трудозатрат. Сложность количественных методик также состоит в том, что принимаемое по каждой потенциальной угрозе решение должно быть учтено в стратегии по устранению последствий кибератаки [99]. Например, в работе [100] учитывается количественное ранжирование рисков для СПД. Однако интерес представляет рассматриваемая в этой работе методика управления системой защиты через сервис облачных вычислений. Хотя данная методика,

несомненно, представляет интерес, но она содержит ряд негативных факторов, связанных с проблематикой облачных ресурсов.

Вторая группа методик [101–104] получила общепринятое название качественных методик. Методики этой группы учитывают угрозы безопасности ресурсам СПД по качественному критерию. Качественные методики сводятся к поиску оптимального решения, баланса между затратами на построение системы защиты и получаемым эффектом (*Cost*/*Benefit Analysis*), т.е. качеством системы защиты. Как правило, в методиках используется математический аппарат теории игр (матричные игры). К недостаткам качественных методик можно отнести высокую сложность вычисления результатов анализа рисков для финансового обоснования целесообразности инвестиций в реализацию системы защиты СПД по той или иной модели угроз, а также недостаточную наглядность результатов качественных методик. Методики, использующие качественные критерии, схожи по своей сути с методикой *FRAP* [105, 106].

Третий подход [107-110] является комплексным. Он сочетает в себе подходы, используемые как в первой группе методик, так и во второй. Чаще всего комплексные методики используются в небольших СПД. Слабыми сторонами данной группы методик являются недостаточные аналитические данные по прогнозируемому ущербу модели угроз воздействия кибератак, а также использование минимального набора факторов при оценке риска.

Так, в работах [111 – 114] представлен структурированный подход к оценке модели угроз информационно-телекоммуникационным ресурсам СПД (методики «*CRAMM*», «*MEHARI*»), выполняется интегрированное представление параметра угрозы информационной безопасности, но практически не рассматривается специфика построения системы защиты СПД.

Методика управления системой информационной безопасности *Microsoft Security Assessment Tool* (*MSAT*) [113, 114] интересна не только своей системой ранжирования моделей угроз, но также реализацией системы принятия решения по угрозе информационной безопасности СПД и оценкой эффективности принятых мер. Однако она обычно реализуется на локальных СПД. Система

MSAT базируется на материалах «Руководства по управлению рисками» [113]. Она выполняет следующие функции: 1) оценка рисков; 2) поддержка принятия решений; 3) реализация контроля; 4) оценка эффективности программы. Это приложение ориентировано на компании, имеющие количество сотрудников менее 1000 человек, и разработано для содействия лучшему пониманию потенциальных проблем в сфере информационной безопасности.

1.5 Выводы по первому разделу

1. В первом разделе дана общая характеристика СПД, декомпозирована модель воздействий КА на СПД, разработанная на основе анализа соответствия между основными характеристиками элементов СПД, спецификой их применения и особенностями реализации КА. Модель представляет собой систематизированный перечень КА, воздействующих как на программное обеспечение элементов СПД, так и на субэлементы. Перечень угроз СПД, представленный в виде таблиц, позволяет определить опасность, а также угрозу для каждого элемента сети.

2. Дана классификация КА, которая гармонично дополняет базовую модель информационной безопасности ФСТЭК и может эффективно использоваться при разработке частных моделей информационной безопасности для локальных сетей критически важного объекта инфраструктуры.

Анализ предложенной модели угроз показывает, что КА в большинстве своем универсальны и могут быть на любой элемент СПД, а их воздействия связаны с преднамеренным нарушением программного обеспечения и информации. Наибольшее количество воздействий направлены на серверы, ПЭВМ, а также на межсетевой экран.

3. Проведен общий обзор современных методов обеспечения безопасности СПД, проанализированы исследования, посвященные выявлению аномалий в СПД. Исходя из анализа современных исследований и коммерческих решений, можно сделать вывод о том, что в настоящее время отсутствуют комплексные подходы, направленные на выявление неизвестных компьютерных атак

Так как воздействия КА приводит к появлению в СПД аномальной активности трафика, то для постоянного мониторинга и обнаружения аномальной активности трафика в СПД необходимо учитывать наличие большого количества сетевых маршрутов, на которых периодически возникают резкие колебания задержки в передаче данных и большие потери пакетов, появление новых свойств сетевого трафика, а также необходимость обеспечения высокого качества обслуживания приложений.

Все это послужило поводом для поиска новых методов обнаружения и прогнозирования КА, к числу которых можно отнести и фрактальный анализ.

Таким образом, можно сделать вывод об объективной необходимости разработки соответствующих методов для устранения вышеописанных недостатков.

РАЗДЕЛ 2 АНАЛИТИЧЕСКАЯ МОДЕЛЬ ВЫЯВЛЕНИЯ АНОМАЛИЙ В СЕТЕВОМ ТРАФИКЕ СЕТИ ПЕРЕДАЧИ ДАННЫХ В УСЛОВИЯХ КОМПЬЮТЕРНЫХ АТАК

2.1 Структура модели сетевого трафика сети передачи данных

Для постоянного мониторинга и обнаружения аномальной активности трафика в СПД необходимо учитывать наличие большого количества сетевых маршрутов, на которых периодически возникают резкие колебания задержки в передаче данных и большие потери пакетов, появление новых свойств сетевого трафика, а также необходимость обеспечения высокого качества обслуживания приложений [115]. Именно поэтому на первоначальном этапе важно определиться с моделью, которая будет максимально точно описывать сетевой трафик.

Для создания адекватной модели требуется использование наиболее подходящего математического аппарата. Принимаемая для описания модель должна быть, по возможности, максимально близка к описываемому реальному процессу. Оценить степень близости модели и реального процесса не всегда возможно, поскольку в некоторых случаях реальные процессы попросту недоступны для проведения и наблюдений. В таких случаях приходится полагаться на те логические и иные доводы, которые принимаются при выборе определенной модели и ее параметров.

Трафик как случайный процесс характеризуется параметрами, которые определяют его основные, наиболее важные для моделирования, свойства. Основной задачей модели трафика является описание поступающего потока при помощи набора параметров таким образом, чтобы эти выбранные значения параметров можно было бы применить для нахождения аномалий и вредоносной активности в сети.

Существуют модели, которые описывают сетевой трафик с помощью методов теории вероятностей и математической статистики, а также теории массового обслуживания [115]. Как правило, такие процессы обладают *свойством*

стационарности – вероятностные характеристики (среднее значение и дисперсия) не меняются с течением времени.

Наиболее простой, часто используемой стационарной моделью является модель *простейшего (стационарного пуассоновского) потока*. Основным свойством потока является то, что количество пакетов, поступающих за заданный интервал времени, случайная величина, которая подчиняется распределению Пуассона, а интервалы времени между пакетами случайны и подчиняются экспоненциальному распределению. Модель *простейшего потока* часто применяется для описания трафика, производимого большим количеством независимых источников, например, трафика в сетях с коммутацией каналов.

В сетях с коммутацией пакетов свойства потоков не всегда могут быть описаны распределением Пуассона [115, 116], ввиду нестационарности потока.

Поэтому выделяют нестационарные модели, способные более корректно, описывать сетевой трафик для СПД с коммутацией пакетов. Такие модели основываются на фрактальном анализе и рассматривают сетевой трафик, как самоподобный нестационарный процесс. Под самоподобием понимается свойство сетевого трафика сохранять свой характер при изменении масштаба времени.

Впервые о самоподобном потоке заговорили еще в 1993 году *Leland, Taqqu, Willinger* и *Wilson* проводили исследования *Ethernet*-трафика в сети корпорации *Bellcore* и пришли к выводу, что на больших интервалах он обладает свойством самоподобия, то есть выглядит качественно одинаково при любых масштабах временной оси.

Самоподобие проявляется в том, что имеется медленно убывающая зависимость между величинами трафика в разные моменты времени, а число переданных пакетов имеет сходный вид в различных временных масштабах. Другими словами, самоподобные потоки зависят не только от времени, но и от предыдущих событий.

Постановка задачи исследования: при проектировании системы защиты необходимо учесть все вышеперечисленное и разработать модель выявления аномалий в сетевом трафике сети передачи данных в условиях компьютерных

атак, которая описывает сетевой трафик сразу двух типов: стационарный и нестационарный.

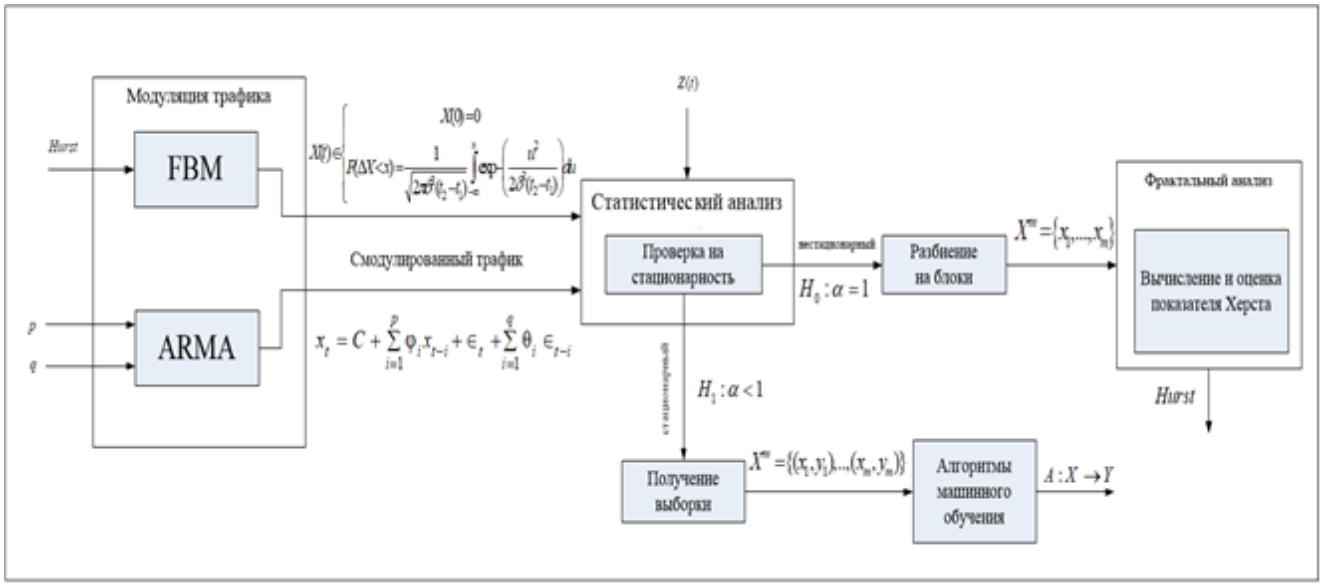

Рисунок 2.1 – Модель выявления аномалий в сетевом трафике СПД

Исходные данные модели:

Hurst – показатель Херста;

p – порядок авторегрессии (зависимость между наблюдениями и число интегрированных наблюдений);

q – порядок скользящего среднего (зависимость между наблюдениями и остатками при применении модели к интегрированным наблюдениям);

Z(*t*) – реальный сетевой трафик;

Y – конечное множество меток класса (аномалия, не аномалия).

Назначение и цель:

Модель выявления аномалий в сетевом трафике сети передачи данных в условиях компьютерных атак предназначена для описания и проверки сетевого трафика на стационарность. После проверки выбирается методика, которая будет производить оценку сетевого трафика на наличие аномалий.

Постановка задачи:

Требуется разработать модель, которая будет максимально точно описывать сетевой трафик в узлах СПД, учитывая, как случайный $X(t)$, так и стационарный

(детерминированный) x_t процесс. Кроме того, модель должна производить проверку на стационарность $H_1 : \alpha < 1$ не только сгенерированного трафика, но и реального $Z(t)$, полученного в ходе эксперимента. Также модель должна принимать решение по выбору алгоритма, с помощью которого будет производиться оценка сетевого трафика на наличие аномальной активности. В случае нестационраности $H_0 : \alpha = 0$ сетевого трафика, оценка производиться на основании вычисления показателя *Hurst*. При $H_1 : \alpha < 1$ с помощью методов машинного обучения, находится целевая зависимость между аномалиями и признаками сетевых фреймов [116-118].

Выходные данные модели:

Hurst - показатель Херста;

$A : X \to Y -$ классификация объектов $x \in X$, где X множество сетевых фреймов.

2.2 Стационарность и нестационарность временного сетевого трафика сети передачи данных

Рассмотрим математическую модель авторегрессии-скользящего среднего *ARMA*:

$$x_t = C + \sum_{i=1}^{p} \varphi_i x_{t-i} + \in_t + \sum_{i=1}^{q} \theta_i \in_{t-i} \tag{2.1}$$

где $\varphi_p, \theta_q \neq 0$ - параметры модели, C - константа, $\in_t$ - белый шум, p - порядок авторегрессии, q - порядок скользящего среднего.

Данная модель используется для анализа и прогнозирования стационарных временных рядов в статистике и обобщает две более простые модели: модель авторегрессии (*AR*) и модель скользящего среднего (*MA*).

Проинтерпретировать модель можно следующим образом: текущее значение зависит от прошлых значений до лага p и от текущего и прошлых внешних шоков до лага q. Запишем авторегрессионный процесс используя лаговый оператор *L*:

$$x_t = C + \sum_{i=1}^{p} \varphi_i L^i x_t + \in_t + \sum_{i=1}^{q} \theta_i L^i \in_t \tag{2.2}$$

Перепишем в виде:

$$\left(1 - \sum_{i=1}^{p} \varphi_i L^i\right) x_t = C + \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \in_t \tag{2.3}$$

Теперь введем два многочлена степени p и q:

$$\varphi(z) = 1 - \sum_{j=1}^{p} \varphi_j z^j = 1 - \varphi_1 z - \varphi_2 z^2 - \ldots - \varphi_p z^p \tag{2.4}$$

$$\theta(z) = 1 + \sum_{j=1}^{q} \theta_j z^j = 1 + \theta_1 z + \theta_2 z^2 + \ldots + \theta_p z^p \tag{2.5}$$

Тогда модель авторегрессии можно формально записать:

$$\varphi(L) x_t = C + \theta(L) \in_t \tag{2.6}$$

где $\varphi(L)x$ - авторегрессионная часть многочлена, а $\theta(L) \in_t$ - часть скользящего среднего.

$$\varphi(z) = 1 - \varphi_1 z - \ldots - \varphi_p z^p \tag{2.7}$$

Временной ряд является стационарным если все корни авторегрессионного многочлена (2.7) лежат вне единичного круга комплексной плоскости $|z_j| > 1$ (то есть по модулю строго больше 1).

Если имеются корни, равные по модулю единице $|z_j| = 1$, то авторегрессионный процесс является нестационарным.

Рассмотрим временной ряд, описываемый моделью *ARMA* при $p = 1$ (рис. 2.2):

$$x_t = C + \varphi x_{t-1} + \in_t + \sum_{i=1}^{q} \theta_i \in_{t-i} \tag{2.8}$$

где $\varphi, \theta_q \neq 0$.

Тогда $\varphi(z) = 1 - \varphi z$ и его корень $z_0 = \frac{1}{\varphi}$.

Так как $|z_0|>1 \Leftrightarrow |\varphi|<1$, то это и будет условием стационарности для этого ряда.

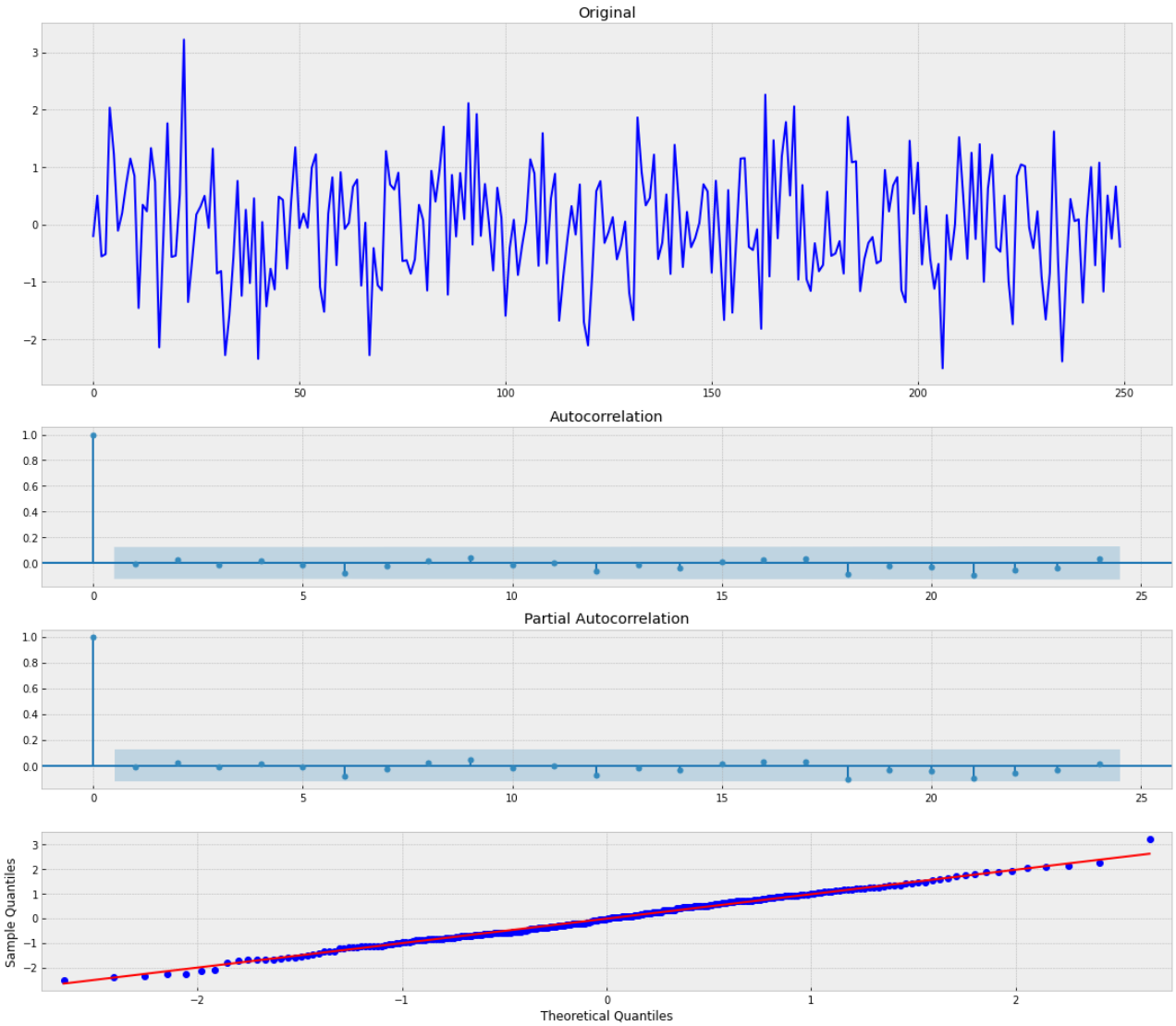

Рисунок 2.2 – Стационарный временной ряд

Кроме того, у стационарного процесса среднее постоянно по времени $Ex_t \equiv \text{const}$ т.е временной ряд не имеет тренда, а ковариация между различными элементами временного ряда зависит только от того, на сколько сильно они отдалены друг от друга по времени. Другими словами, ковариация зависит только от лага h $\text{cov}(x_t, x_{t+h}) = \gamma(h)$. Величина h, характеризующая разницу во времени между элементами временного ряда, называется лаговой переменной или

запаздыванием. Так как $\gamma(0) = \mathrm{cov}(x_t, x_t) = \mathrm{Var}(x_t)$, то дисперсия стационарного временного ряда также не меняется со временем.

2.3 Фрактальные свойства нестационарного временного сетевого трафика сети передачи данных

Одна из наиболее корректных моделей, описывающих поведение нестационарного сетевого трафика – временного ряд, описанный через фрактальное броуновское движение.

Процесс $X(t)$ называется фрактальным броуновским движением с параметром H, $0 \le H \le 1$, если приращения случайного процесса $\Delta X(\tau) = X(t+\tau) - X(t)$ имеют гауссовское распределение:

$$P(\Delta X < x) = \frac{1}{\sqrt{2\pi}\delta_0 \tau^H} \int_{-\infty}^{x} \exp\left[-\frac{z^2}{2\delta_0^2 \tau^{2H}} \right] dz \qquad (2.9)$$

где δ_0— коэффициент диффузии.

Процесс нестационарен, если эти условия нарушаются (рис. 2.3).

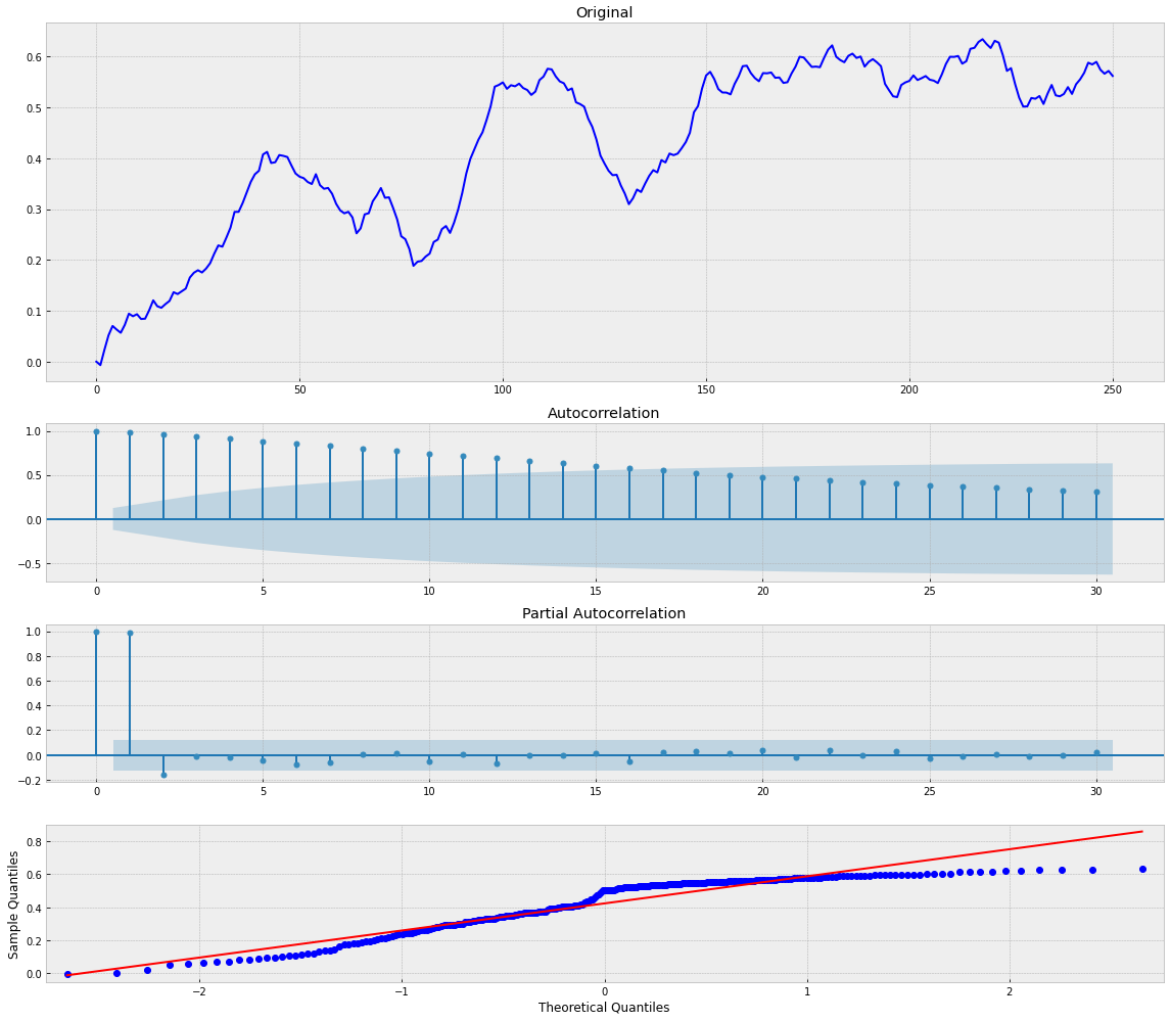

Рисунок 2.3 – Нестационарный временной ряд

2.4 Проверка на стационарность сетевого трафика сети передачи данных

Для проверки гипотезы о стационарности ряда используется расширенный тест Дики-Фуллера.

При помощи этого теста проверяют значение коэффициента авторегрессии α в авторегрессионном уравнении *AR*. Рассмотрим авторегрессионное уравнение первого порядка *AR*(1):

$$y_t = \alpha \cdot y_{t-1} + \varepsilon_t \tag{2.10}$$

где y_t-временной ряд, а ε - белый шум, $t = 1,...,T$.

1. Если $H_1 : \alpha < 1$, то ряд y_t будет стационарным, $y_t \sim I(0)$ и OLS-оценка $\hat{\alpha}$ будет иметь нормальное распределение с нулевым средним и дисперсией $1-\alpha^2$

.

Для тестирования гипотезы единичного корня строится OLS-оценка $\hat{\alpha}$:

$$\hat{\alpha} = \frac{\sum_{t=1}^{T} y_{t-1} y_t}{\sum_{t=1}^{T} y_{t-1}^2} \tag{2.11}$$

И соответствующая ей *t*-статистика

$$t_\alpha = \frac{\hat{\alpha} - 1}{S / \sqrt{\sum_{t=1}^{T} y_{t-1}^2}} \tag{2.12}$$

где $S^2 = T^{-1} \sum_{t=1}^{T} (y_t - \hat{\alpha} y_{t-1})^2$ - оцененная дисперсия остатков.

Если $t_\alpha < t_{табл}^{5\%}$ - временной ряд стационарен на уровне значимости 5%.

2. Если $H_0 : \alpha = 1$, то распределение этой оценки больше не будет нормальным, и процесс y_t будет нестационарным с зависящей от времени дисперсией $y_t \sim I(1)$. В этом случае для моделирования динамики такого ряда необходимо использовать его первую разность $\Delta y_t = y_y - y_{t-1}$. При нулевой гипотезе статистика нормализованного смещения $T(\hat{\alpha} - 1)$ и t-статистика t_α имеют нестандартные предельные распределения Дики-Фуллера:

$$T(\hat{\alpha} - 1) \Rightarrow \frac{\int_0^1 W(r) dW(r)}{\int_0^1 W^2(r) dr} \tag{2.13}$$

$$t_\alpha \Rightarrow \frac{\int_0^1 W(r) dW(r)}{\sqrt{\int_0^1 W^2(r) dr}} \tag{2.14}$$

где $W(r)$ - стандартный Винеровский процесс (Броуновское движение).

Если $t_\alpha > t_{табл}^{5\%}$ - временной ряд нестационарен на уровне значимости 5%.

2.5 Вычисление и оценка показателя Херста с помощью R/S

Для расчета показателя Херста в нестационарном трафике на малых выборках используется *R*/*S* анализ. Многие исследователи [116-118] применяют *R*/*S* анализ для нахождения показателя Херста в сетевом трафике. Одно из основных преимуществ *R*/*S*-анализа заключается в том, что в отличие от многих широко распространенных статистических критериев, он не основан на каких бы то ни было предположениях об организации исходных данных (о том, какому закону распределения они подчиняются). Очень быстрый и легко реализуемый.

Алгоритм *R*/*S*:

$$\frac{R}{S} = \left(aN\right)^H \tag{2.15}$$

откуда

$$H = \frac{\log(R/S)}{\log(aN)} \tag{2.16}$$

где H – показатель Херста;

S – среднеквадратичное отклонение ряда наблюдений x;

N – число периодов наблюдений;

a - заданная константа, положительное число.

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{2.17}$$

$\overline{x}$ - среднее арифметическое ряда наблюдений x за N периодов

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{2.18}$$

Размах накопленного отклонения R это разность между максимальным и минимальным накопленными отклонениями:

$$R = \max_{1\le u\le N} Z_u - \min_{1\le u\le N} Z_u \tag{2.19}$$

Где Z_u - накопленное отклонение ряда x от среднего $\overline{x}$:

$$Z_u = \sum_{i=1}^{u}(x_i - \overline{x}) \tag{2.20}$$

Из формулы видно, что на рост показателя Херста влияют:

увеличение размаха колебаний R;

уменьшение среднеквадратичного отклонения S;

уменьшение количества наблюдений N.

При $0{,}5 \leq H \leq 1{,}0$ мы наблюдаем персистентные, или трендоустойчивые ряды. Если ряд возрастает (убывает) в предыдущий период, то вероятно, что он будет сохранять эту тенденцию еще какое-то время в будущем. Наблюдения не являются независимыми. Каждое наблюдение несет память обо всех предшествующих событиях. Процесс обладает длительной памятью. Эта память долговременная, теоретически она сохраняется навсегда. Трендоустойчивость поведения, или сила персистентности, увеличивается при приближении H к единице. Обычно тот факт, что $0.5 < H < 1$, считается достаточным основанием для признания процесса самоподобным.

При $H = 0.5$ ряд является случайным (последующие значения временного ряда не связаны с его предыдущими значениями).

При $0 < H < 0.5$ ряд является антиперсистентным (последующие изменения значений временного ряда противоположны его предыдущему поведению.

Для проверки R/S сформирован датасет состоящий из легитимного (рис 2.4.) и аномального (рис 2.6.) сетевых трафиков.

Рисунок 2.4 – Легитимный сетевой трафик

После применения R/S анализа для легитимного трафика, построена логарифмическая регрессия (рис. 2.5) и вычислен показатель Херста равный 0.56.

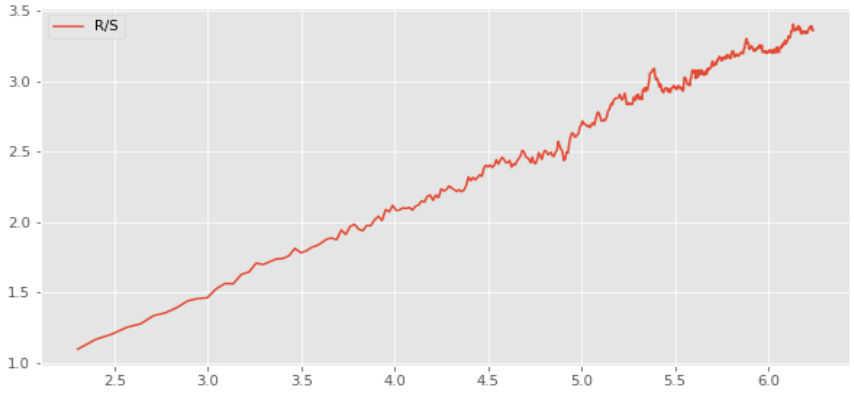

Рисунок 2.5 – Зависимость *R/S* от времени в логарифмической шкале (Херста = 0.56)

Далее *R/S* применялся к аномальному сетевому трафику (рис. 2.6).

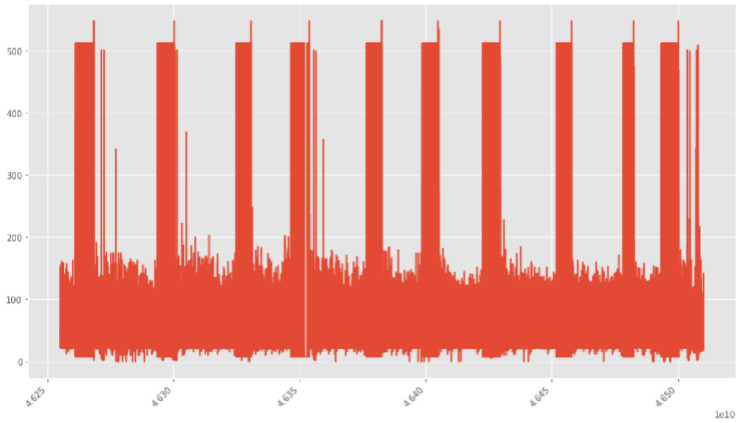

Рисунок 2.6 – Аномальный сетевой трафик

Применив *R/S* анализ для аномального трафика, построена логарифмическая регрессия (рис. 2.7) и вычислен показатель Херста равный 1.378.

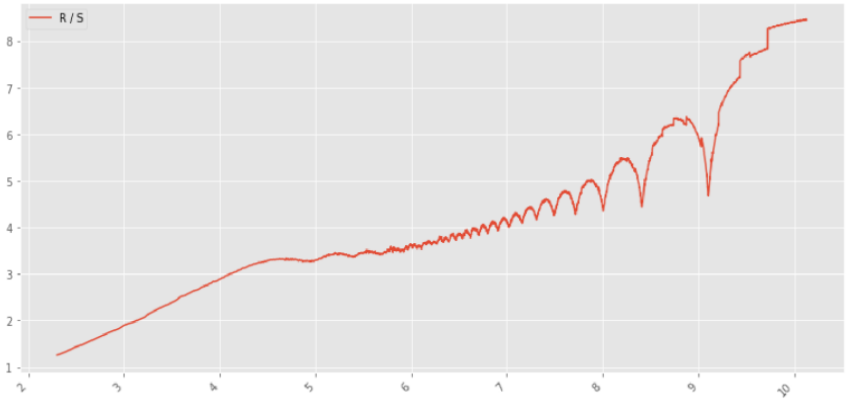

Рисунок 2.7 – Зависимость *R/S* от времени в логарифмической шкале (Херста = 1.378)

Как видно из рисунка 2.7 показатель Херста превышает максимальное

константное значение 1, что подтверждает наличие аномалий в сетевом трафике.

2.6 Вычисление и оценка показателя Херста с помощью DFA

Для расчета показателя Херста в нестационарном трафике на зашумленных и больших объемах данных, для более точных вычислений, предпочтительнее использовать *DFA* анализ:

1. Преобразование временного ряда $x(t)$ в функцию кумулятивных сумм (профиль функции) путем суммирования значений временного ряда:

$$X(t) = \sum_{i=1}^{N} \left(x_i(t) - \bar{x} \right) \tag{2.20}$$

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i(t) \tag{2.21}$$

2. Временной ряд, разбивается на $\frac{N}{n}$ непересекающихся интервалов.

3. В пределах каждого интервала осуществляется линейная аппроксимация ряда $x(t)$ методом наименьших квадратов – выделяется локальный тренд: $y_j(t) = a_j t + b_j$, где a_j и b_j - константы для каждого интервала.

4. Для каждого интервала устраняется локальный тренд путем перехода к разности $X(t) - y_j(t)$ и проводится анализ среднеквадратичного отклонения от локального тренда, т.е. вычисляется функция:

$$F_j^2(n) = \frac{1}{n} \sum_{t=jn+1}^{(j+1)n} \left(X(t) - y_j(t) \right)^2 \tag{2.22}$$

5. Далее вычисляется среднее значение:

$$F^2(n) = \frac{n}{N} \sum_{j=0}^{\frac{N}{n}-1} F_j^2(n) \tag{2.23}$$

Если исследуемый ряд сводится к самоподобному множеству, проявляющему дальнодействующие корреляции, то флуктуационная функция $F(n)$ представляется степенной зависимостью:

$$F(n) \sim n^H \tag{2.24}$$

где H – показатель Херста. H может быть вычислен с помощью метода наименьших квадратов как угловой коэффициент прямой, определяющей зависимости $\log F\ (n)$ от $\log(n)$.

С помощью алгоритма *DFA* проведен расчет показателя Херста нестационарного трафика, описываемого моделью с заранее заданным показателем Херста = 0.5, что соответствует зашумленному временному ряду (рис 2.8).

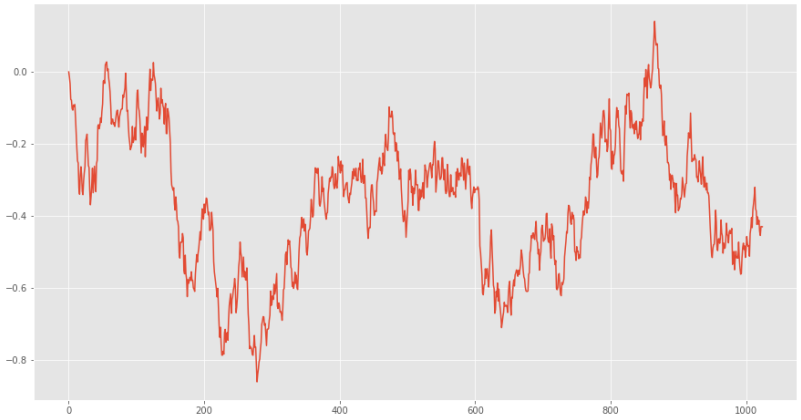

Рисунок 2.8 – Фрактальное броуновское движение при H = 0.5

С помощью *DFA*, построена логарифмическая регрессия и найден показатель Херста, равный 0.49 (рис. 2.9). Полученный результат почти полностью совпадает с заданной величиной, что подтверждает эффективность текущего метода на зашумленных рядах.

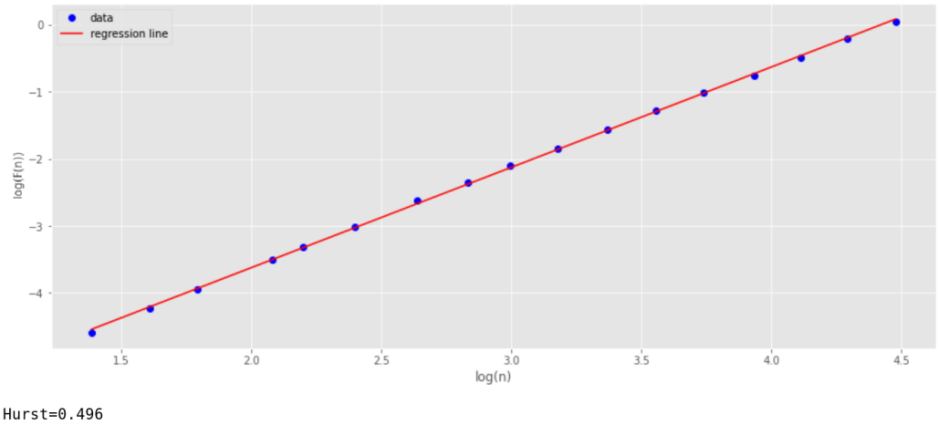

Рисунок 2.9 – Зависимость $F(n)$ от времени в логарифмической шкале (Херст = 0.496)

Анализ показал, что метод *DFA* исключает линейный тренд из каждого анализируемого фрагмента временного ряда, что позволяет повысить точность в условиях низкочастотных помех или на больших объемах данных. В тоже время *R*/*S* является более быстрым алгоритмом вычисления показателя Херста, не уступающим в точности на небольших объемах данных. Поэтому *R*/*S* является предпочтительным для дальнейшего исследования [118-120].

Таким образом, эксперименты, проведенные на эталонных выборках, состоящие из легитимного и аномального трафика, продемонстрировали наличие самоподобия трафика КС и возможность достаточно точного определения показателя самоподобия на основе рассмотренных алгоритмов.

2.7 Выводы по второму разделу

1. Во втором разделе проведен анализ существующих подходов к проектированию моделей описывающих сетевой трафик СПД и алгоритмов по оценке свойств самоподобия в нестационарном сетевом трафике в СПД.

2. Разработана модель выявления аномалий в сетевом трафике сети передачи данных в условиях компьютерных атак, отличающаяся от известных возможностью описывать стационарный и нестационарный сетевой трафик, классифицировать его и в зависимости от типа трафика обосновать метод по выявлению аномалий.

Модель основана на использовании основных положений теории фракталов и предлагаемых этой теорией методов оценки самоподобия, *R*/*S* -анализ и метод *DFA*. При тестировании фрактальных методов, позволяющих проводить исследования долговременных зависимостей в трафике КС, метод *DFA* оказался более эффективен, чем *R*/*S* -анализ на зашумленных данных или больших выборках, из-за исключения линейного тренда из каждого анализируемого фрагмента временного ряда. Следовательно, *DFA* позволяет обнаруживать корреляции на большие расстояния, встроенные в нестационарные ряды, что характерно для КС, избегая ложного обнаружения явных корреляций на большие расстояния, которые являются артефактами нестационарности. Неоспоримым

преимуществом *R*/*S* -анализа являются более быстрые вычисления показателя Херста, а эффективность алгоритма не уступает в точности *DFA*, на небольших объемах данных.

3. Поведены эксперименты, которые показали, что существует характерное время, после которого показатель Херста резко меняется. Это время указывает на объем системной памяти. Экспериментальные результаты также свидетельствуют о том, что самоподобные свойства присущи любому сетевому трафику на канальном уровне модели *tcp*/*ip*. При появлении сетевых аномалий, вызванных, например, кибератаками типа *DDoS* и «сканирование сети и ее уязвимостей», характер этих свойств начинает существенно отличаться от нормального трафика. Основываясь на результатах эксперимента, можно сделать вывод, что предложенная модель является достаточно адекватной.

4. Реализованы программные модули для определения стационарности сетевого трафика с помощью расширенного теста Дики-Фуллера, а также вычисления и оценки показателя Херста с помощью *R*/*S* и *DFA*.

РАЗДЕЛ 3. МЕТОДИКА РАННЕГО ОБНАРУЖЕНИЯ АНОМАЛИЙ В СЕТЕВОМ ТРАФИКЕ СЕТИ ПЕРЕДАЧИ ДАННЫХ

С целью повышения полноты и точности выявления аномалий в сетевом трафике в условиях компьютерных атак разработана методика (рис. 3.1), позволяющая обнаруживать КА на раннем этапе их проявления с помощью методов машинного обучения для стационарного сетевого трафика СПД и фрактального анализа для нестационарного [121-129].

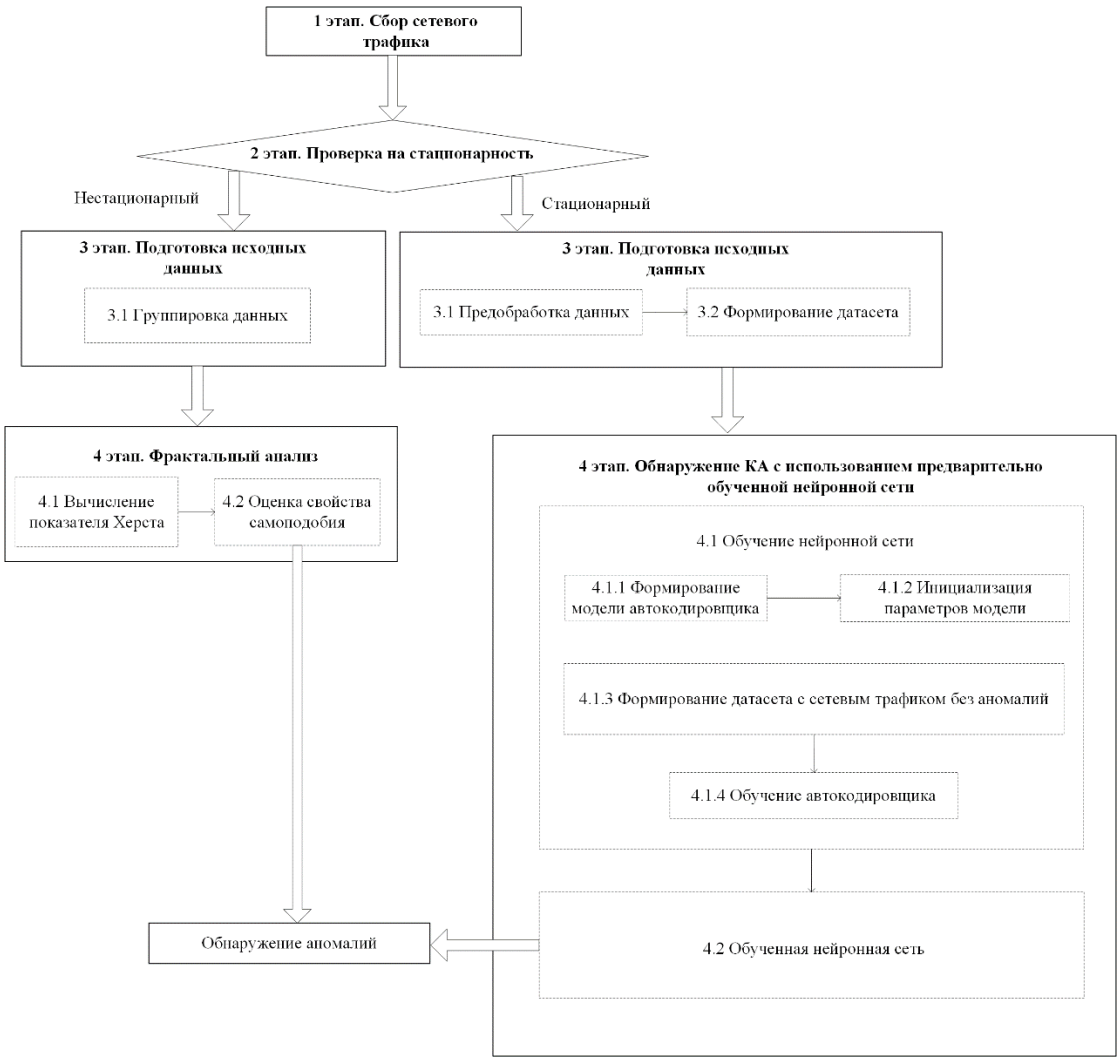

Рисунок 3.1 – Блок-схема методики раннего обнаружения КА в сетевом трафике СПД

Методика состоит из следующих этапов: сбор сетевого трафика в СПД; осуществляется его проверка на стационарность; подготовка исходных данных; фрактальный анализ для нестационарного сетевого трафика в СПД; машинное обучение для стационарного сетевого трафика в СПД.

3.1 Обнаружение аномалий в нестационарном сетевом трафике сети передачи данных с помощью фрактального анализа

Для выявления аномалий в нестационарном сетевом трафике СПД, применяется фрактальный анализ, который главным образом базируется на вычислении и оценки показателя Херста. С целью оценки аномальности предложено использовать не полезное содержимое пакетов, а взять за основу

предложение Унтерова Д.С. [130] о том, что информации из заголовков пакетов будет достаточно. В качестве такой информации будем использовать количественные значения передаваемых флагов (меток, указывающих на тип пакета).

На рисунке 3.2 представлена блок-схема методики обнаружения аномалий в нестационарном сетевом трафике СПД.

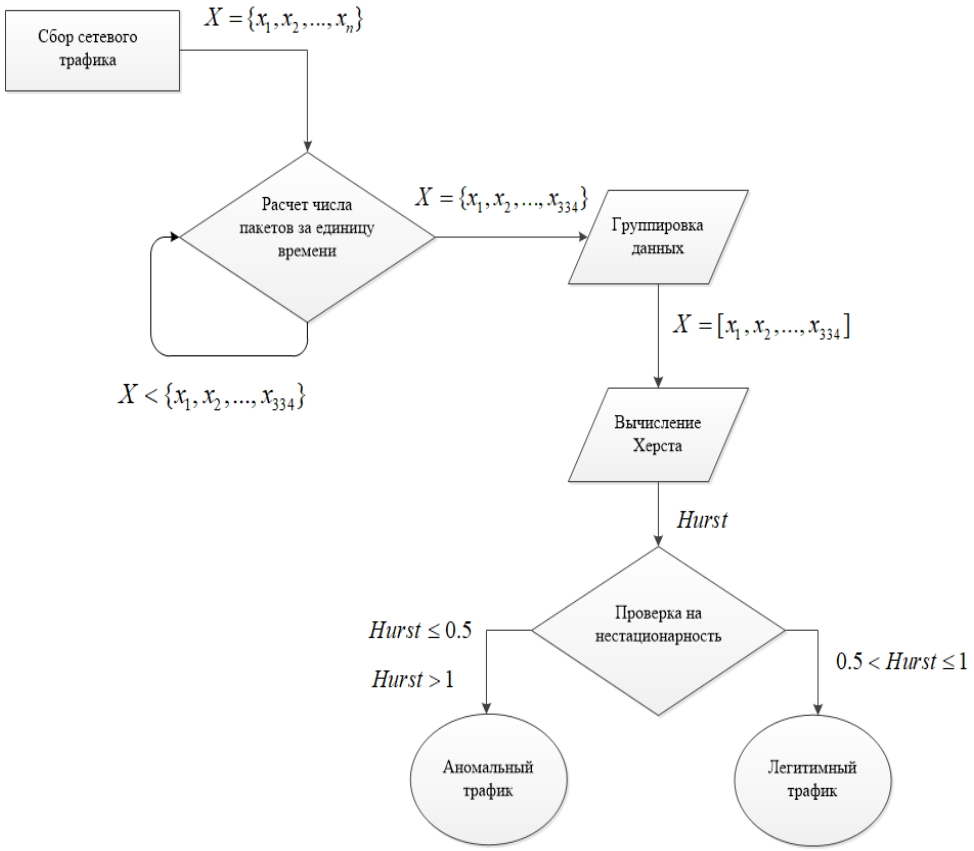

Рисунок 3.2 – Блок-схема методики обнаружения аномалий в нестационарном сетевом трафике СПД

Анализ [131-137] также показывает, что для выявления аномального поведения в трафике, достаточно анализировать его основные параметры и нет необходимости изучать содержимое каждого пакета. Примерами аномалий, обнаруженных на основе анализа телеметрии трафика, является внезапное увеличение интенсивности трафика от рабочей станции или изменение структуры в сравнении с обычными ежедневными показателями для данной сети устройства.

Сетевой поток делится на группы и рассчитывается показатель Херста для каждой из групп. Сетевые пакеты помечаются аномальными в том случае, когда

нарушается свойство самоподобия в исследуемой группе. Для проверки эффективности предложенного подхода, сперва он был протестирован на легитимном сетевом трафике (рис. 3.3).

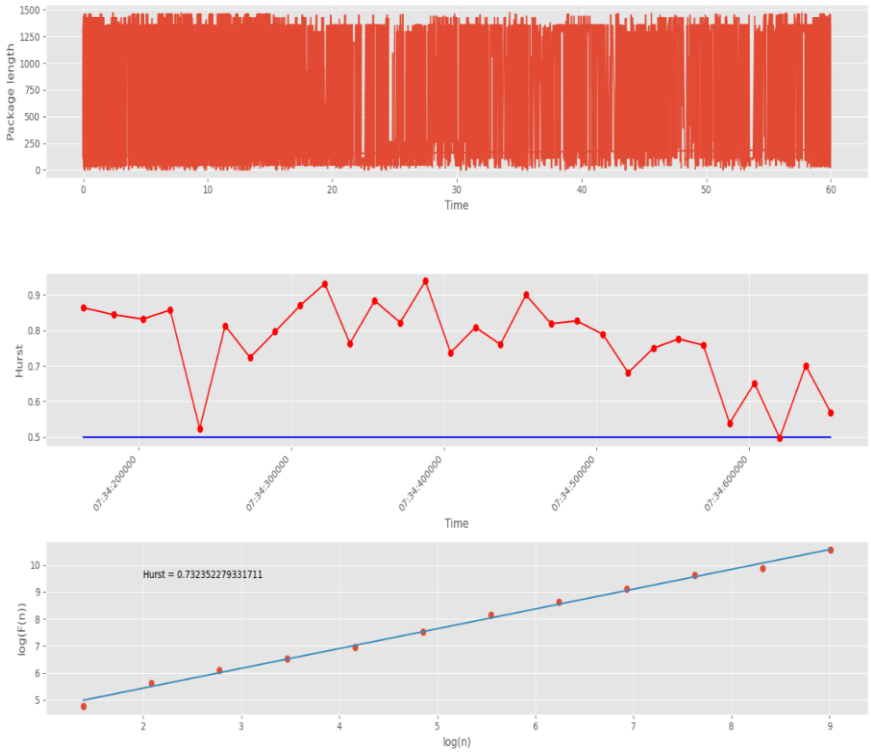

Рисунок 3.3 – Вычисление *H* методом фрактального анализа легитимного *UDP* трафика. Разбиение 10000 точек на 19 групп

Синей прямой линией обозначен порог, соответствующий границе белого шума (*Hurst* = 0,5). Точки на втором графике соответствуют номерам групп пакетов (всего 30 точек). Точки на третьем соответствуют *scales* (всего 12 точек). Количество *scales* влияет на точность и на длительность работы алгоритма. Чем больше количество scales, тем выше точность, и, наоборот, меньше длительность его работы.

Как видно из рис. 3.3, мера фрактальности для всех групп пакетов полностью лежит выше отметки 0.5. Это указывает на наличие самоподобных свойств у каждой группы. Кроме того, на третьем графике (логарифмической регрессии) отражен параметр Херста для всего *DataFrame*, который подтверждает наличие фрактальных свойств и повторяющихся процессов. Далее проводилось тестирование аномального сетевого трафика, полученного во время проведения

DoS атаки и компьютерно-технической разведки. При этом преследовалась цель подбора максимального числа групп разбиения, при котором оцениваемый параметр *H* будет вычисляться с высокой точностью.

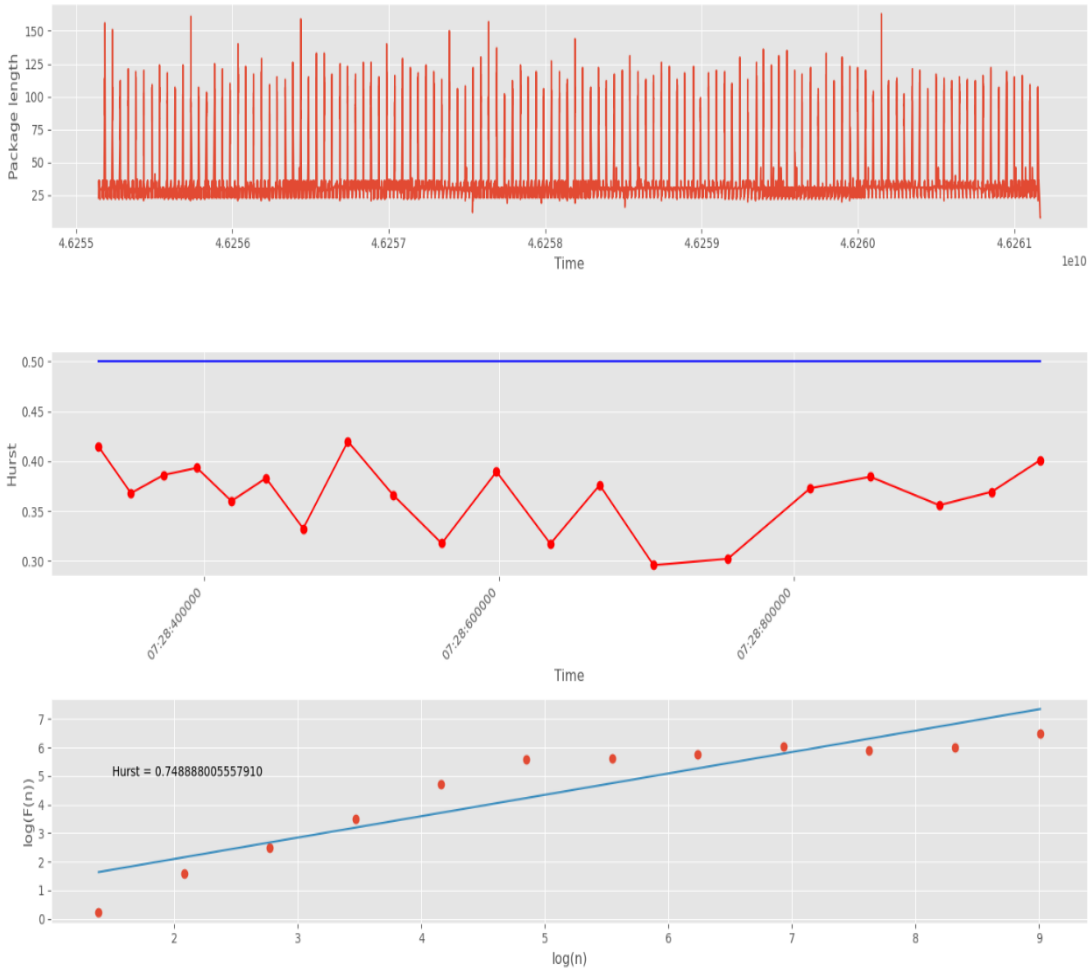

Рисунок 3.4 – Вычисление *H* методом фрактального анализа аномального *UDP* трафика. Разбиение 10000 точек на 19 групп

Из рисунка 3.4 видно, что свойство самоподобия нарушается, т.к. показатель Херста, на каждом из интервалов, меньше порогового значения 0.5. Это свидетельствует о нарушении фрактальной структуры трафика и наличии в нем аномалий. Следовательно, предлагаемый подход способен обнаруживать аномалии в интервалах (группах), состоящих из 526 сетевых пакетов.

Увеличив количество групп до 29, сокращается временной интервал без потери точности обнаружения аномальной активности в сети (рис. 3.5).

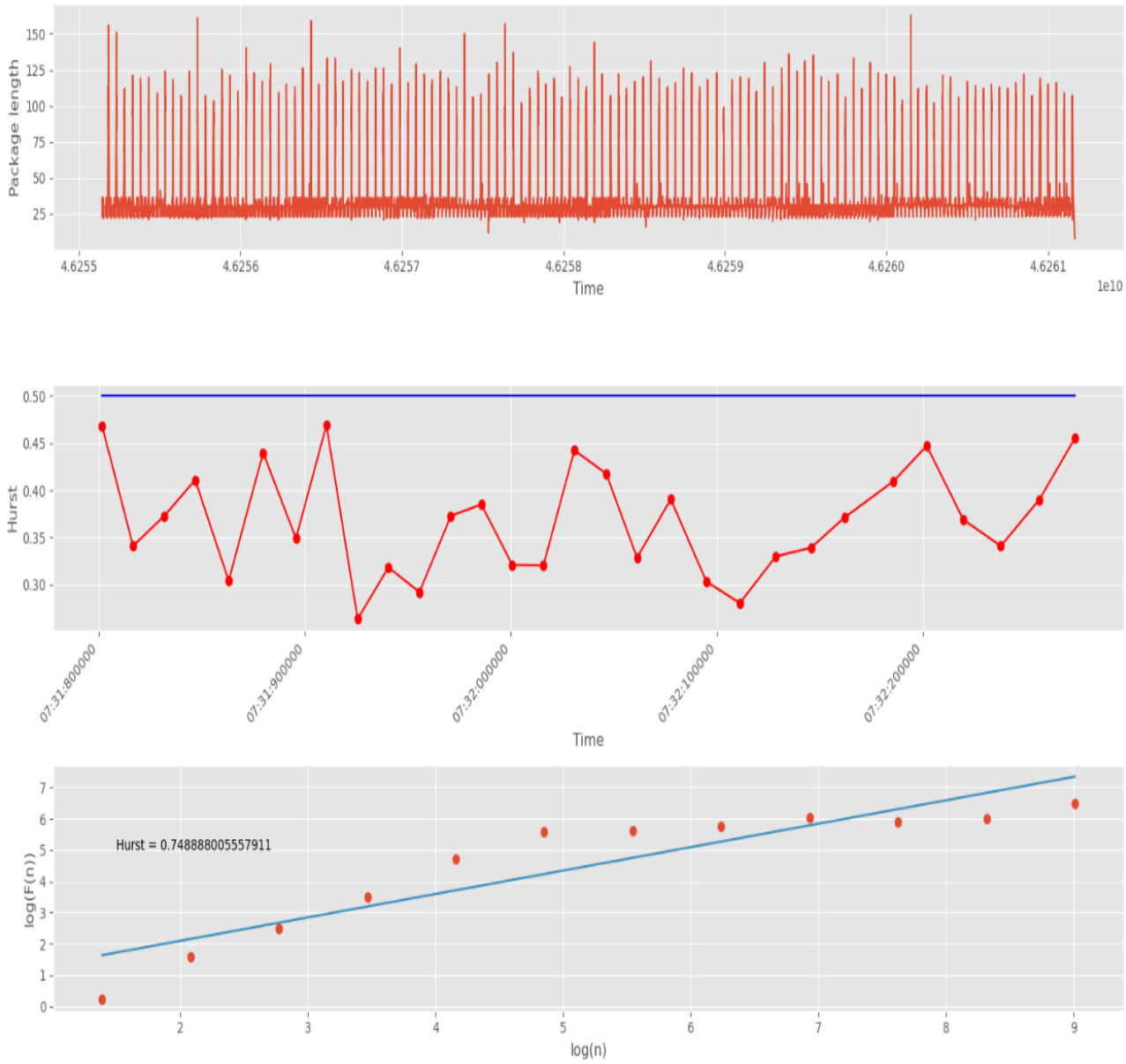

Рисунок 3.5 – Вычисление *H* методом фрактального анализа аномального *UDP* трафика. Разбиение 10000 точек на 29 групп

При анализе 39 групп, каждая из которых состоит из 256 сетевых пакетов, наблюдается проявление свойств самоподобия на некоторых участках (рис. 3.6).

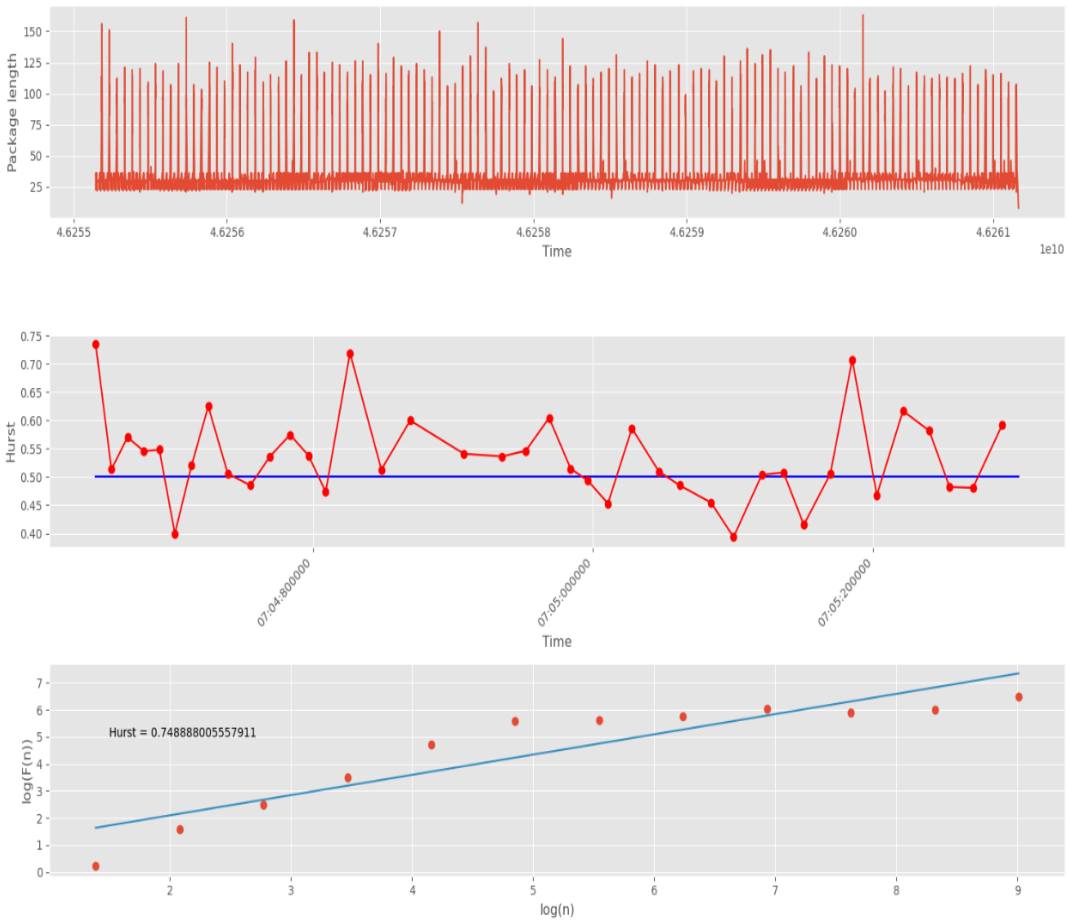

Рисунок 3.6 – Вычисление *H* методом фрактального анализа аномального *UDP* трафика. Разбиение 10000 точек на 39 групп

Такое поведение указывает на ухудшение точности из-за малой выборки сетевых пакетов. Следовательно, такое разбиение является неприемлемым. В качестве оптимального разбиения следует считать предыдущее разбиение, состоящее из 256 сетевых пакетов на интервал. Такое количество пакетов обрабатывается за 0.00125 с, что является существенным достоинством данного подхода.

3.2 Обнаружение аномалий в стационарном сетевом трафике сети передачи данных

3.2.1 Обнаружение аномалий с помощью методов машинного обучения

Существует множество способов, которые позволяют определить аномалии. На рисунках 3.7-3.9 продемонстрирована работа наиболее популярных алгоритмов машинного обучения, протестированных на временных рядах,

сгенерированных с помощью модели авторегрессионного интегрированного скользящего среднего.

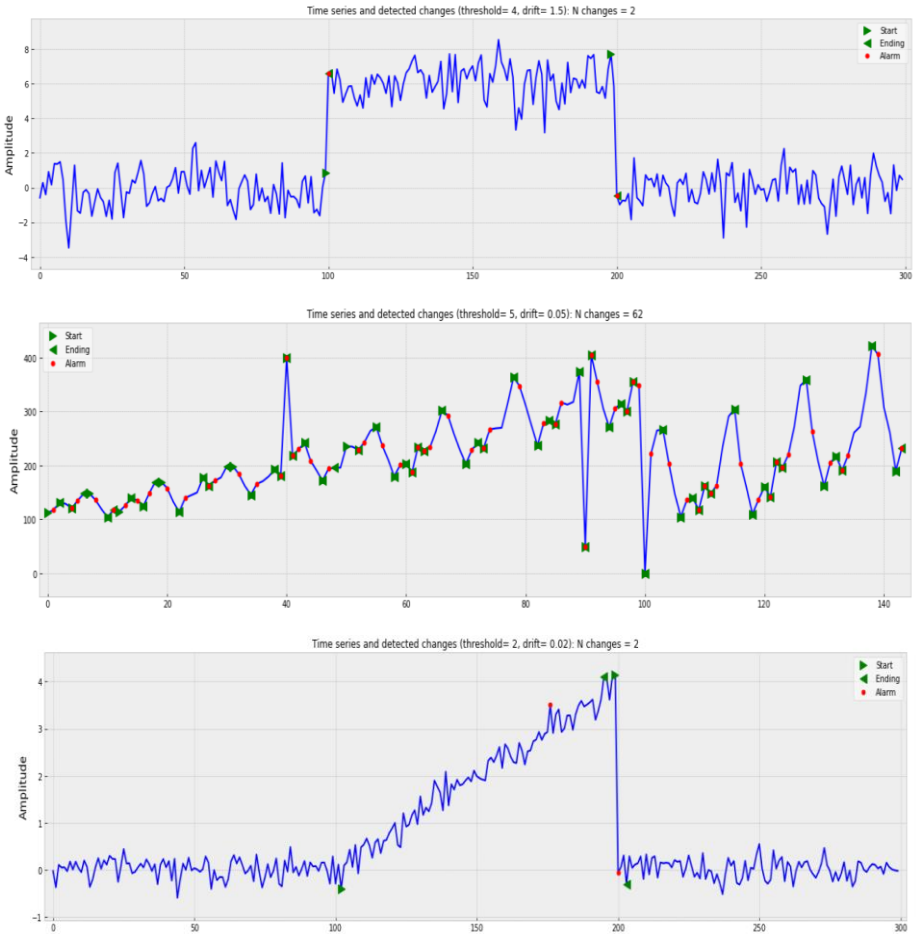

Рисунок 3.7 – Кумулятивные суммы

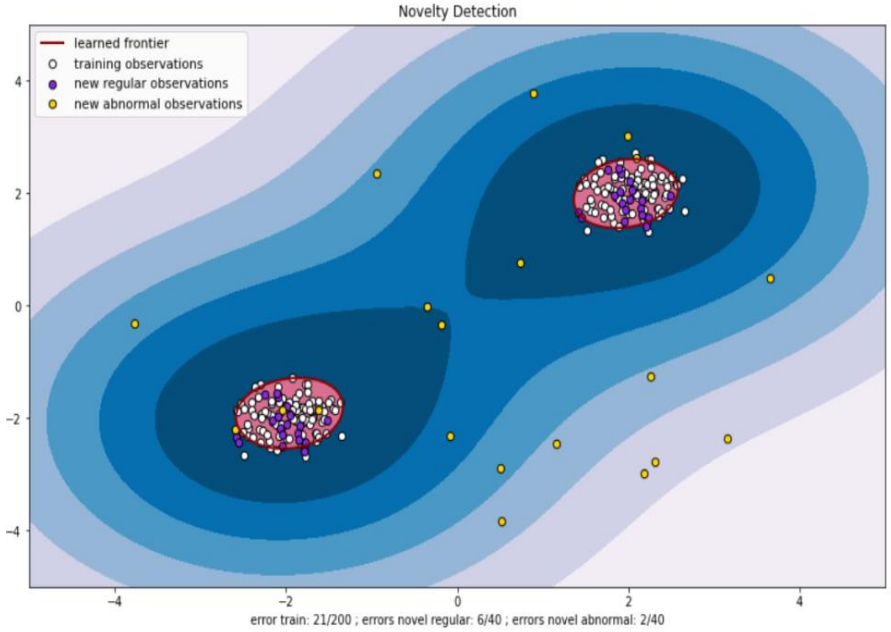

Рисунок 3.8 – Метод опорных векторов

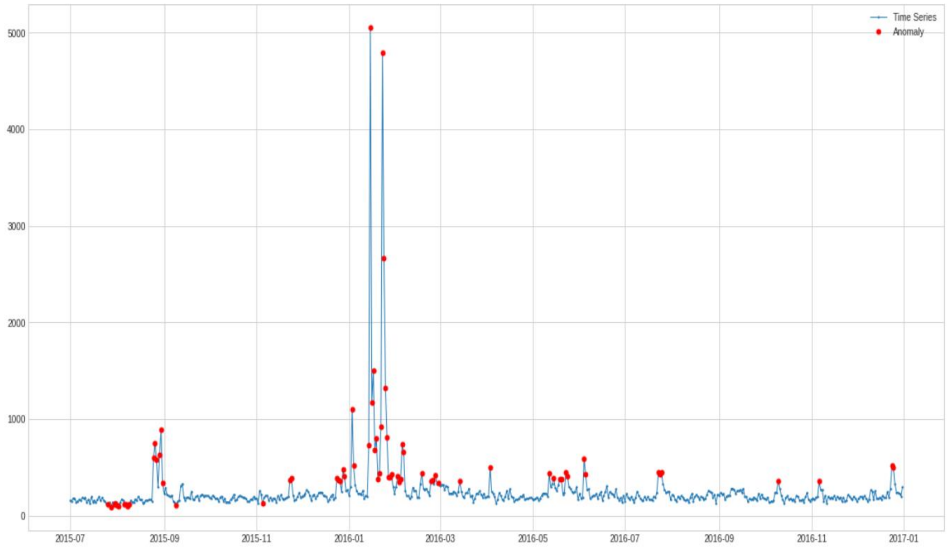

Рисунок 3.10 – Изолированный лес

Как видно из рисунков, алгоритмы отлично справляются с обнаружением аномальных выбросов. В таком случае, аномалия проявляется в виде нестационарности некоторых наблюдаемых временных рядов. Это не только мгновенные скачки амплитуды измерений, но и медленные тренды, практически невидимые за время наблюдений. Однако, при тестировании вышеуказанных алгоритмов на реальном сетевом трафике, оказалось, что не всегда выбросы являются аномальными.

На рисунках 3.10-3.13 изображены протоколы сетевого трафика СПД. Аномальные пакеты помечены красными точками, а легитимные пакеты – зелеными. Как видно из рисунков многие всплески являются легитимными, а отсутствующие всплески – аномальные.

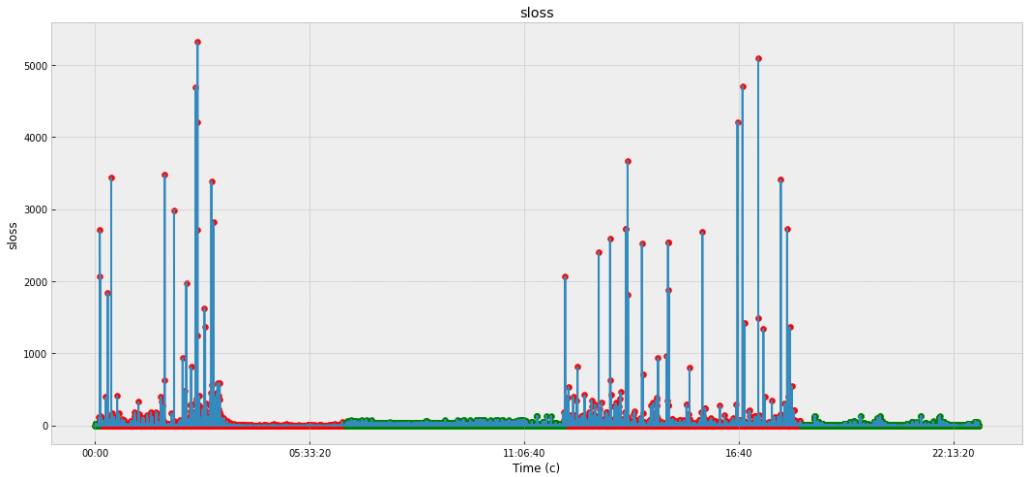

Рисунок 3.10 – Отправленные пакеты повторно переданы или отброшены

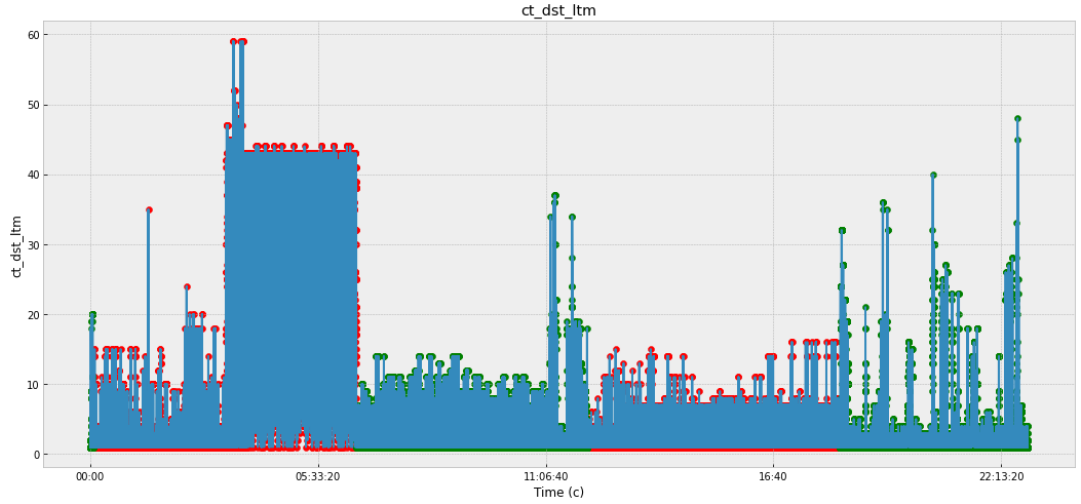

Рисунок 3.11 – Количество подключений к серверу

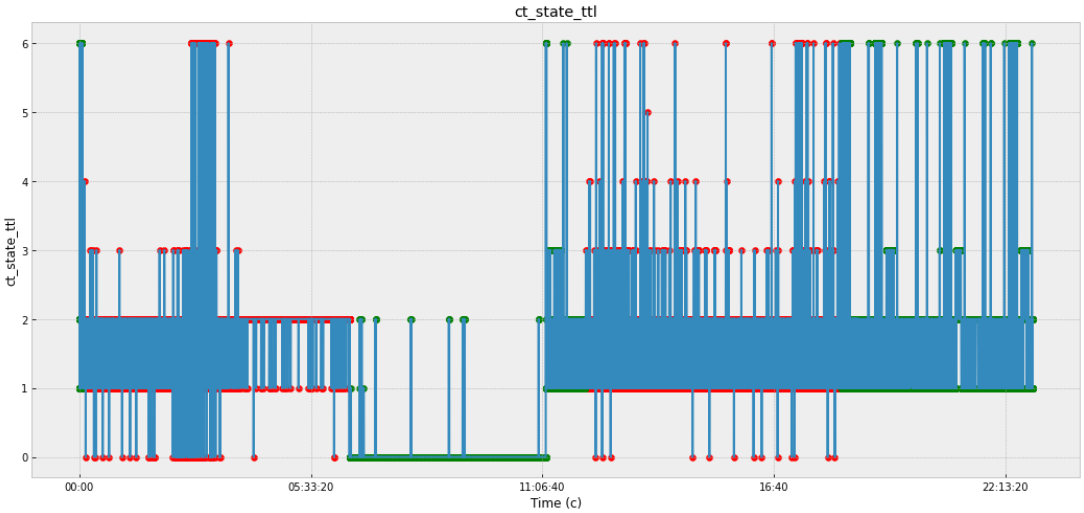

Рисунок 3.12 – Состояние параметров *tcp* заголовка за время жизни *ip*-пакета

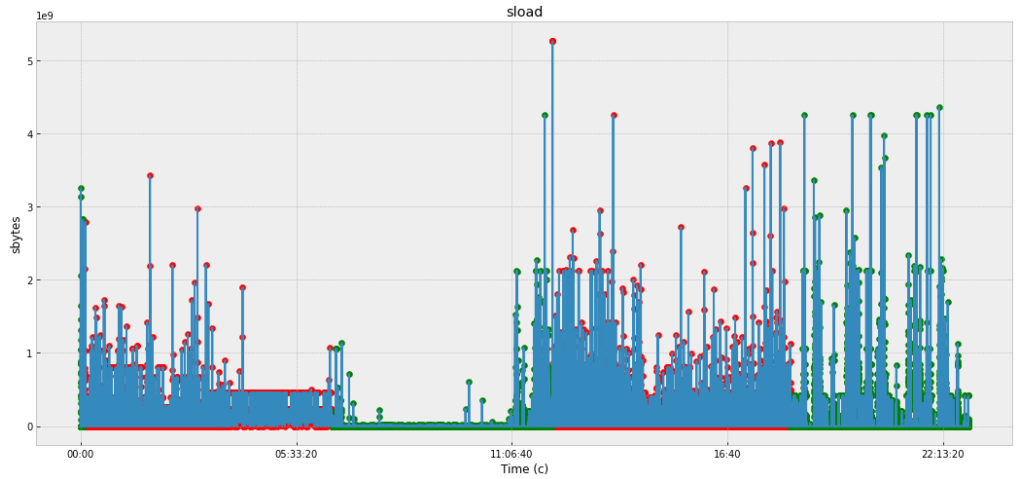

Рисунок 3.13 - Скорость передачи пакетов *bit*/сек

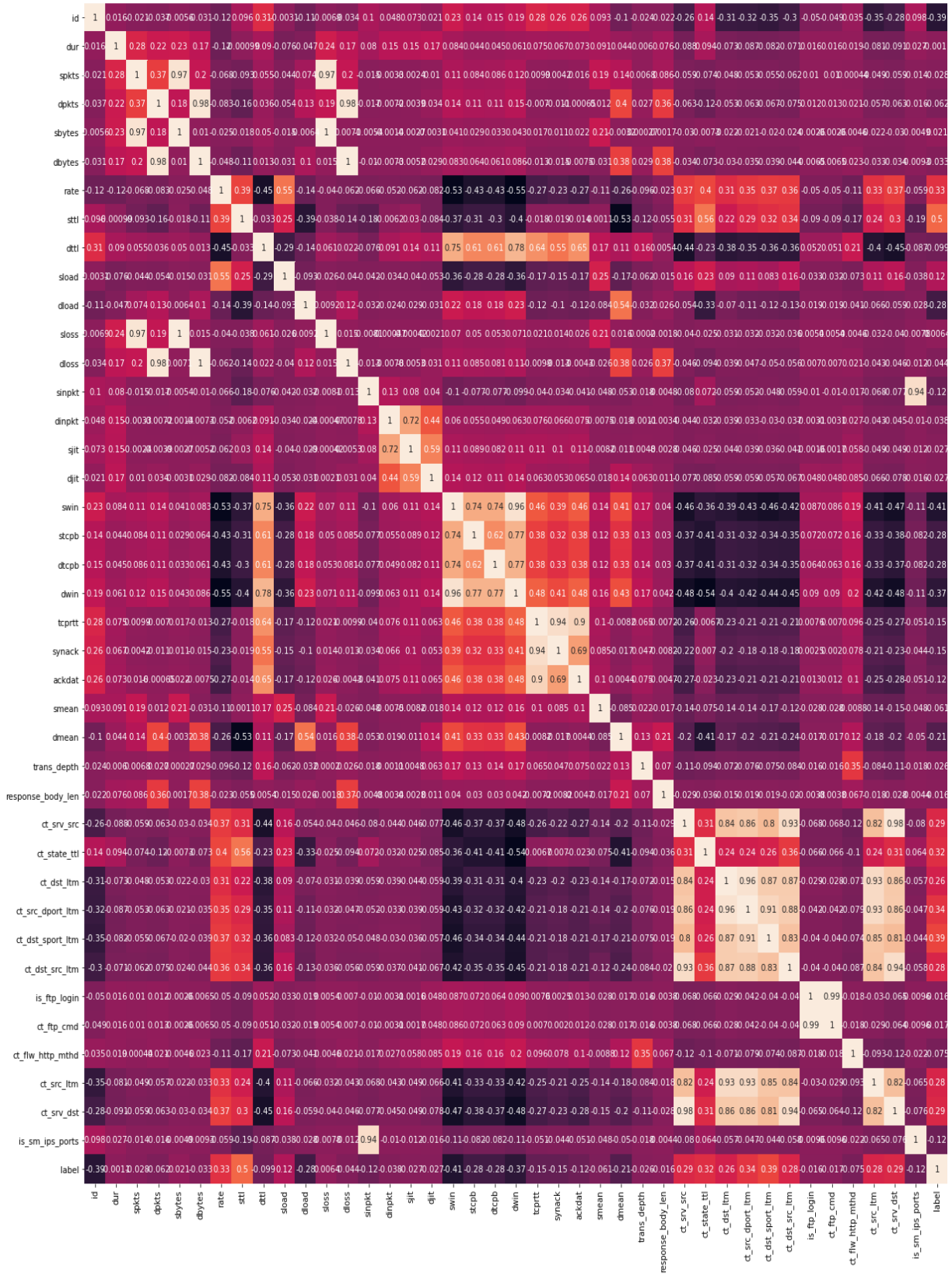

Рисунок 3.14 – Матрица корреляции

Однако, при тестировании вышеуказанных алгоритмов на реальном сетевом трафике оказалось, что не всегда выбросы являются аномальными; происходят ложные срабатывания.

Поэтому остро стоит вопрос по выбору иного подхода к прогнозированию факта воздействия КА и выработке эффективных мероприятий противодействия.

3.2.2 Обнаружение аномалий в сетевом трафике сети передачи дынных с помощью классификаторов

Чтобы оценить эффективности популярных классификаторов, формируется новый датасет, содержащий коррелируемые параметры с аномальными запросами. Для этого строится матрица корреляции (рис. 3.14).

Параметр *label* – индикатор, показывающий наличие аномалий. Для удобства выбрано 20 наиболее коррелируемых с аномалиями параметров (рис. 3.15). Наибольшему воздействию подвергается параметр *sttl* (рис. 3.16).

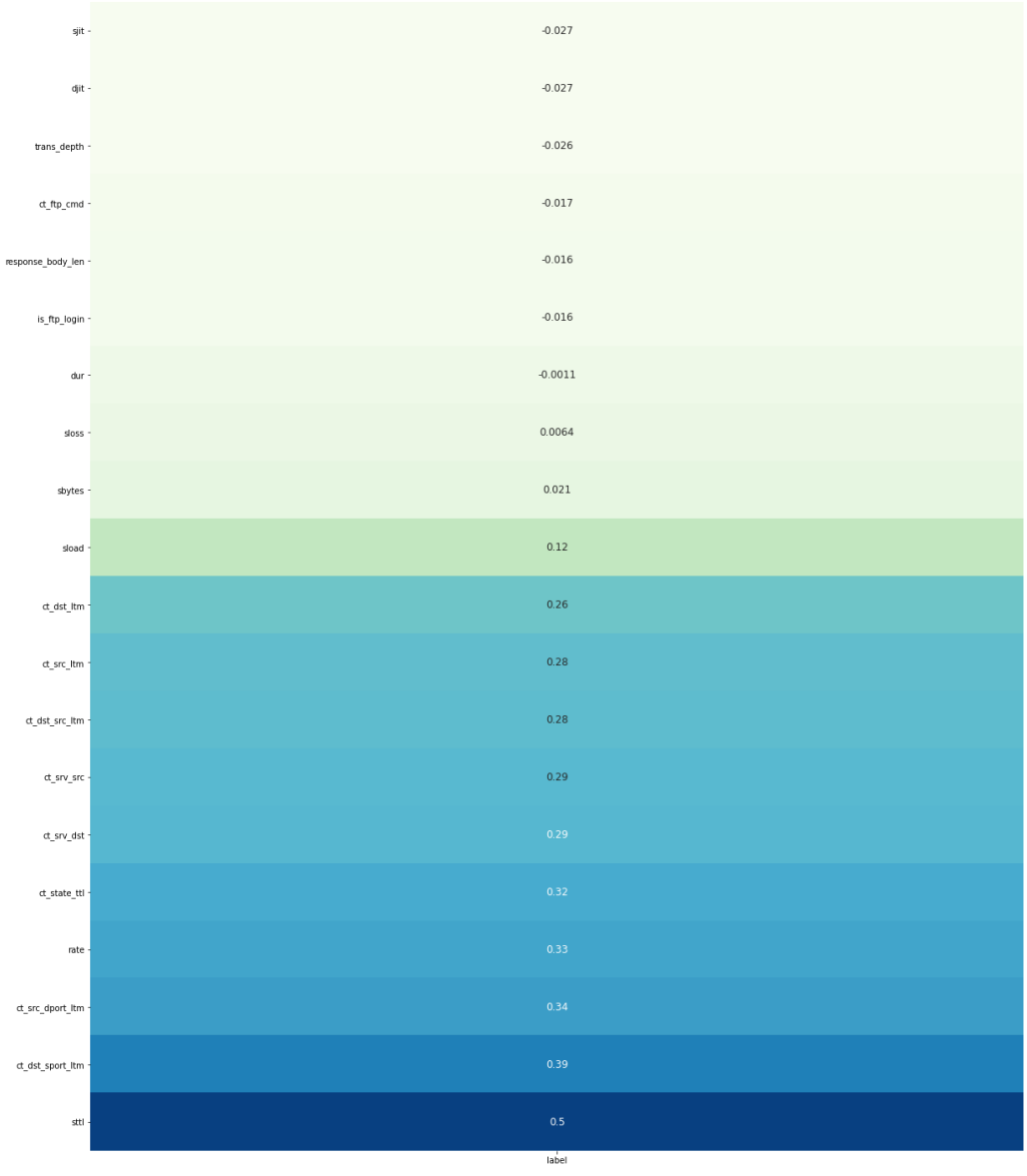

Рисунок 3.15 – Самые коррелируемые параметры

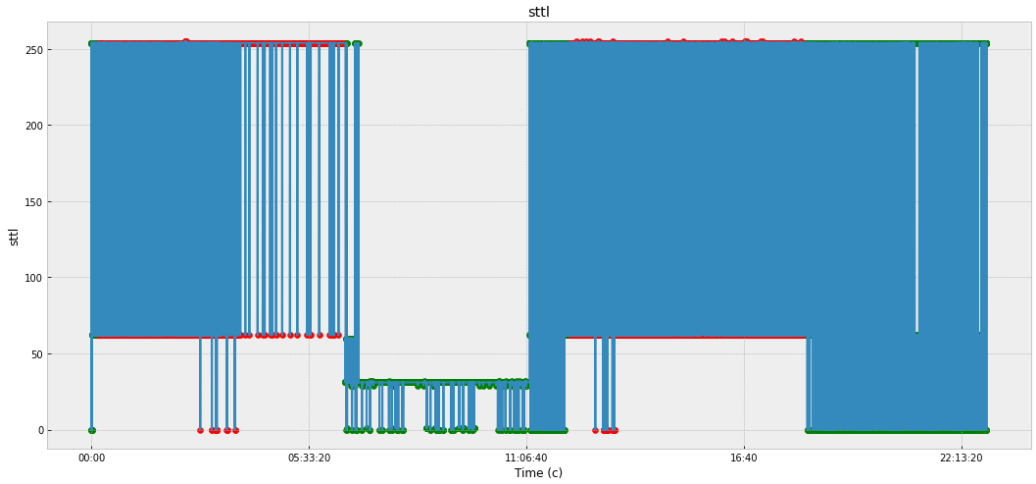

Рисунок 3.16 – Время жизни пакета от источника к отправителю

В качестве классификаторов использовались: логистическая регрессия, метод *k*-ближайших соседей, бэггинг, градиентный бустинг, случайный лес и адаптивный бустинг. Для каждой модели подбирались гиперпараметры влияющие на точность классификации (рис. 3.17).

```
In [49]: models = []
         names = ['LogisticRegression','KNeighborsClassifier','BaggingClassifier','GradientBoostingClassifier','RandomForest

         models.append((LogisticRegression(max_iter=1000,n_jobs=-1)))
         models.append(KNeighborsClassifier(n_jobs=-1))
         models.append(BaggingClassifier())
         models.append(GradientBoostingClassifier())
         models.append(RandomForestClassifier())
         models.append(AdaBoostClassifier())

In [50]: params = {
             models[0]:{'solver': ['sag', 'lbfgs'], 'penalty': ['l1', 'l2']},
             models[1]:{'n_neighbors': list(range(1, 31)),'weights': ['uniform', 'distance']},
             models[2]:{'n_estimators': list(range(10, 31))},
             models[3]:{'loss': ['deviance','exponential'], 'learning_rate': [0.1, 0.03, 0.5], 'max_depth': list(range(3, 30
             models[4]:{'n_estimators':list(range(10, 30)), 'max_depth': list(range(5, 31))},
             models[5]:{'learning_rate': list(np.arange(0.0,2.0,0.1)), 'n_estimators':list(range(50, 100))},
         }

In [51]: import warnings
         warnings.filterwarnings('ignore')

         for name, model in zip(names, models):
             search = GridSearchCV(estimator=model, param_grid=params[model], n_jobs=-1, cv=5)
             search.fit(X_train, y_train)

             print('_____________________________________________________')
             print('Classificator: '+ str(search.best_estimator_))
             print('Best parametrs: '+ str(search.best_params_))
             print('Best score: '+ str(search.best_score_))
             print('_____________________________________________________')
```

Рисунок 3.17 – Подбор гиперпараметров

На рисунке 3.18 представлены наиболее значимые параметры, влияющие на точность классификации и оценка эффективности обнаружения аномалий классификаторами.

```
_____
Classificator: LogisticRegression(max_iter=1000, n_jobs=-1)
Best parametrs: {'penalty': 'l2', 'solver': 'lbfgs'}
Best score: 0.8961687849910088
_____
_____
Classificator: KNeighborsClassifier(n_jobs=-1, n_neighbors=1)
Best parametrs: {'n_neighbors': 1, 'weights': 'uniform'}
Best score: 0.999323288700103
_____
_____
Classificator: BaggingClassifier(n_estimators=15)
Best parametrs: {'n_estimators': 15}
Best score: 0.9995141569933788
_____
_____
Classificator: GradientBoostingClassifier(learning_rate=0.5, max_depth=6)
Best parametrs: {'learning_rate': 0.5, 'loss': 'deviance', 'max_depth': 6}
Best score: 0.9999305932110504
_____
_____
Classificator: RandomForestClassifier(max_depth=23, n_estimators=13)
Best parametrs: {'max_depth': 23, 'n_estimators': 13}
Best score: 0.9991324294388775
_____
_____
Classificator: AdaBoostClassifier(learning_rate=1.9000000000000001, n_estimators=51)
Best parametrs: {'learning_rate': 1.9000000000000001, 'n_estimators': 51}
Best score: 0.9999305932110504
```

Рисунок 3.18 - Сравнение точности классификаторов

Несмотря на высокую точность обнаружения КА, современные классификаторы имеют существенный недостаток - большое количество ложных срабатываний. В этом легко убедиться, построив матрицу ошибок первого и второго рода, по которой определялась не только точность, но и количество ложных срабатываний (рис. 3.19).

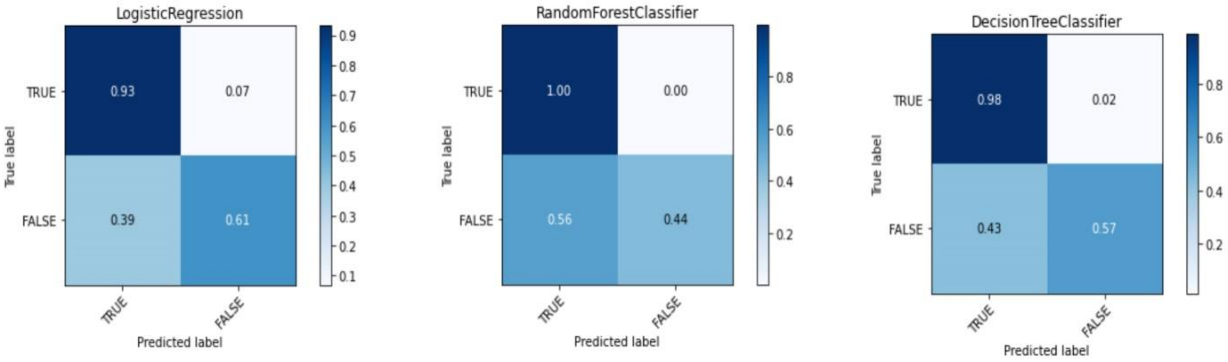

Рисунок 3.19 – Матрица ошибок первого и второго рода логистической регрессии, случайного леса и дерева решений

Полученные результаты подтверждают тот факт, что проблемой современных классификаторов является плохая способность распознавать неизвестные аномалии (атаки нулевого дня) [119-129].

3.2.3 Обнаружение аномалий в сетевом трафике сети передачи данных с помощью автокодировщика

Автокодировщики – нейронные сети прямого распространения, которые восстанавливают входной сигнал на выходе. Внутри у них имеется скрытый слой, который представляет собой код, описывающий модель. Автокодировщик состоит из двух частей кодировщик и декодировщик. Кодировщик пытается сжать данные, а затем восстановить с помощью декодировщика. Автокодировщики конструируются таким образом, чтобы иметь возможность точно скопировать вход на выходе (рис. 3.20).

Тем не менее, входной сигнал восстанавливается с ошибками из-за потерь при кодировании, и для того, чтобы их минимизировать, сеть вынуждена учиться отбирать наиболее важные признаки.

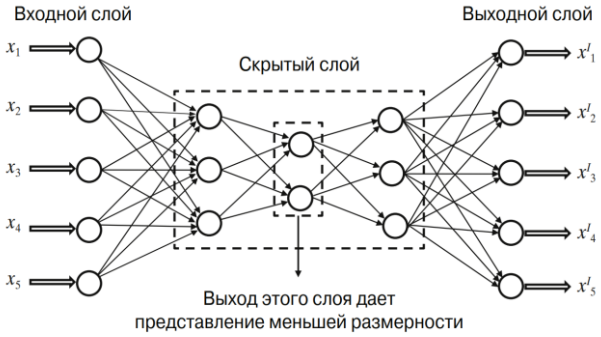

Рисунок 3.20 – Автокодировщик

Идея состоит в том, чтобы научить сеть декодировать значения, которые она видела, или, другими словами, приближать тождественное отображение. Если обученный автокодировщик встречает аномальный образец, он, вероятно, воссоздает его с высокой степенью ошибки, просто потому, что никогда его не видел.

На рисунке 3.21 представлена блок-схема методики обнаружения аномалий в стационарном сетевом трафике СПД, который базируется на модели автокодировщика.

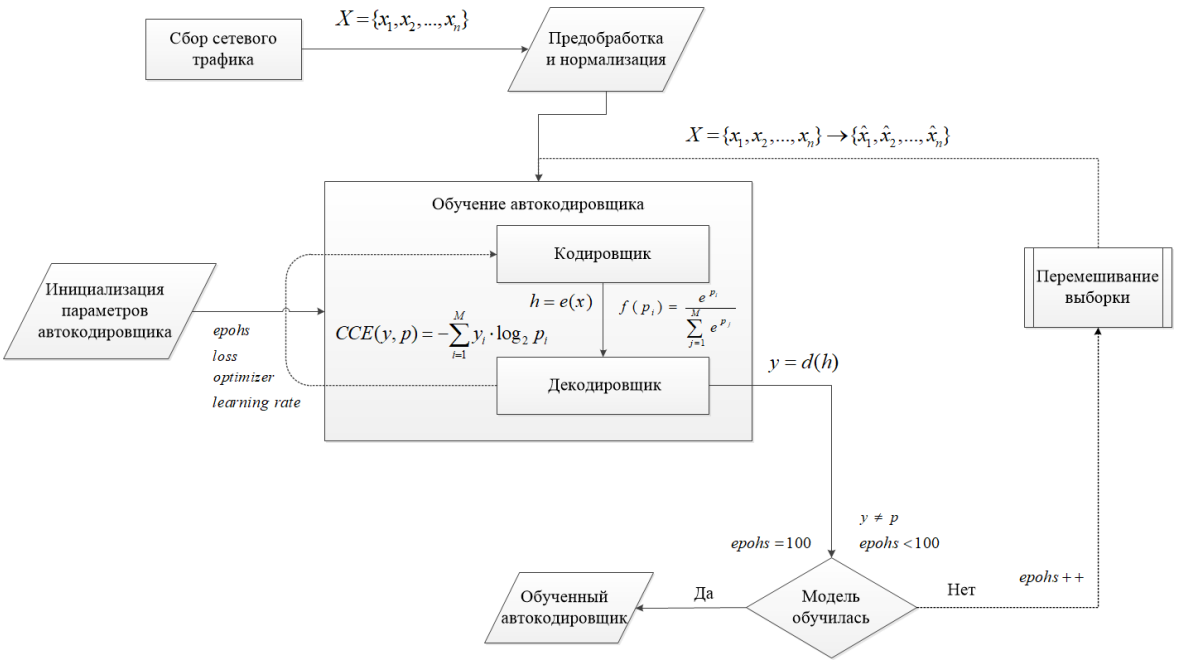

Рисунок 3.21 – Блок-схема методика обнаружения аномалий в стационарном сетевом трафике СПД

Обучение автокодировщика осуществляется по блок-схеме, представленной на рисунке 3.22.

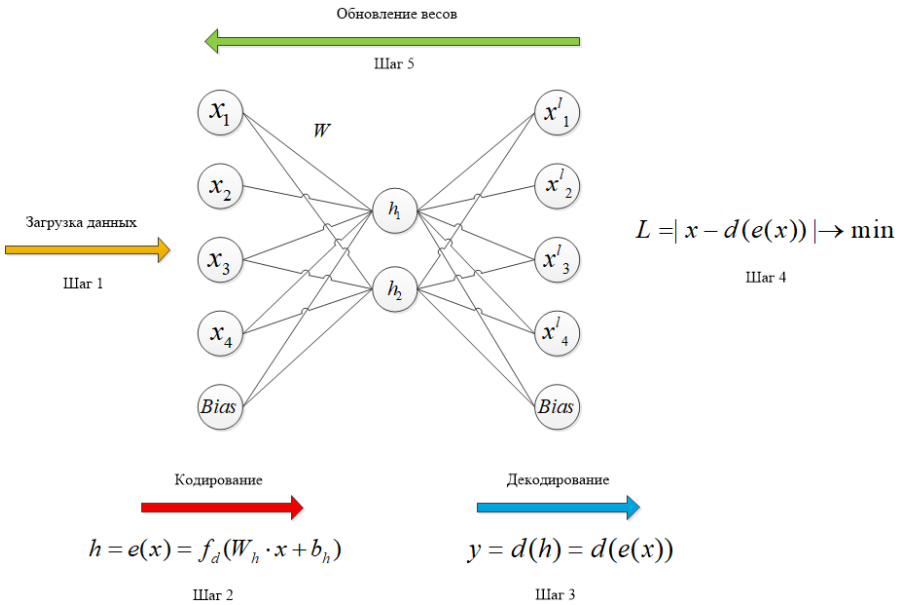

Рисунок 3.22 – Блок-схема обучения автокодировщика

Алгоритм обучения автокодировщика:

Шаг 1: Входные данные x_n подаются на вход нейронной сети.

Шаг 2: Кодировщик кодирует входной вектор x_n в вектор h меньшего размера, чем входной:

$$h = e(x) = f_d(W_h \cdot x + b_h) \tag{3.1}$$

Где f_d - функции активации (*ReLU*, сигмоидальная функция, гиперболический тангенс) промежуточного слоя;

b_h - Вектор смещений промежуточного слоя;

W_h - Матрица весов промежуточного слоя;

Шаг 3: Вектор h декодируется, чтобы воссоздать ввод:

$$y = d(h) = f_e(W_y \cdot h + b_y) \tag{3.2}$$

Параметры f_e, W_y, b_y соответствуют входному слою.

Выход будет такого же размера, что и вход.

Шаг 4: Рассчитывается ошибка L:

$$L = | x - d(e(x)) | \rightarrow \min \tag{3.3}$$

Ошибка это разница между входным и выходным вектором. Цель – минимизировать ошибку, чтобы выходной вектор был похож на входной вектор.

Шаг 5: С помощью алгоритма обратного распространения ошибки нужно обновить веса.

Шаг 6: Повторяются шаги с 1 по 5 до тех пор, пока ошибка не снизится до приемлемого результата.

После обучения автокодировщик сможет восстановить наблюдения с небольшой ошибкой или разницей между исходными данными.

Однако, когда он пытается предсказать/реконструировать аномальное наблюдение, то обнаружит, что никогда не видел таких последовательностей во время обучения. Таким образом, ошибка восстановления между исходными и восстановленными данными будет выше для аномальных данных, чем для обычных.

У предложенного подхода (рис. 3.23) в качестве функции потерь выступает – категориальная (перекресная) кроссентропия, которая часто используется в задачах классификации:

$$CCE(y,p) = -\sum_{i=1}^{M} y_i \cdot \log_2 p_i \tag{3.4}$$

где p прогнозируемая вероятность выходной метки y, а M - количество классов.

Функцией определяющей выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения, является softmax. Она является обобщенной логистической функцией для многомерного случая:

$$f(p_i) = \frac{e^{p_i}}{\sum_{j=1}^{M} e^{p_j}} \tag{3.5}$$

Программная реализация модели автокодировщика представлена на рисунке 3.23. Нейронная сеть проектировалась на *Framework Keras* (*API TensorFlow*). Для подбора гиперпараметров нейронной сети не подходит инструментов машинного обучения *GridSearchCV* (библиотека *scikit-learn*), т.к. он обращается к *KerasClassifier*, который не работает с трехмерными тензорами используемыми в предложенной модели.

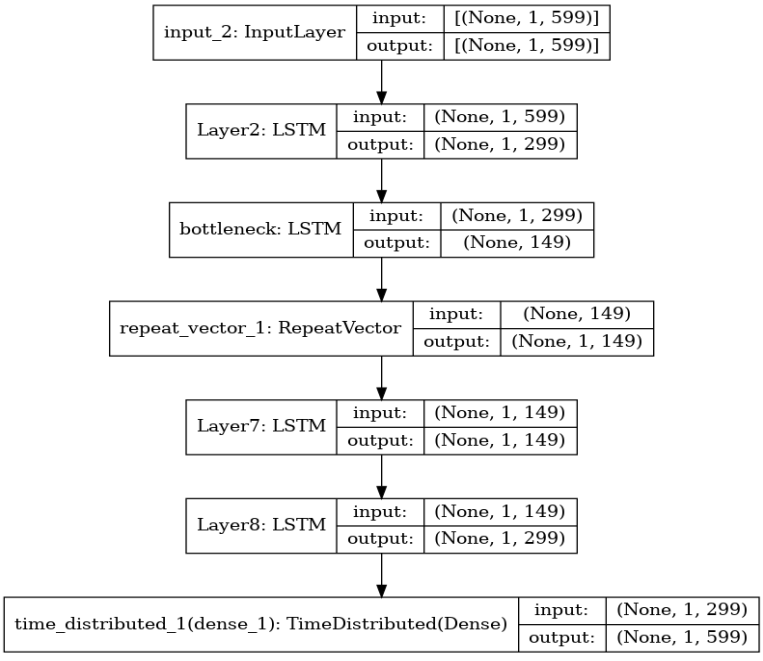

Рисунок 3.23 – Автокодировщик с ячейками *LSTM*

Поэтому на языке программирования *Python* был реализован дополнительный модуль, предназначенный для подбора *learning rate* (рис. 3.24).

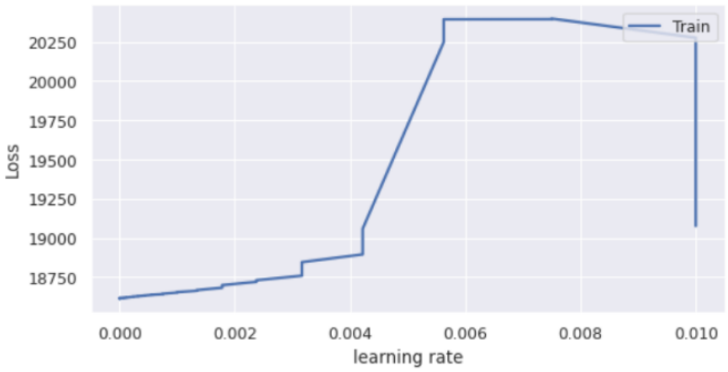

Рисунок 3.24 – Зависимость скорости обучения модели от значения ошибок

Из рисунка 3.24 прослеживается корреляция между скоростью обучения и значениями функции потерь.

Из проведенного эксперимента была выбрана скорость обучения модели с самым минимальным средним значением ошибки (рис. 3.25).

	loss
lr	
3.181966e-08	18614.585938
1.006794e-08	18614.590820
1.342392e-08	18614.592773
1.789856e-08	18614.592773
2.386475e-08	18614.593750

Рисунок 3.25 – Пример скорости обучения модели с самым минимальным средним значением ошибки

Для прогнозирования необходимо пропустить через автокодировщик раннее предобработанную выборку, и обучать модель до значительного снижения ошибки восстановления (рис 3.26).

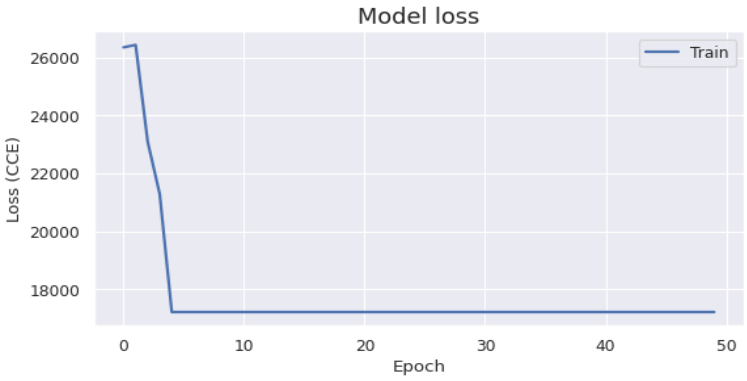

Рисунок 3.26 – Уменьшение ошибки восстановления с увеличением эпох обучения

После обучения, перед тестированием модели необходимо определить пороговое значение, преодолевая которое, запросы будут помечаться как аномальные. Пороговое значение, может быть выбрано по максимальной ошибке, возникающей при обучении автокодировщика, т.е. с какой максимальной ошибкой он восстанавливает нормальные данные и аномальные. Но такой режим работы будет являться агрессивным. Поэтому в качестве такого значения выбираем пересечение полноты и точности, чтобы по максимуму охватить аномалии и исключить ложные срабатывания. Пересечение (красная вертикальная линия) кривых точности и полноты показывает оптимум в обнаружении аномалий таким образом, чтобы не блокировали легитимные запросы (рис. 3.27).

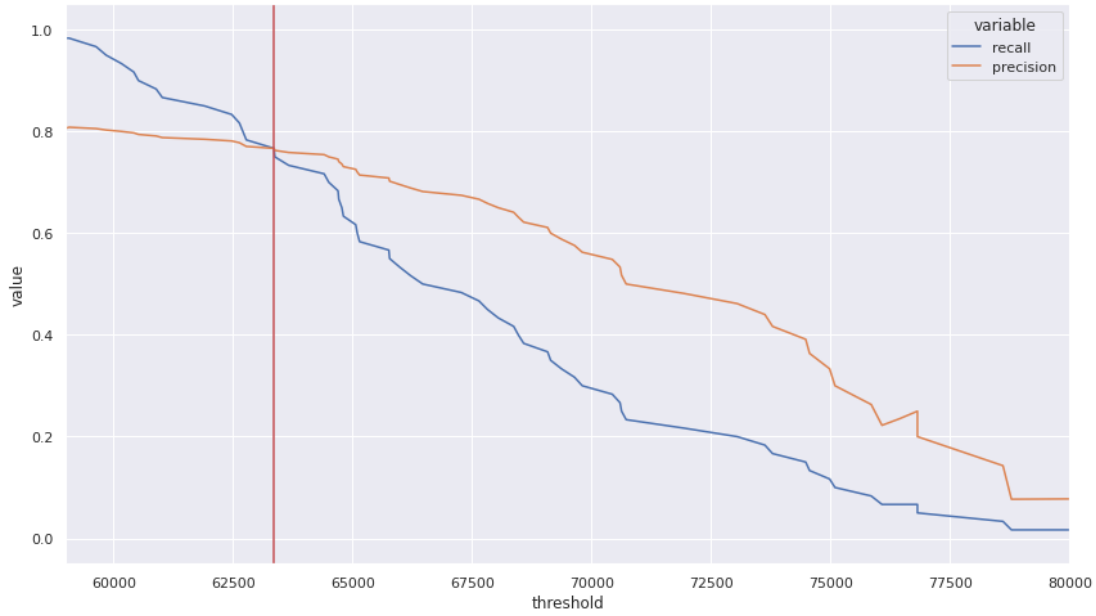

Рисунок 3.27 – Выбор порогового значения по пересечению точности и полноты

Физическим смыслом точности и полноты являются ошибки первого и второго рода. На рисунках 3.28 и 3.29 продемонстрирована работа автокодировщика на выборке, состоящей из неизвестных при обучении атаках – *XSS*. Аномальные (красным цветом) и неаномальные (синим цветом) запросы, которые отделены пороговым значением (прямая линия).

Анализ показал, что методика хорошо справляется с выявлением неизвестных раннее КА, при этом пороговое значение можно отрегулировать, увеличив точность обнаружения. Для оценки точности алгоритма построена *ROC* кривая (рис 3.30), и матрица ошибок первого и второго рода (рис 3.31).

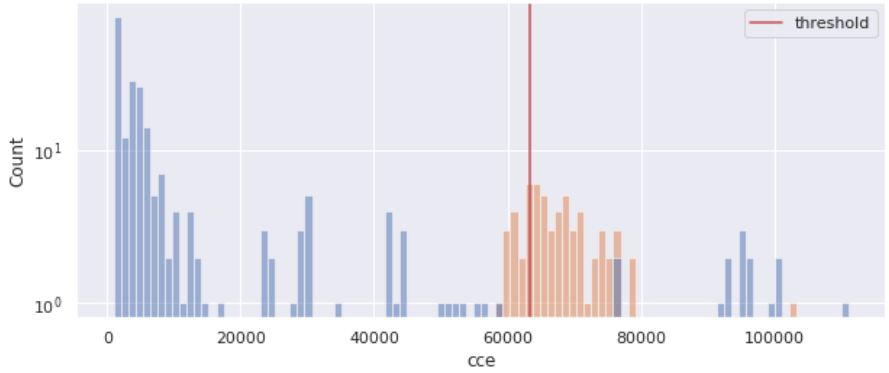

Рисунок 3.28 – Результат работы методики раннего обнаружения КА в сетевом трафике СПД

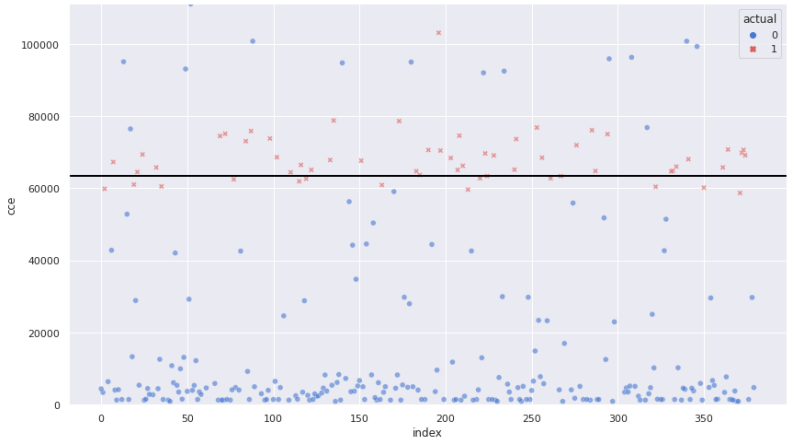

Рисунок 3.29 – Результат работы методики раннего обнаружения КА в сетевом трафике СПД

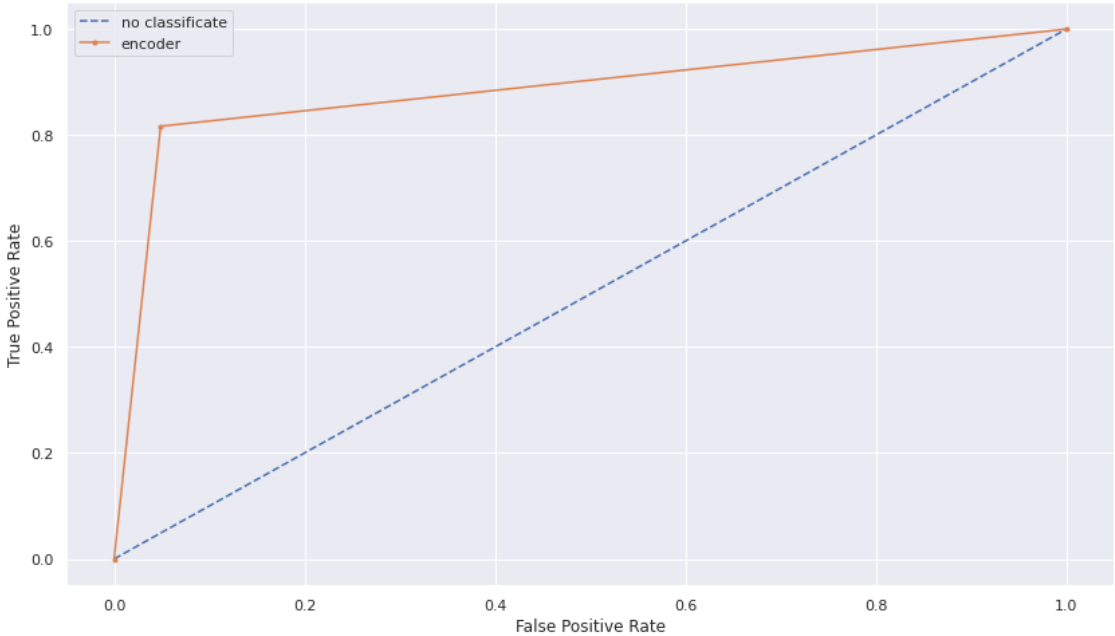

Рисунок 3.30 – *ROC*-кривая

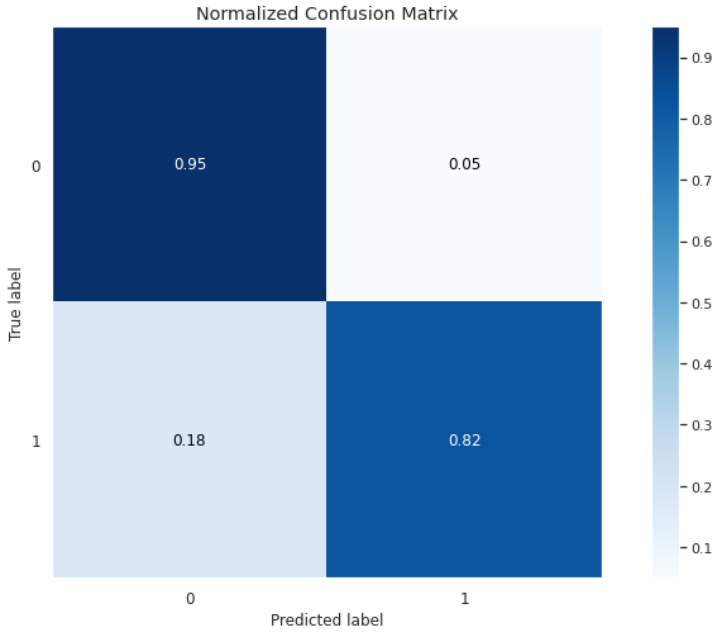

Рисунок 3.31 – Матрица ошибок

ROC-кривая позволяет оценить качество бинарной классификации и отображает соотношение между долей объектов от общего количества верно классифицированных аномалий, и легитимных запросов, ошибочно классифицированных как аномальные. Матрица ошибок интерпретируется согласно таблице 3.1.

На матрице ошибок первого и второго рода (рис 3.31) видно, что алгоритм распознал 82% неизвестных раннее КА (атаки нулевого дня), и верно определил, что 95% легитимных запросов не являются аномальными. При этом количество ложных срабатываний составило 5%.

Таблица 3.1 – Ошибки первого и второго рода

Прогноз	**Реальность**	
	+	**-**
+	*True Positive* (истинно-положительное решение): прогноз совпал с реальностью, результат положительный произошел, как и было предсказано моделью	*False Positive* (ложноположительное решение): ошибка 1-го рода, модель предсказала положительный результат, а на самом деле он отрицательный
-	*False Negative* (ложноотрицательное решение): ошибка 2-го рода – модель предсказала отрицательный результат, но на самом деле он положительный	*True Negative* (истинно-отрицательное решение): результат отрицательный, прогноз совпал с реальностью

3.3 Выводы по третьему разделу

1. В третьем разделе разработана методика раннего обнаружения аномалий в сетевом трафике СПД, позволяющая обнаруживать аномалии на раннем этапе их проявления с помощью методов машинного обучения для стационарного сетевого трафика СПД и фрактального анализа для нестационарного. Проведен эксперимент по нахождению рационального числа пакетов, необходимых для своевременного выявления аномалий с помощью фрактального анализа, с целью достижения своевременности и полноты при обнаружении аномалий в нестационарном трафике. Вычислено оптимальное число пакетов необходимых для выявления аномалий в СПД за интервал времени.

2. Проведен анализ алгоритмов машинного обучения и классификаторов, предназначенных для выявления аномалий. Выявлены их недостатки, которые главным образом заключаются в большом количестве ложных срабатываний, а также отсутствии возможности обнаружения КА «нулевого дня».

3. Разработана модель автокодировщика, позволяющая выявлять КА «нулевого дня» в стационарном сетевом трафике, с минимальным количеством ложных срабатываний. Главным достоинством модели является возможность работы в режиме реального времени, а также возможность работы с любыми видами трафика. Выявление факта воздействия КА производится за несколько микросекунд в зависимости от производительности вычислительной техники. К числу других достоинств этого подхода следует отнести нетребовательность к системным ресурсам.

РАЗДЕЛ 4 МЕТОДИКА КЛАССИФИКАЦИИ КОМПЬЮТЕРНЫХ АТАК В СЕТЕВОМ ТРАФИКЕ СЕТИ ПЕРЕДАЧИ ДАННЫХ

4.1 Методика классификации компьютерных атак в сетевом трафике сети передачи данных

С целью постоянного мониторинга и обнаружения аномальной активности трафика в СПД при КА, а также уменьшения ложных срабатываний, необходимо учитывать наличие большого количества сетевых маршрутов, на которых периодически возникают резкие колебания задержки в передаче данных и большие потери пакетов, появление новых свойств сетевого трафика, а также необходимость обеспечения высокого качества обслуживания приложений.

Все это послужило поводом для разработки методики классификации КА в сетевом трафике СПД (рис. 4.1).

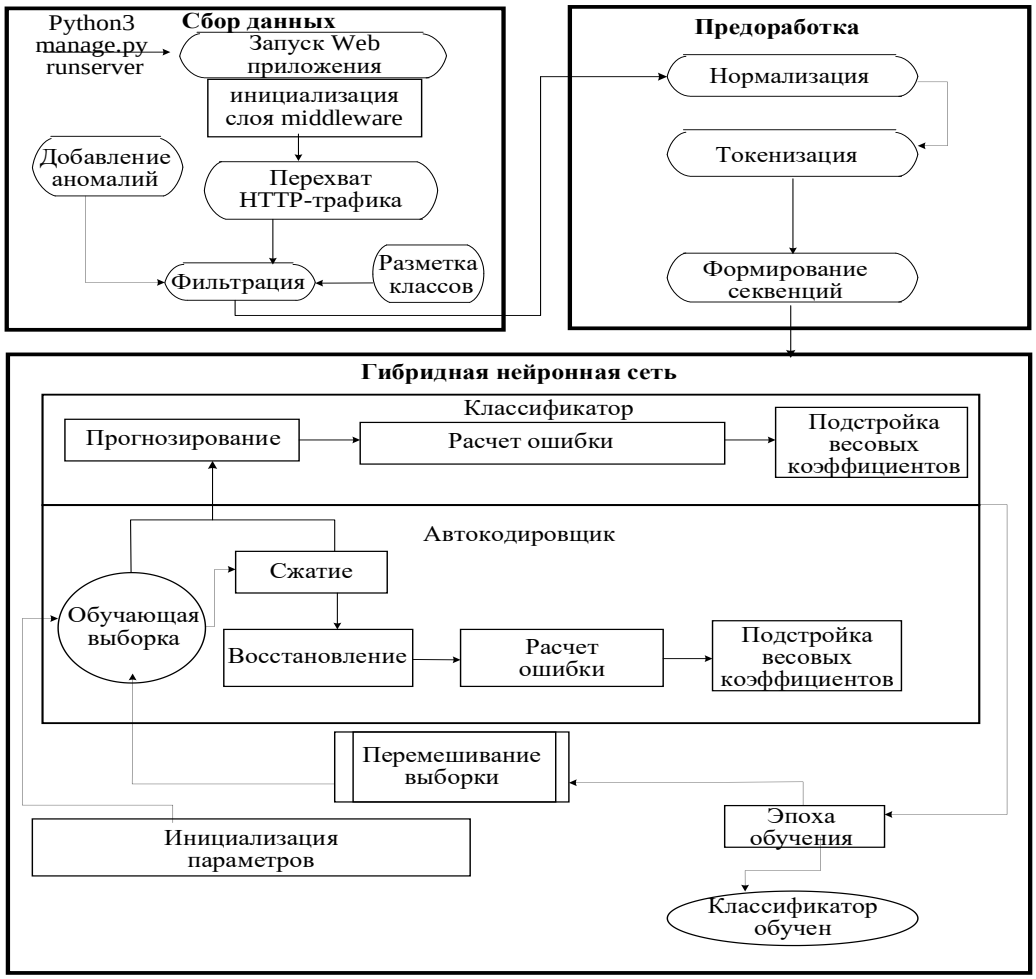

Рисунок 4.1 – Основные этапы методики классификации КА в сетевом трафике СПД

Методика позволяет выявлять КА с использованием гибридной нейронной сети, состоящей из классификатора и автокодировщика, обученного на основе данных работы функционирования СПД, учитывающего все отклонения от ее штатной работы.

В процессе работы классификатор дополнительно обучается на скрытых латентных представлениях полученных автокодировщиком, т.е. в итоге получается генеративно-состязательная сеть, в которой нейронные сети учатся друг у друга.

Данная методика базируется на рассмотренной в третьем разделе методике с измененным этапом обнаружения КА в стационарном трафике. В качестве метода машинного обучения предлагается использовать гибридную нейронную сеть (рис. 4.2).

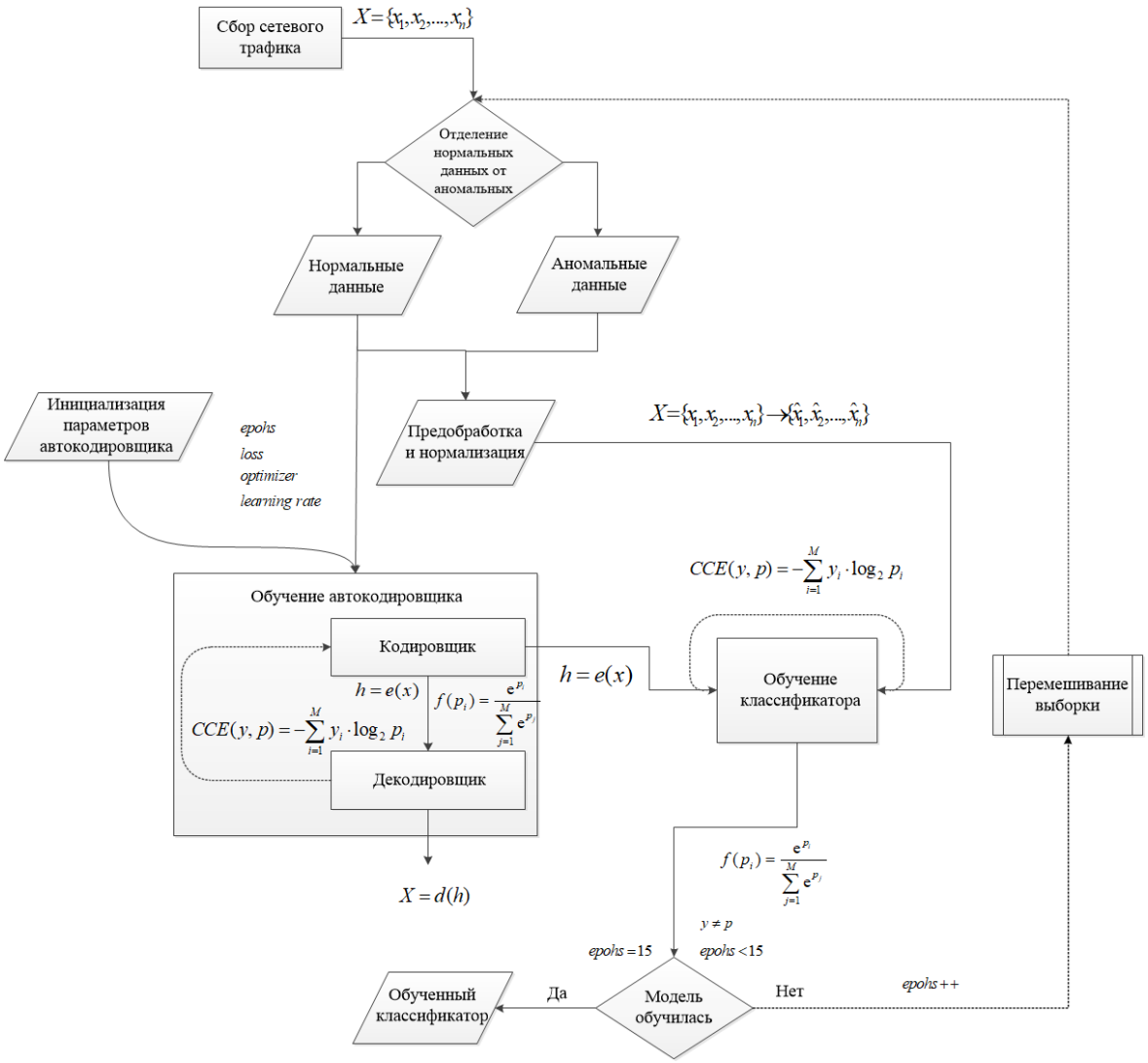

Рисунок 4.2 – Гибридная нейронная сеть

Для вычисления выходного значения сигналов нейронной сети, используется функция активации.

Активационной называется функция, аргументом которой является взвешенная сумма входов искусственного нейрона (4.1), а значением – выход нейрона (4.1):

$$p = \sum_{i=1}^{N} w_i x_i - b \tag{4.1}$$

$$y = f(p) \tag{4.2}$$

где p – взвешенная сумма входов нейрона, N – число входов нейрона, x_i – входной сигнал, w_i – вес входного нейрона, b – смещение выходного значения, $f(p)$ – активационная функция, y – выходное значение нейрона.

В автокодировщике и классификаторе используется функция *softmax,* которая применяется на выходном слое нейронной сети, для решения задач множественной классификации и представляет собой логистическую функцию, обобщенную для многомерного случая:

$$f(p_i) = \frac{\mathrm{e}^{p_i}}{\sum_{j=1}^{M} e^{p_j}} \tag{4.3}$$

где M – количество классов.

В таком случае в качестве функции потерь используется категориальная кросс энтропия, которая, в случае мульти-классовой классификации (M>2), рассчитывается, как сумма значений логарифмических функций потерь для каждого прогноза наблюдаемых классов:

$$CCE(y, p) = -\sum_{i=1}^{M} y_i \cdot \log_2 p_i \tag{4.4}$$

где y – истинное значение класса, p – предсказанное значение класса

Обучение нейронной сети, осуществляется на основе алгоритма «обратного распространения», который позволяет рассчитать значения весовых коэффициентов таким образом, чтобы ошибка сети была минимальна. Для

минимизации ошибки необходимо изменять веса в направлении противоположном градиенту. Соответственно для классификатора и автокодировщика расчет весов будет производиться по формуле (4.5):

$$\Delta w = -\alpha \cdot \frac{dCCE}{dw} \tag{4.5}$$

где CCE – функция ошибок, Δw – величина, на которую необходимо изменить значение w, α – скорость обучения (*learning rate*).

Таким образом, для минимизации ошибки необходимо изменять w в направлении противоположном градиенту [125, 126].

4.2 Программная модель нейронной сети

Модель нейронной сети состоит из рекуррентных ячеек – *LSTM* и *GRU*. На вход нейронной сети подаются тензоры размерностью в 120 векторных признаков. Выходных слоев у нейронной сети несколько. Выходной слой у автокодировщика имеет точно такую же размерность, как и входной. Выходной слой классификатора 6. Он определяет, является ли запрос аномальным или легитимным.

У нейронной сети имеется два выхода, поэтому она называется гибридной. Левая ветвь на рисунок 4.3 соответствует автокодировщику, а правая – классификатору. Гибридная сеть имеет различные по своему назначению слои. Наиболее специфичными слоями являются следующие:

Dropout (отсеивать) – позволяет решить проблему переобучения в нейронных сетях;

Bidirectional (двунаправленный) – с помощью этой формы генеративного глубокого обучения выходной слой может одновременно получать информацию из прошлого (назад) и будущего (вперед) состояний;

Bottleneck (узкое место) – уменьшает количество свойств, а значит и операций в каждом слое, что позволяет сохранить на высоком уровне скорость получения результата.

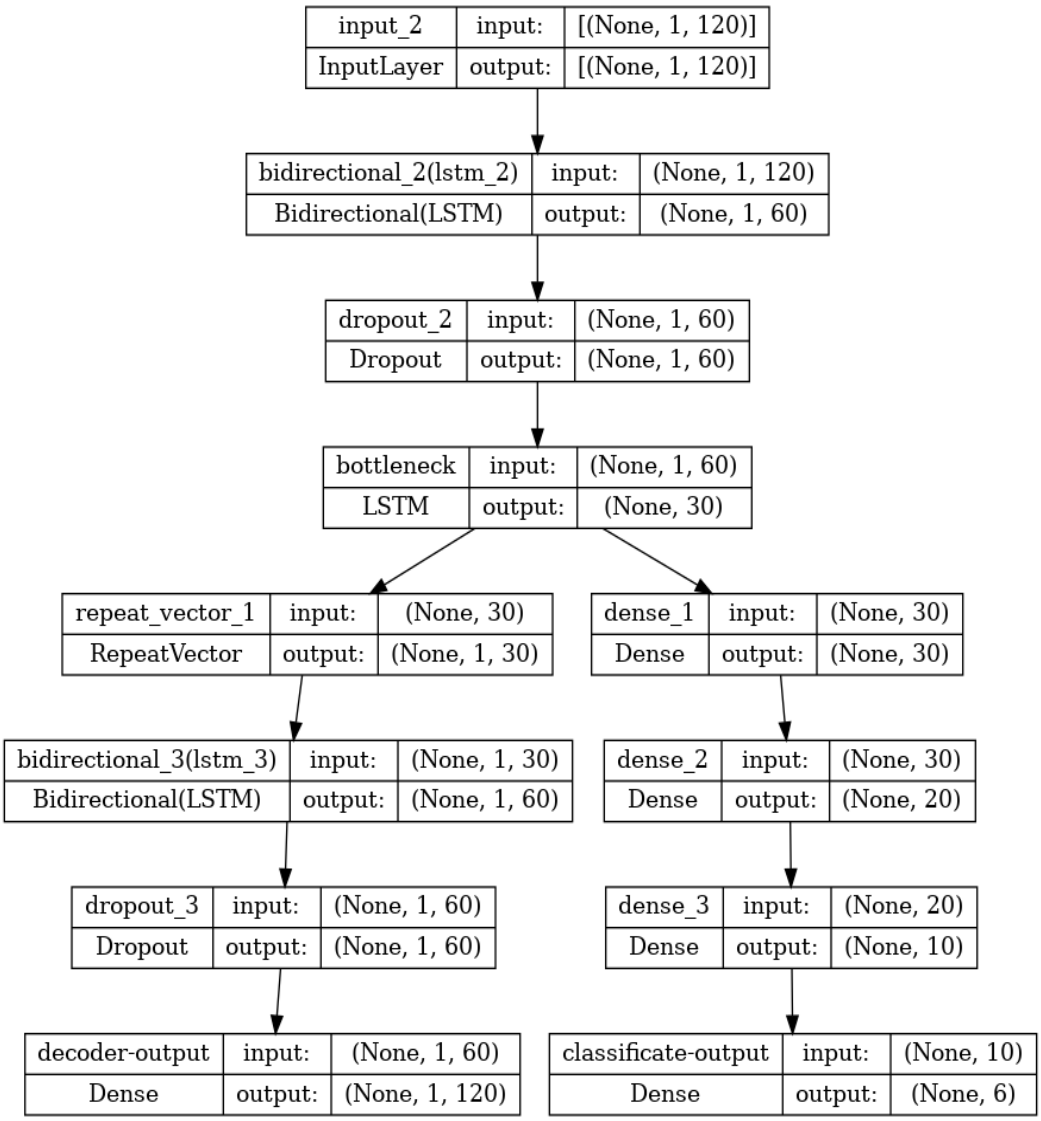

Рисунок 4.3 – Граф гибридной нейронной сети

Входной слой гибридной нейронной сети имеет 120 нейронов, применяющиеся как для автокодировщика, так и для классификатора. В качестве слоев автокодировщика используются рекуррентные ячейки *LSTM*.

В качестве слоев автокодировщика, используются рекуррентные ячейки с долгой краткосрочной памятью – *LSTM* и *GRU*.

Свойство рекуррентности позволяет искусственной нейронной сети «обращаться» к результатам своей работы в прошлом, делать анализ предикций. Тем самым контекст решений в будущем будет зависеть не только от первичного глубокого обучения *LSTM*, но и её дальнейшей работы в потоке.

Сети *LSTM* являются подтипом более общих рекуррентных нейронных сетей (*RNN*). Ключевым атрибутом повторяющихся нейронных сетей является их

способность сохранять информацию или состояние ячейки для дальнейшего использования в сети. Это делает их особенно подходящими для анализа временных данных, которые меняются с течением времени. Сети *LSTM* используются в таких задачах, как распознавание речи, перевод текста и, в данном случае, для обнаружения аномалий сети.

LSTM может удалять информацию из состояния ячейки; этот процесс регулируется структурами, называемыми фильтрами (*gates*). Фильтры позволяют пропускать информацию на основании некоторых условий. Они состоят из слоя сигмоидального слоя нейронной сети и операции поточечного умножения. Сигмоидальный слой возвращает числа от нуля до единицы, которые обозначают, какую долю каждого блока информации следует пропустить дальше по сети. Ноль в данном случае означает «не пропускать ничего», единица – «пропустить все».

Алгоритм работы ячейки *LSTM*:

Определяется информация, которую можно удалить из состояния ячейки. Это решение принимает сигмоидальный слой, называемый «слоем фильтра забывания» (*forget gate layer*). Он учитывает значения входного вектора $x_t (x_t \in R^L)$ и выходного вектора $h_{t-1} (h_{t-1} \in R^M)$ и возвращает число от 0 до 1 для каждого числа из состояния ячейки C_{t-1}. 1 означает «полностью сохранить», а 0 – «полностью выбросить» [124, 125].

Расчет производится по следующей формуле:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{4.6}$$

где $h_{t-1} (h_{t-1} \in R^M)$ – выходной вектор нейронной сети на $(t-1)$ шаге; $x_t (x_t \in R^L)$ – входной вектор нейронной сети на t-м шаге; σ – функция активации; $W_f (W_f \in R^{L \times K})$ – матрица весов для входного вектора x_t; $U_f (U_f \in R^{K \times K})$ – матрица весов для выходного вектора h_{t-1} нейронной сети на $(t-1)$ шаге; $b_f (b_f \in R^K)$ – вектор смещения слоя функции забывания.

1. Принимается решение о том, какая новая информация будет храниться в состоянии ячейки. Этот этап состоит из двух частей.

a) Сначала сигмоидальный слой под названием «слой входного фильтра» (*input layer gate*) определяет, какие значения следует обновить:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4.7}$$

b) Затем *tanh*-слой строит вектор новых значений-кандидатов, которые можно добавить в состояние ячейки:

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C) \tag{4.8}$$

2. Обновление старого значения состояния C_{t-1} ячейки на новое состояние C_t.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4.9}$$

3. Генерация выходных данных:

a) С помощью сигмоидального слоя определяется, какая информация из состояния ячейки будет выводиться:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{4.10}$$

b) Значения состояния ячейки проходят через *tanh*-слой, чтобы получить на выходе значения из диапазона от -1 до 1, и перемножаются с выходными значениями сигмоидального слоя, что позволяет выводить только требуемую информацию:

$$h_t = o_t * \tanh(C_t) \tag{4.11}$$

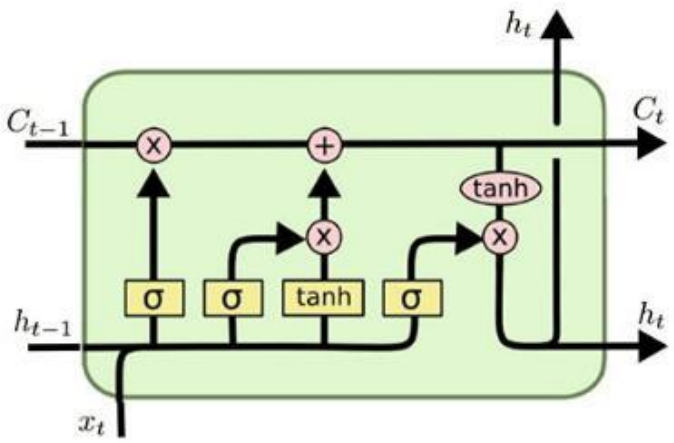

Рисунок 4.4 – Ячейка LSTM

Алгоритм работы ячейки *GRU*:

В управляемом рекуррентном блоке используется функция обновления и функция сброса. Функция обновления отбирает информацию из предыдущих шагов, которая должна быть передана дальше, а функция удаления решает, какую информацию из предыдущих шагов следует удалить.

1. Выходной вектор $z_t (z_t \in R^K)$ функции обновления определяет, какая информация должна хранится в новом состоянии ячейки $C_t (C_t \in R^K)$:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{4.12}$$

где $h_{t-1} (h_{t-1} \in R^M)$ – выходной вектор значений нейронной сети на предыдущем шаге; x_t – входной вектор значений нейронной сети; σ – функция активации; $W_z (W_z \in R^{LxK})$ – матрица весов для входного вектора x_t; $U_z (U_z \in R^{K \times K})$ – матрица весов для выходного вектора h_{t-1} нейронной сети на $(t-1)$ шаге; $b_z (b_z \in R^K)$ – вектор смещения слоя функции обновления.

2. Выходной вектор $r_t (r_t \in R^K)$ функции сброса определяет, какую часть информации необходимо удалить:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{4.13}$$

где $h_{t-1} (h_{t-1} \in R^M)$ – выходной вектор значений нейронной сети на предыдущем шаге; x_t – входной вектор значений нейронной сети; σ – функция активации; $W_r (W_r \in R^{LxK})$ – матрица весов для входного вектора x_t; $U_r (U_r \in R^{K \times K})$ – матрица весов для выходного вектора h_{t-1} нейронной сети на $(t-1)$ шаге; $b_r (b_r \in R^K)$ – вектор смещения слоя функции сброса.

3. Для вычисления выходного вектора $h_t (h_t \in R^K)$ нейронной сети необходимо:

а) найти выходной вектор $h_t^{'} (h_t^{'} \in R^K)$ позволяющий определить какую информацию на $(t-1)$ шаге следует отбросить:

$$h_t^{'} = \tanh(W_h x_t + r_t \cdot U_h h_{t-1} + b_h) \tag{4.14}$$

где $r_t(r_t \in R^K)$ – выходной вектор функции сброса; $h_{t-1}(h_{t-1} \in R^M)$ – выходной вектор нейронной сети на $(t-1)$ шаге; $W_h(W_h \in R^{L \times K})$ – матрица весов для входного вектора x_t; $U_h(U_h \in R^{K \times K})$ – матрица весов для выходного вектора h_{t-1} нейронной сети на $(t-1)$ шаге; $b_h(b_h \in R^K)$ – вектор смещения слоя tanh -слоя.

b) выходной вектор нейронной сети определяется на основе векторов $h_{t-1}(h_{t-1} \in R^M)$, $h_t^{'}(h_t^{'} \in R^M)$ и выходного вектора $z_t(z_t \in R^K)$ функции обновления как:

$$h_t = (1 - z_t) \cdot h_{t-1} + z \cdot h_t^{'} \tag{4.15}$$

где $z_t(z_t \in R^K)$ – выходной вектор функции обновления; $h_{t-1}(h_{t-1} \in R^M)$ – выходной вектор нейронной сети на $(t-1)$ шаге; $h_t^{'}(h_t^{'} \in R^M)$ – выходной вектор нейронной сети на *t*-м шаге.

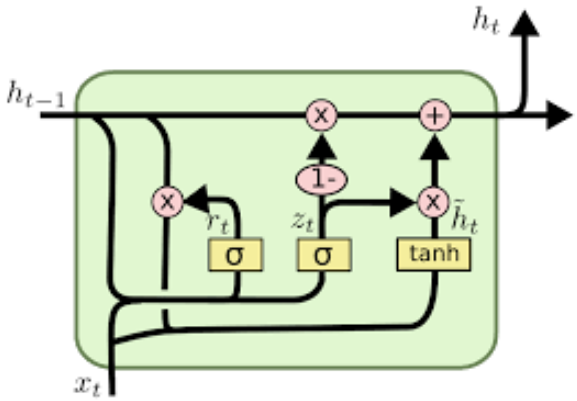

Рисунок 4.5 – Ячейка GRU

Несмотря на то, что *GRU*-сеть эффективнее в вычислительном плане из-за меньшего числа функций, он все равно стоит на втором месте после *LSTM* в плане исполнения. В связи с этим *GRU*-сеть целесообразно использовать тогда, когда необходимо быстро обучить модель при нехватке вычислительных мощностей [125, 138].

4.3 Архитектура системы раннего обнаружения и классификации компьютерных атак в сетевом трафике сети передачи данных

На основании проведенного исследования решений мировых производителей в области разработок систем раннего обнаружения и

классификации КА в сетевом трафике, сформируем общие и специфические требования для СПД.

Общие требования:

1. Деление на уровни;

2. Деление на компоненты;

3. Взаимодействие с внешними системами.

Специфические требования:

1. Возможность функционирования процесса сбора информации;

2. Возможность процесса анализа сетевого трафика и выявления аномалий;

3. Функционирование процесса обнаружения и классификации КА;

4. Поддержка процессов формирования исходных данных, ранжирования контрмер;

5. Возможность выбора и реализации средств противодействия.

Разработанные методики и алгоритмы реализованы в рамках системы раннего обнаружения и классификации КА. Предлагаема архитектура данной системы включает три уровня и восемь компонентов (рис. 4.6):

1. уровень управления:

1.1. компонент сбора информации,

1.2. компонент управления,

1.3. компонент отображения;

2. уровень обнаружения и классификации:

2.1. компонент классификации трафика,

2.2. компонент выявления аномалий,

2.3. компонент обнаружения и классификации КА;

3. уровень противодействия:

3.1. компонент выбора средств противодействия,

3.2. компонент реализации средств противодействия.

Структура системы делится в соответствии с уровнями и компонентами:

1. Уровень управления.

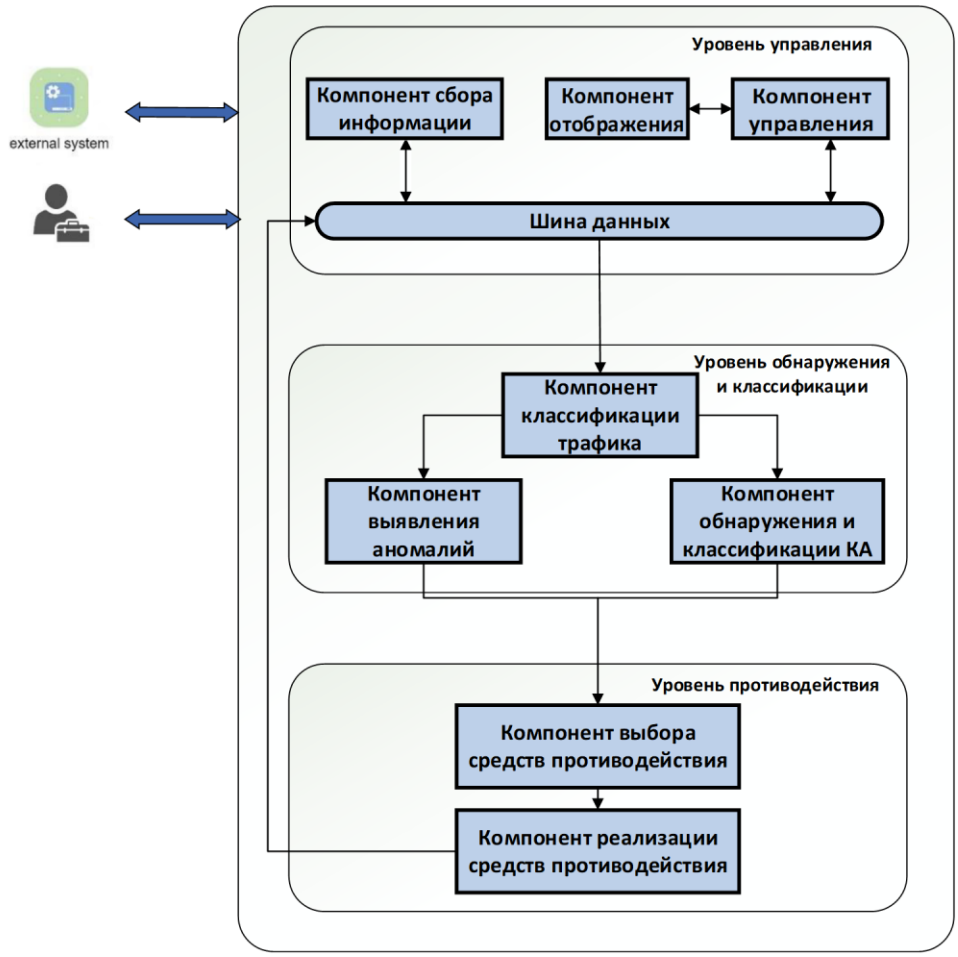

Рисунок 4.6 – Архитектура системы раннего обнаружения и классификации КА в сетевом трафике СПД

Функция сбора информации включает сбор, фильтрацию и предобработку сетевого трафика с помощью библиотеки libpcap. Компонент управления осуществляет управление и настройку чувствительности классификатора, а также количество пакетов, анализируемых с помощью фрактального анализа. Функция отображения осуществляет мониторинг и отображение информации о системе и метрических характеристик всех компонентов системы (включая предупреждения о проблемах).

2. Уровень обнаружения и классификации.

Компонент классификации трафика определяет стационарность сетевого трафика. Функция выявления аномалий обрабатывает нестационарный сетевой трафик с помощью фрактального анализа. Компонент обнаружения и

классификации КА обрабатывает стационарный сетевой трафик с помощью гибридной нейронной сети.

3. Уровень противодействия.

В функцию компонента выбора средств противодействия входят задачи поддержки принятия решения, формирование целей и средств противодействия, а через компонент поддерживается связь с внешними агентами и системами реализации.

4.3.1 Программные прототипы компонентов

Программные прототипы компонентов системы раннего обнаружения

Элементы архитектуры реализованы в качестве программных прототипов: (1) программный прототип компонента сбора информации сетевого трафика, который включает в себя алгоритм выявления степени самоподобия; (2) программный прототип компонента обнаружения аномалий сетевого трафика СПД на основе принципов фрактального анализа данных и методах машинного обучения.

Программный прототип компонента сбора информации сетевого трафика

Компонент состоит из нескольких связанных в комплекс алгоритмов:

1. Алгоритм вычисления флуктуаций.

2. Алгоритм оценки влияния степени самоподобия на систему раннего обнаружения.

На вход алгоритма вычисления степени самоподобия подается массив временного ряда <*data*>, на выходе – среднеквадратичные отклонения от локального тренда (флуктуации), по которым далее оценивается свойство самоподобия. Фрагмент кода алгоритма представлен в листинге на рисунке 4.7.

На вход алгоритма оценки влияния степени самоподобия на систему раннего обнаружения подается длина окна <*scales*>, в котором будет вычисляться среднеквадратичное значение и флуктуации <*fluctuations*>, на выходе – оценка влияния степени самоподия <*coef*>. Фрагмент кода алгоритма представлен в листинге на рис. 4.8.

```
#Шаг 1 Создаем профиль сигнала
def dfa(data, scales=None,  fit_trend="poly"):
  data = np.asarray(data)
  total_N = len(data)
  if scales is None:
    if total_N > 70:
      scales = logarithmic_n(4, 0.1 * total_N, 1.2)
    elif total_N > 10:
      scales = [4, 5, 6, 7, 8, 9]
    else:
      scales = [total_N-2, total_N-1]
      msg = "выберите scales = {}, DFA"
      warnings.warn(msg.format(scales),RuntimeWarning)
  if len(scales) < 2:
    raise  ("необходимо как минимум 2 scales ")
  if np.min(scales) < 2:
    raise ValueError("scales должно быть не менее двух ")
  if np.max(scales) >= total_N:
    raise ValueError("scales не может быть больше, чем размер входного сигнала")
  walk = np.cumsum(data - np.mean(data))
  fluctuations = []
  for n in scales:
    assert n >= 2
    #Шаг 2 Разделить данные на фрагменты размером n
    if overlap:
      d = np.array([walk[i:i + n] for i in range(0, len(walk) - n, n // 2)])
    else:
      d = walk[:total_N - (total_N % n)]
      d = d.reshape((total_N // n, n))
    #Шаг 3 Вычислять локальный тренд как полином
    x = np.arange(n)
    tpoly = [poly_fit(x, d[i], order, fit=fit_trend)
          for i in range(len(d))]
    tpoly = np.array(tpoly)
    trend = np.array([np.polyval(tpoly[i], x) for i in range(len(d))])
    #Шаг 4 Расчёт стандартного отклонения (флуктуации)
    flucs = np.sqrt(np.sum((d - trend) ** 2, axis=1) / n)
    f_n = np.sum(flucs) / len(flucs)
    fluctuations.append(f_n)
  fluctuations = np.array(fluctuations)
  nonzero = np.where(fluctuations != 0)
  scales = np.array(scales)[nonzero]
  fluctuations = fluctuations[nonzero]
  if len(fluctuations) == 0:
    poly = [np.nan, np.nan]
  else:
    poly = poly_fit(np.log(scales), np.log(fluctuations))
```

Рисунок 4.7 – Фрагмент кода алгоритма вычисления степени самоподобия

```
# Шаг 1 инициализация регрессионной модели
def poly_fit(x, y):
    model = sklin.RANSACRegressor(sklin.LinearRegression(fit_intercept=False))
    xdat = np.asarray(x)
    if len(xdat.shape) == 1:
      xdat = xdat.reshape(-1, 1)
# Шаг 2 оценивание с помощью полиномиальной регрессии
    polydat = PolynomialFeatures(degree=1).fit_transform(xdat)
    try:
      model.fit(polydat, y)
      coef = model.estimator_.coef_[::-1]
    except ValueError:
      warnings.warn(
        "ошибка RANSAC "
        + "использовать numpy's polyfit",
        RuntimeWarning)
      coef = np.polyfit(x, y)
    return coef
```

Рисунок 4.8 – Фрагмент кода алгоритма оценки влияния степени самоподобия на систему раннего обнаружения

Программный компонент написан на языке *Python*, версия 3.8. В компоненте используются внешние модули из библиотек языка программирования: *scikit-learn*, *numpy*, а также компонент реализован с помощью *framework Django*.

Программный прототип компонента обнаружения аномалий сетевого трафика СПД

Компонент состоит из двух независимых алгоритмов и работает в двух режимах:

1. Алгоритм фрактального анализа данных и нейронной гибридной сети;

2. Алгоритм ранжирования аномалий по степени опасности на основе искусственного интеллекта.

Алгоритм фрактального анализа данных и методах машинного обучения позволяет формировать наборы исходных данных для исследований и обучения гибридной нейронной сети. Фрагмент кода алгоритма представлен в листинге на рисунке 4.9.

```
#Шаг 1 Минимизация разницы градиентов
class MultiAutoencoder(tf.keras.Model):
      def train_step(inputs, targets):
            loss_objects = ["categorical_crossentropy","categorical_crossentropy"]
             with tf.GradientTape() as tape:
           outputs = model(inputs)
           losses = [l(t, o) for l,o,t in zip(loss_objects, outputs, targets)]
       gradients = tape.gradient(losses, model.trainable_variables)
       print(gradients)
       op.apply_gradients(zip(gradients, model.trainable_variables))
       return outputs
   #Шаг 2 Создание ветки автокодировщика
   def decoder_category_branch(encoder_GRU):
       decoded = RepeatVector(max_len_str, name = 'bottleneck')(encoder_GRU)
       decoder_BLSTM = Bidirectional(LSTM(25, return_sequences=True, name='Decoder
BLSTM'))(decoded)
       decoder_drop = Dropout(0.1)(decoder_BLSTM)
       decoder_GRU = GRU(100, activation='tanh', return_sequences=False,
name='Decoder-GRU')(decoder_drop)
       decoder_output = Dense(max_len_str, activation='softmax', name='decoder-
output')(decoder_GRU)
       return decoder_output
#Шаг 3 Создание ветки классификатора
   def encoder_binary_branch(emb_norm):
       encoder_BLSTM = Bidirectional(LSTM(25, activation='tanh',
return_sequences=True, name='Encoder-LSTM'))(emb_norm)
       encoder_drop = Dropout(0.1)(encoder_BLSTM)
       encoder_GRU = GRU(25, activation='tanh', return_sequences=False, name =
'bottleneck-1')(encoder_drop)
       encoder_dense_1 = Dense(20, activation='relu')(encoder_GRU)
       encoder_dense_2 = Dense(10, activation='relu')(encoder_dense_1)
       encoder_dense_3 = Dense(8, activation='relu')(encoder_dense_2)
       encoder_output = Dense(6, activation='softmax', name = 'encoder-
output')(encoder_dense_3)
       return encoder_GRU, encoder_output
#Шаг 4 объединение веток нейронной сети
   def build():
       encoder_inputs = Input(shape=(max_len_str,), name='Encoder-Input')
       emb_layer = Embedding(input_dim=len(word_index)+1, output_dim=100,
input_length = max_len_str, name='Embedding', mask_zero=False)(encoder_inputs)
       emb_drop = SpatialDropout1D(0.1)(emb_layer)
       emb_norm = BatchNormalization()(emb_drop)
       encoder_GRU, encoder_output =
MultiAutoencoder.encoder_binary_branch(emb_norm)
       decoder_output = MultiAutoencoder.decoder_category_branch(encoder_GRU)
       model = Model(inputs=encoder_inputs, outputs=[encoder_output, decoder_output],
name="MultiAutoencoder")
       return model
```

Рисунок 4.9 – Фрагмент кода алгоритма гибридной нейронной сети

Алгоритм ранжирования аномалий выявляет известные и неизвестные КА в реальном масштабе времени, вырабатывать решения по реализации средств защиты, уведомлять администратора о степени опасности КА. Фрагмент кода алгоритма представлен в листинге на рисунке 4.10.

```
def my_middleware(get_response):
#Шаг 1 инициализация слоя перехватчика событий системы
    def middleware(request):
    # Код выполняется при каждом запросе до передачи в обработчик.
        if not hasattr(request, 'ataked'):
            request.ataked = False
        request_header = dict(request.headers)
        request_middle = '<START>' + str(request.method) + ' ' +
str(request.path_info) + request.body.decode('utf-8') + '<STOP>'

        if regex.search(r'\csrfmiddlewaretoken=.+?&', request_middle):
            request_middle = regex.sub(r'\csrfmiddlewaretoken=.+?&', '',
request_middle)

        X = tokenizer.texts_to_sequences([urllib.parse.unquote(request_middle)])
        X_pad = pad_sequences(X, maxlen=max_len_str, padding='post')
        # Шаг 2: Ранжирование аномалий
         model.predict(X_pad)[0].sort())
```

Рисунок 4.10 – Фрагмент кода алгоритма ранжирования аномалий по степени опасности на основе искусственного интеллекта

Таким образом, программный прототип компонента обнаружения аномалий сетевого трафика СПД, включающий алгоритм фрактального анализа данных и методах машинного обучения и алгоритм ранжирования аномалий по степени опасности на основе искусственного интеллекта, позволяет сформировать список аномалий по степени опасности, а также возможных средств противодействия.

Программный компонент также написан на языке *Python* 3.8. В нем используются внешние модули из библиотек языка программирования: *numpy*, *tensorflow*, *keras*, *pandas* [139].

4.4 Экспериментальная и теоретическая оценка методик обнаружения аномалий и классификации компьютерных атак в сетевом трафике сети передачи данных

Для формирования датасета, который годился бы для обучения предложенной архитектуры нейронной сети и оценки ее возможности обнаруживать аномалии, необходимо было сначала смоделировать КА обхода *WAF*.

Общая идея нахождения способов обхода *WAF* заключается в приведении запроса к виду, в котором он остается понятным для атакуемого веб-приложения, но при этом является не понятным или кажется безобидным для *WAF*. Для этой цели могут использоваться различные спецсимволы, которые могут нарушать логику работы *WAF* и при этом быть понятными серверу.

На рис. 4.11 представлен пример обычного запроса и ответа на него сервера. Запрос представляет собой попытку передать методом *POST* параметр «*select*». Перехват запросов осуществлялся с помощью программы *Burp Suite*, предназначенной для выполнения тестов по безопасности веб-приложений [124].

Как видно из рис. 4.11 параметр «*select*» заблокирован *WAF*. На рис 4.12 представлен другой запрос, в котором параметр «*select*» закодирован спецсимволами, взятыми из кодировочной таблицы *Unicode*.

Рисунок 4.11 – Попытка отправить запрос без преобразования параметра

Рисунок 4.12 – Попытка отправить запрос с кодированием параметра

Как видно из рис. 4.12, если передать параметр «*select*» в кодировке *Unicode*, то *WAF* его не обнаруживает.

Передадим теперь на сервер таким же способом (то есть с использованием кодировки *Unicode*) параметр «*union select sleep*(10)#» (рис. 4.13). Запрос также успешно обошел *WAF* и поступил на обработку на сервер.

Рисунок 4.13 – Пример обхода *WAF* зашифрованным запросом

Запрос, который был направлен на обработку на сервер, выглядит следующим образом:

SELECT note FROM notes WHERE assignee = «union select sleep(10)#».

Сервер вернул ответ на запрос с задержкой в 10 секунд. Тем самым была реализована эффективная кибератака вида «*time base sql injection*». Пример реализации этой кибератаки, осуществляющей обход *WAF*, представлен на рис. 4.14.

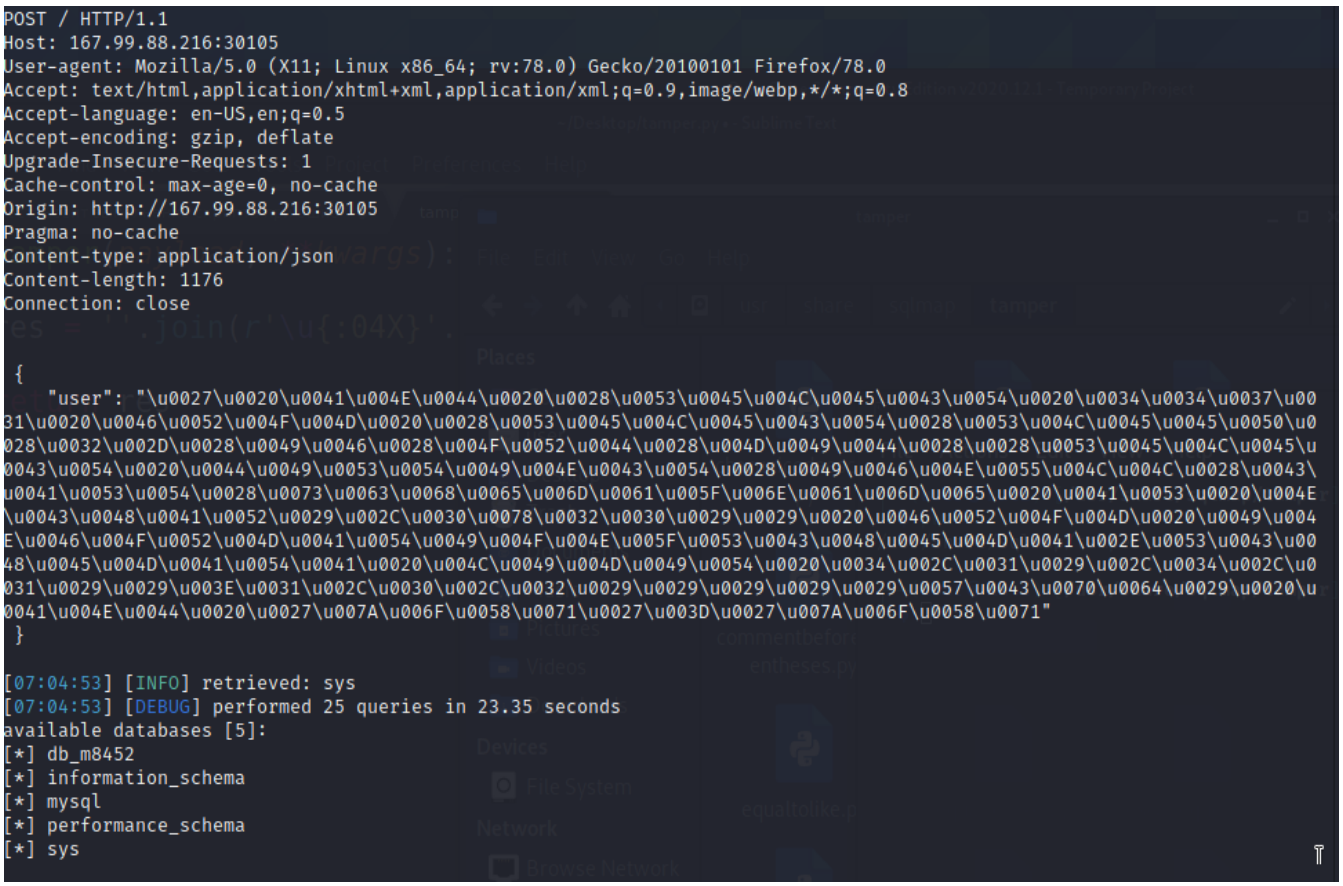

Рисунок 4.14 – Пример кибератаки обхода *WAF*

Существуют интеллектуальные *WAF*, основанные на методах машинного обучения, однако у них агрессивный режим работы и большое количество ложных срабатываний [23].

Программная реализация методики раннего обнаружения КА в сетевом трафике сетей передачи данных разработана на языке программирования *Python* с использованием библиотеки *Pandas*, с помощью которой осуществлялись обработка и анализа данных. Библиотека *Pandas* написана на языках программирования *Cu*, *Cython*, и *Python*. Она делает *Python* мощным инструментом для анализа данных и дает возможность на высоком уровне строить сводные таблицы, выполнять группировки, предоставлять удобный доступ к табличным данным.

Кроме того, кроме библиотеки *Pandas* использовалась библиотека *NumPy*, которая представляет собой инструмент более низкого уровня, обеспечивающий работу с высокоуровневыми математическими функциями, а также с многомерными массивами (тензорами).

Графики строились с помощью модуля *Matplotlib* на основе полученного набора данных. Все расчеты производились в интегрированной среде разработки *Jupiter notebook*.

На рисунке 4.15 изображена блок-схема отражающая этапы формирования данных и обучения гибридной модели нейронной сети.

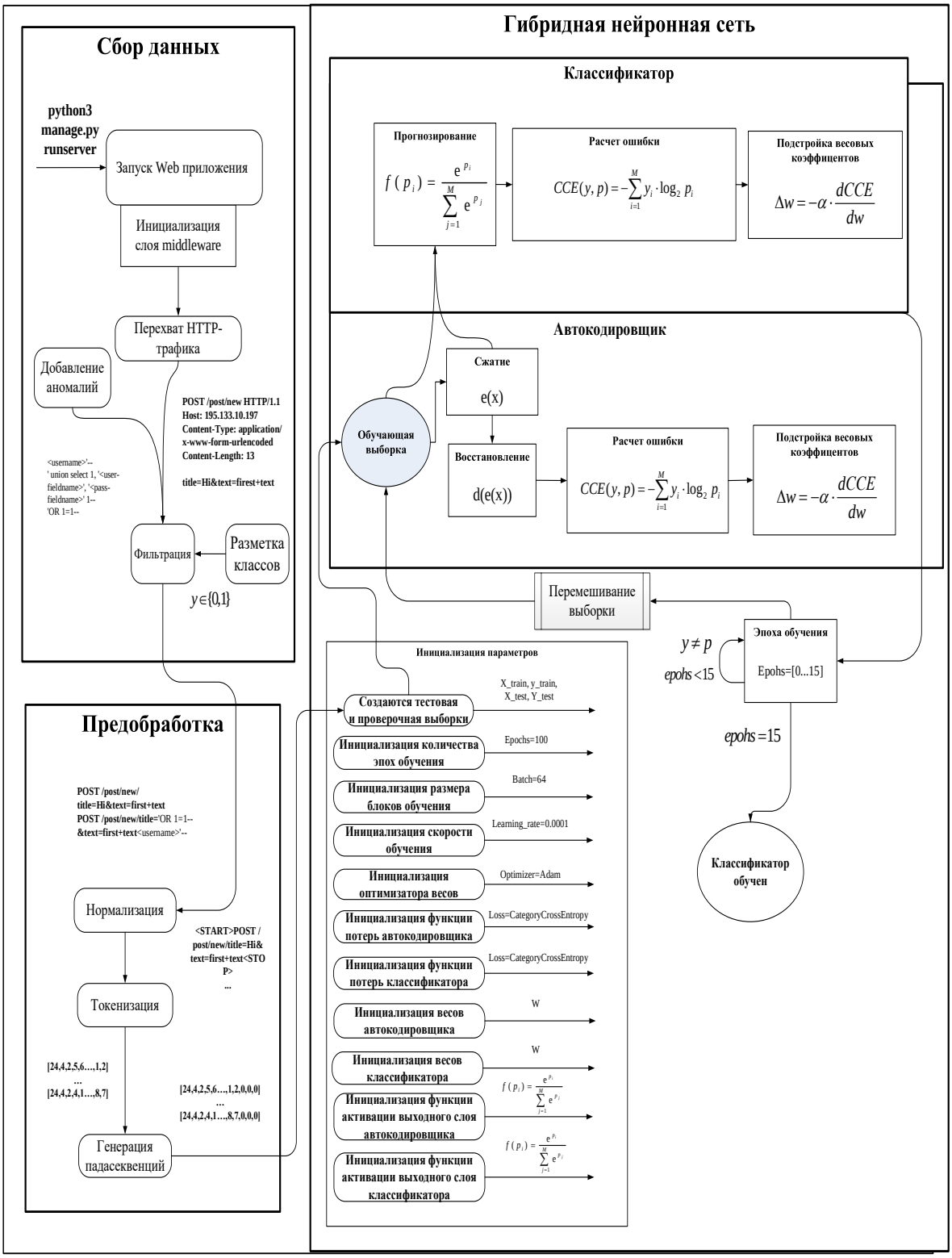

Рисунок 4.15 – Блок-схема этапов обучения гибридной нейронной сети

Для того, чтобы сформировать датасет данных предназначенных для обучения нейронной сети, на языке программирования *python* реализовано web-приложение способное перехватывать любые пользовательские запросы с помощью промежуточного слоя *middleware*. Такой подход позволяет обрабатывать запросы из браузера, прежде чем они, достигнут представления *Django* (сервера), а также ответы от представлений до того, как они возвращаются в браузер.

Следующий этап обучения гибридной нейронной сети включает в себя нормализацию данных (рис. 4.16, 4.17). Запросы оборачиваются специальными токенами <*START*> и <*STOP*>, что задаёт верное вероятностное распределение над последовательностями разной длины [124, 140-146].

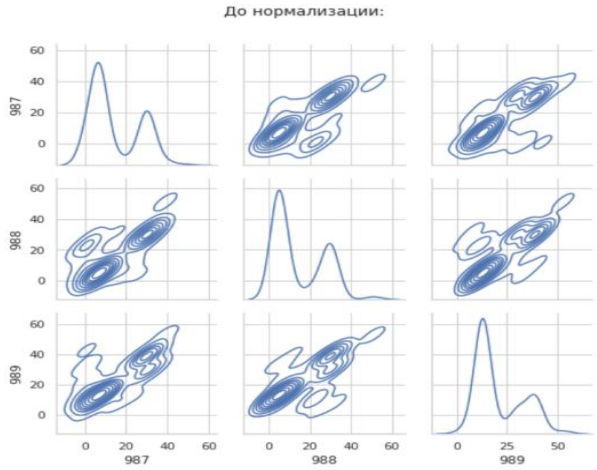

Рисунок 4.16 – Распределения последовательностей разных длин до нормализации

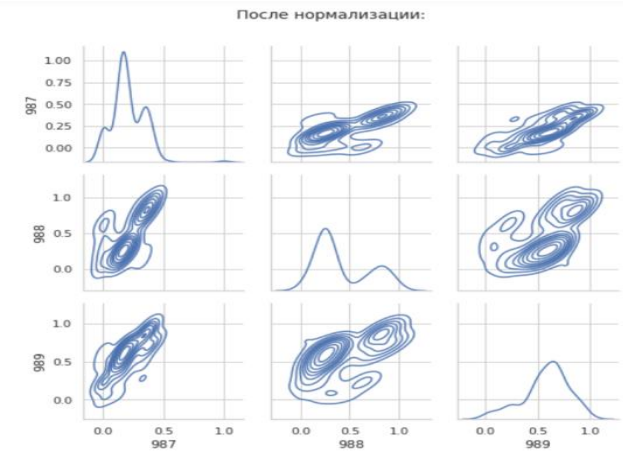

Рисунок 4.17 – Распределения последовательностей разных длин после нормализации

В [126] изучалось распределение расстояний (т.е количество символов) между одинаковыми буквами в тексте. Оказалось, что это квазистационарный ряд, одинаково распределенный (по гамма-распределению) для любой буквы алфавита, и близкий к белому шуму. Исходя из этого, для проверки стационарности *HTTP*-трафика, проведен эксперимент, который заключался в построении графика распределения длин между двумя одинаковыми символами (рис. 4.18) и оценки стационарности получившегося ряда с помощью теста Дики-Фуллера.

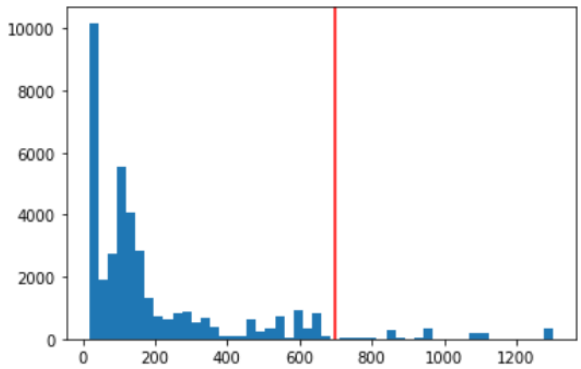

Рисунок 4.18 – Распределение длин легитимных запросов

Далее произведена предобработка и нормализация получившейся выборки. Поскольку протокол *HTTP* – текстовый протокол, использовалось векторное представление символов. Для этого сперва осуществляется замена символов, встречающихся в датасете на числовой эквивалент, который не имеет самостоятельного смысла/значения для внешнего или внутреннего использования (токенизировать), а затем переводятся слова в последовательность секвенций (рис. 4.19), с помощью токенизации.

```
data_train = pad_sequences(X_train_sequences, maxlen=max_len_str, padding='post')
data_test = pad_sequences(X_test_sequences, maxlen=max_len_str, padding='post')
```

```
data_train
```

```
array([[24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0],
       ...,
       [24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0],
       [24,  4,  2, ...,  0,  0,  0]], dtype=int32)
```

Рисунок 4.19 – Секвенции

При этом, важным обстоятельством является то, что все секвенции должны быть одной длины. Если запрос меньше длины секвенции, то оставшиеся символы заполняются нулями.

Далее секвенции подаются на вход гибридной нейронной сети, и подбираются гиперпараметры. Но перед этим, необходимо настроить среду выполнения таким образом, чтобы все вычисления происходили на *GPU*. Для этой цели достаточно установить библиотеки, строго определенных версий:

conda create --name tf python=3.8

conda activate tf

conda install cudatoolkit=10.0.130

conda install cudnn=7.6.0=cuda10.0_0

pip install --upgrade tensorflow-gpu

sudo apt install libcudnn8

Подбор параметров осуществляется таким образом, чтобы функция потерь при обучении автокодировщика уменьшалась, при этом точность классификатора росла.

```
Trial 27 Complete [00h 27m 58s]
decoder-output_loss: 19800.93359375

Best decoder-output_loss So Far: 19800.93359375
Total elapsed time: 05h 00m 36s

Search: Running Trial #28

Hyperparameter    |Value             |Best Value So Far
decoder-output    |0.0014            |0.0039
encoder-output    |90                |45
learning_rate     |1e-06             |1e-05
tuner/epochs      |10                |10
tuner/initial_e...|0                 |0
tuner/bracket     |0                 |0
tuner/round       |0                 |0

Epoch 1/10
1351/1351 [==============================] - 218s 156ms/step - loss: 92.7079 - encoder-
Epoch 2/10
1351/1351 [==============================] - 208s 154ms/step - loss: 92.4347 - encoder-
Epoch 3/10
1026/1351 [=====================>........] - ETA: 43s - loss: 92.1161 - encoder-output_
```

Рисунок 4.20 – Подбор гиперпараметров нейронной сети

В процессе обучения гибридной нейронной сети на входной слой поступают вектора (рис. 4.21) (падасеквенции, секвенции или эмбеддинги) в зависимости от реализации.

Секвенция – пронумерованный набор объектов, среди которых допускаются повторения, причём порядок объектов имеет значение. Нумерация происходит натуральными числами. Падасеквенция – функция библиотеки *Tensorflow*, которая используется для того, чтобы все последовательности в списке имели одинаковую длину.

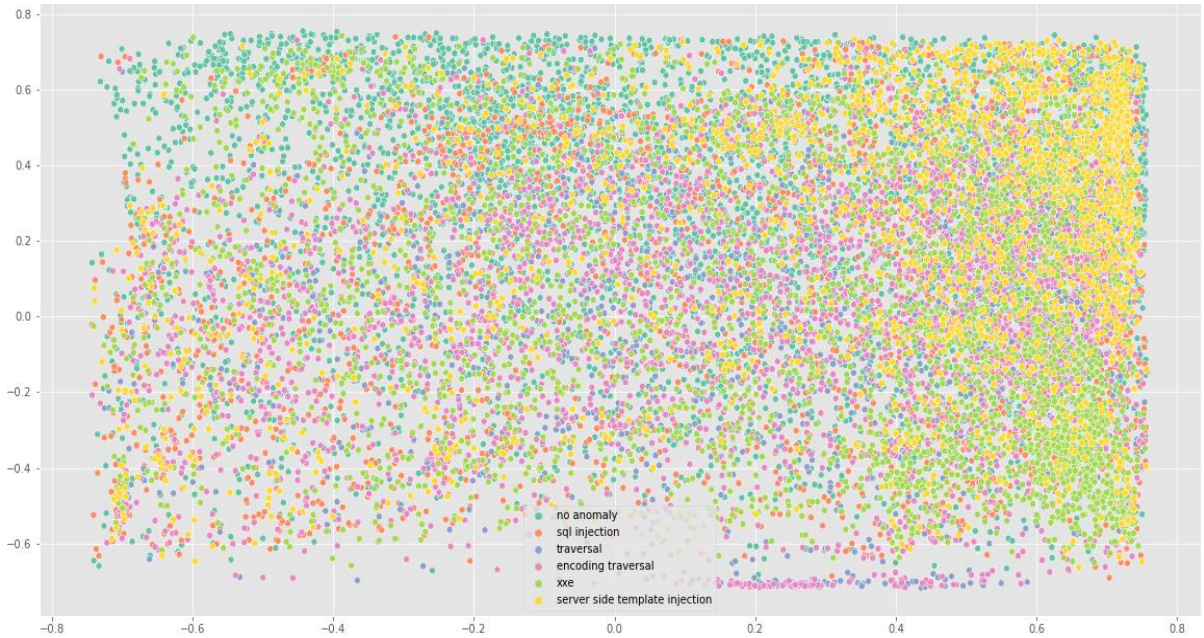

Рисунок 4.21 – Визуальное представление векторного отображения данных, подаваемых на вход гибридной нейронной сети

По умолчанию это делается путем добавления 0 в начало каждой последовательности, пока каждая последовательность не будет иметь ту же длину, что и самая длинная последовательность. А для преобразования положительных целых чисел (индексы) в плотные (*dense*) векторы фиксированного размера применяется первый слой нейронной сети - *Embedding*.

Эксперименты показали, что наилучший результат дают эмбеддинги, но они сильно нагружают электронные вычислительные машины, поэтому для быстрых вычислений в исследованиях применялись падосеквенции длиной 120 символов.

В середине гибридной нейронной сети число нейронов уменьшается до 30. Это приводит к потере информации, так как из 120 нейронов не вся информация попадает на 30 нейронов. Поэтому нейронная сеть учится отбрасывать лишнюю (избыточную) информацию (шум), сохраняя, по ее мнению, самую важную.

У классификатора (правая ветвь) на последнем слое имеется шесть нейронов. Они соответствуют шести размеченным классам, упомянутым выше. В случае если классификатор не может отнести данные ни к одному из классов с вероятностью больше 0.6, то такой запрос помечается подозрительным (считается атакой нулевого дня).

У автоэнкодера (левая ветвь) на последнем слое 120 нейронов (столько же, сколько и на входном). Он пытается воссоздать информацию из 30 нейронов к первоначальному виду (ту, что была на 120 входных нейронах). Каждый раз в процессе обучения у автокодировщика получается делать это все лучше и лучше.

Таким образом, уменьшается ошибка, а в середине слоя (там, где 30 нейронов) сохраняется только самая важная информация, из которой можно восстановить информацию в исходном виде – «скрытые латентные представления» (рис. 4.22).

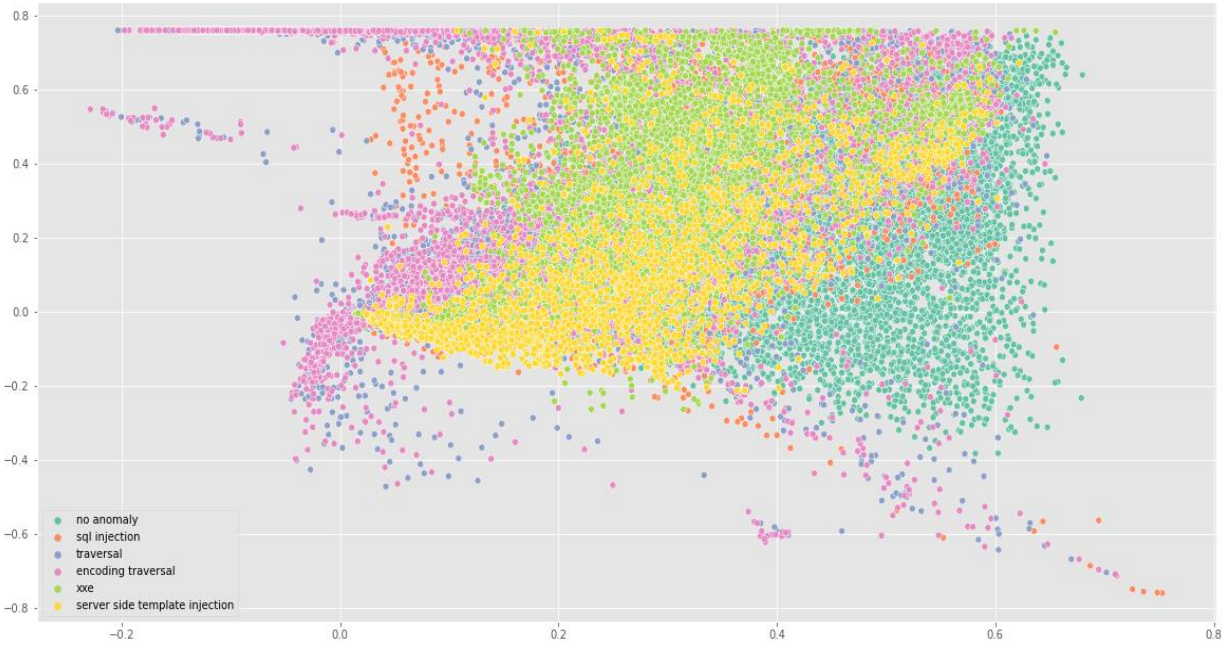

Рисунок 4.22 – Скрытые латентные представления полученные в результате сжатия информации автокодировщиком

Такие представления позволяют классификатору находить дополнительные закономерности в данных, что существенно уменьшает ложные срабатывания и повышает вероятность обнаружения атак нулевого дня.

Для эмпирического оценивания обобщающей способности модели

нейронной сети, использовалась 10-кратный стратифицированный кросс-валидатор *K-Folds* на независимых данных с наиболее равномерным использованием имеющихся данных (рис. 4.23).

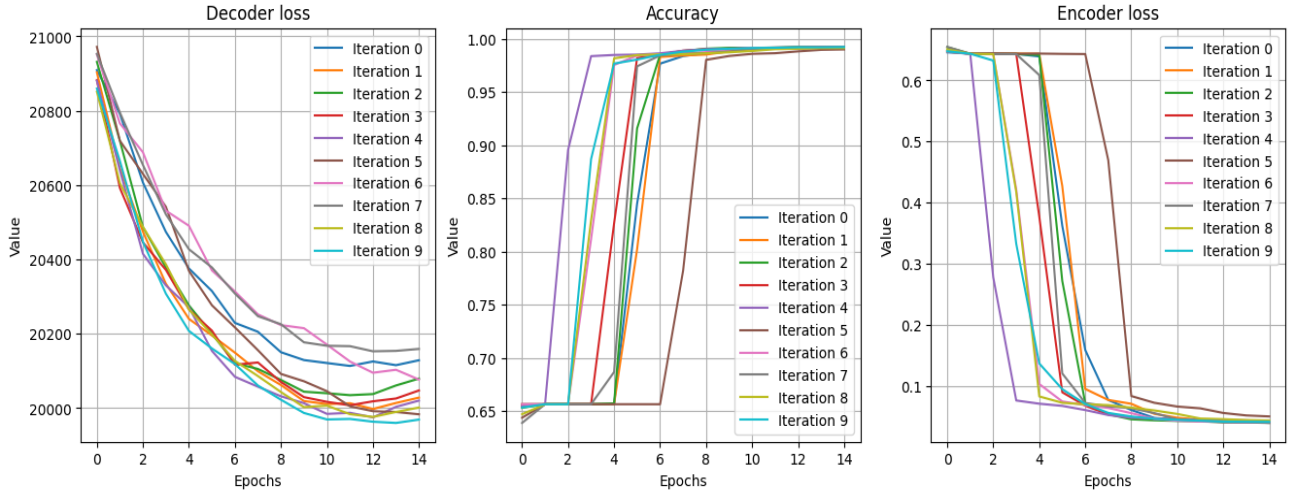

Рисунок 4.23 — Обучение декодера и классификатора на 15 эпохах в 10 итераций с перемешиванием данных

После обучения нейронной сети, была проведена экспериментальная оценка точности и полноты (рис. 4.24) с применением ROC-кривой на данных, не участвующих в обучении модели. Визуальный анализ подтвердил высокую точность предложенного подхода, с минимальным числом ложных срабатываний.

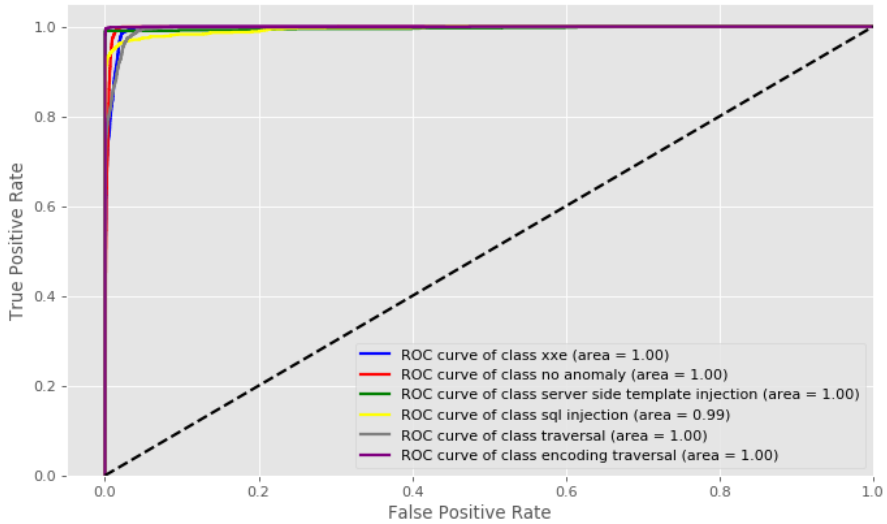

Рисунок 4.24 – *ROC*-кривая для мультиклассовой классификации

Для более качественной оценки выделены главные метрики классификации (рис 4.25).

Из рисунка 4.25 видно, что алгоритм не только справляется с классификацией КА, но также с высокой долей вероятности обнаруживает легитимные запросы «*no anomaly*».

	precision	recall	f1-score	support
xxe	0.83	0.95	0.88	1292
no anomaly	0.99	1.00	0.99	9266
server side template injection	1.00	0.99	0.99	1265
sql injection	0.96	0.93	0.95	1271
traversal	0.94	0.80	0.86	1263
encoding traversal	0.99	0.99	0.99	1305
accuracy			0.97	15662
macro avg	0.95	0.94	0.94	15662
weighted avg	0.97	0.97	0.97	15662

Рисунок 4.25 – Главные метрики классификаций

Это само по себе является минимизацией ложных срабатываний (ошибок первого рода), связанных с отнесением легитимных запросов к КА и наоборот.

Таким образом классификация КА подтверждает отсутствие ложных срабатываний при мультиклассовой классификации аномалий.

Кроме того, для оценки эффективности текущего подхода проведен сравнительный анализ гибридной нейронной сети с методами машинного обучения, основанными на мультиклассовой классификации: логистическая регрессия, дерево принятия решений, градиентный бустинг, линейный дискриминантный анализ, искусственные нейронные сети, квадратичный дискриминантный анализ.

Оценка эффективности методов машинного обучения производилась с помощью трех метрик:

Точность – Определяет долю правильно предсказанных положительных объектов среди всех объектов, предсказанных положительным классом. Чем меньше ложноположительных срабатываний (ошибка первого рода) будет допускать модель, тем больше будет точность.

Полнота – Показывает долю правильно найденных положительных объектов среди всех объектов положительного класса. Чем меньше ложно отрицательных срабатываний (ошибка второго рода), тем выше полнота модели.

F-мера – среднее гармоничное скомпонованной пары точность-полнота.

Логистическая регрессия не справляется с распознаванием инъекций внешних сущностей XML и уязвимостями Path Traversal, об это свидетельствует низкая оценка f1-score (рисунок 4.26).

	precision	recall	f1-score	support	classifiers
xxe	0.579972	0.444089	0.503016	939.000000	Logistic Regression
no anomaly	0.903699	0.999436	0.949159	7089.000000	Logistic Regression
server side template injection	0.986715	0.857293	0.917462	953.000000	Logistic Regression
sql injection	0.923077	0.581498	0.713514	908.000000	Logistic Regression
traversal	0.552239	0.558190	0.555198	928.000000	Logistic Regression
encoding traversal	0.980000	0.895699	0.935955	930.000000	Logistic Regression
accuracy	0.868137	0.868137	0.868137	0.868137	Logistic Regression
macro avg	0.820950	0.722701	0.762384	11747.000000	Logistic Regression
weighted avg	0.864330	0.868137	0.860543	11747.000000	Logistic Regression

Рисунок 4.26 – Оценка эффективности метода машинного обучения «Логистическая регрессия» при распознавании различных КА

Кроме того, при обнаружении атак типа sql injection, у логистической регрессии большое количество ложно-отрицательных срабатываний (из-за маленького значения полноты).

Дерево принятия решений является эффективным инструментом интеллектуального анализа данных и предсказательной аналитики. Модель машинного обучения гораздо лучше справилась с распознаванием компьютерных атак (рис 4.27), чем логистическая регрессия.

	precision	recall	f1-score	support	classifiers
xxe	0.800439	0.777423	0.788763	939.000000	Decision Tree
no anomaly	0.987484	0.990549	0.989014	7089.000000	Decision Tree
server side template injection	0.975967	0.980063	0.978010	953.000000	Decision Tree
sql injection	0.913930	0.888767	0.901173	908.000000	Decision Tree
traversal	0.778714	0.796336	0.787427	928.000000	Decision Tree
encoding traversal	0.970053	0.975269	0.972654	930.000000	Decision Tree
accuracy	0.948242	0.948242	0.948242	0.948242	Decision Tree
macro avg	0.904431	0.901401	0.902840	11747.000000	Decision Tree
weighted avg	0.948040	0.948242	0.948104	11747.000000	Decision Tree

Рисунок 4.27 – Оценка эффективности метода машинного обучения «Дерево принятия решений» при распознавании различных КА

Однако у данного подхода сохраняется высокий уровень возникновения ошибок второго рода при обнаружении XXE и уязвимостями Path Traversal.

Следует отметить, что дерево принятия решений чувствительны к шумам во входных данных и небольшие изменения обучающей выборки могут привести к глобальным корректировкам модели, что скажется на смене правил классификации и интерпретируемости модели. Поэтому для получения хорошего результата, следует уделять особое внимание формированию обучающего датасета.

Градиентный бустинг (рис. 4.28) оказался самым непригодным алгоритмом машинного обучения для мультиклассовой классификации компьютерных атак из-за крайней чувствительности к выбросам.

	precision	recall	f1-score	support	classifiers
xxe	0.498494	0.352503	0.412976	939.000000	Gradient Boosting
no anomaly	0.791183	1.000000	0.883420	7089.000000	Gradient Boosting
server side template injection	0.409712	0.911857	0.565387	953.000000	Gradient Boosting
sql injection	0.000000	0.000000	0.000000	908.000000	Gradient Boosting
traversal	0.500000	0.001078	0.002151	928.000000	Gradient Boosting
encoding traversal	0.000000	0.000000	0.000000	930.000000	Gradient Boosting
accuracy	0.705712	0.705712	0.705712	0.705712	Gradient Boosting
macro avg	0.366565	0.377573	0.310655	11747.000000	Gradient Boosting
weighted avg	0.590043	0.705712	0.612169	11747.000000	Gradient Boosting

Рисунок 4.28 – Оценка эффективности метода машинного обучения «Градиентный бустинг» при распознавании различных КА

Поэтому перед обучением данной модели применялся алгоритм поиска наиболее подходящих гиперпараметров по сетке и подбор ансамбля моделей.

Линейный дискриминантный анализ справился только с распознаванием не аномальных образов, рисунок 4.29.

	precision	recall	f1-score	support	classifiers
xxe	0.457529	0.252396	0.325326	939.000000	Linear DA
no anomaly	0.959525	0.979828	0.969570	7089.000000	Linear DA
server side template injection	0.834297	0.908709	0.869915	953.000000	Linear DA
sql injection	0.803077	0.574890	0.670090	908.000000	Linear DA
traversal	0.478916	0.685345	0.563830	928.000000	Linear DA
encoding traversal	0.839836	0.879570	0.859244	930.000000	Linear DA
accuracy	0.853409	0.853409	0.853409	0.853409	Linear DA
macro avg	0.728863	0.713456	0.709662	11747.000000	Linear DA
weighted avg	0.849702	0.853409	0.846051	11747.000000	Linear DA

Рисунок 4.29 – Оценка эффективности метода машинного обучения «Линейный дискриминантный анализ» при распознавании различных КА

Поэтому в данном случае, наиболее подходящее применение линейного дискриминантного анализа является – бинарная классификация, отображающая есть в выборке аномалия или ее нет.

Многослойный персептрон, хуже всех справился с задачей мультиклассовой классификации КА (рис. 4.30).

	precision	recall	f1-score	support	classifiers
xxe	0.498462	0.172524	0.256329	939.000000	Neural Net
no anomaly	1.000000	0.080971	0.149811	7089.000000	Neural Net
server side template injection	0.552133	0.733473	0.630014	953.000000	Neural Net
sql injection	0.116336	1.000000	0.208424	908.000000	Neural Net
traversal	0.496416	0.298491	0.372813	928.000000	Neural Net
encoding traversal	0.484003	0.634409	0.549093	930.000000	Neural Net
accuracy	0.273261	0.273261	0.273261	0.273261	Neural Net
macro avg	0.524558	0.486645	0.361081	11747.000000	Neural Net
weighted avg	0.774638	0.273261	0.251041	11747.000000	Neural Net

Рисунок 4.30 – Оценка эффективности метода машинного обучения «Искусственные нейронные сети» при распознавании различных КА

Это неудивительно, т.к персептрон классифицирует только линейно разделимые объекты (то есть только такие множества векторов, между которыми можно провести разделяющую гиперплоскость).

Квадратичный дискриминантный анализ (рис. 4.31), как и линейный дискриминантный анализ плохо справился с выявлением КА типа Path Traversal.

	precision	recall	f1-score	support	classifiers
no anomaly	0.882097	0.999436	0.937107	7089.000000	Quadratic DA
server side template injection	0.988462	0.809024	0.889786	953.000000	Quadratic DA
sql injection	0.929066	0.591410	0.722746	908.000000	Quadratic DA
traversal	0.548456	0.554957	0.551687	928.000000	Quadratic DA
encoding traversal	0.997382	0.819355	0.899646	930.000000	Quadratic DA
accuracy	0.855538	0.855538	0.855538	0.855538	Quadratic DA
macro avg	0.821084	0.696478	0.746343	11747.000000	Quadratic DA
weighted avg	0.853061	0.855538	0.846514	11747.000000	Quadratic DA

Рисунок 4.31 – Оценка эффективности метода машинного обучения «Квадратичный дискриминантный анализ» при распознавании различных КА

Большое количество ложно-отрицательных срабатываний при обнаружении sql injection (маленькая полнота).

Алгоритмы плохо справляются с обнаружением КА типа: инъекций внешних сущностей xxe и traversal. Кроме того, у методов машинного обучения большое количество ложных срабатываний, особенно у классификаторов: квадратичный дискриминантный анализ, искусственные нейронные сети, линейный дискриминантный анализ, градиентный бустинг и логистическая регрессия.

Кроме этого, сравнительная оценка эффективности рассматриваемой системы была проведена на основе его сравнения с другими известными системами, например, *IDS* и *IPS*, которые при обнаружений компьютерных атак используют сигнатурный метод, статистический метод и методы машинного обучения. Результаты такого сравнения представлены в таблице 4.1.

Таблица 4.1 – Результаты сравнительного анализа известных методов обнаружения кибератак

Наименование методов	Скорость обнаружения (сек)	Точность обнаружения КА		Ложные срабатывания		Тип трафика	
		Известных	Неизвестных	Бинарная классификация	Мультиклассовая классификация	Стационарный	Нестационарный
Сигнатурные методы	5	0,99	0,5	-	-	+	-
Статистические методы	30	0,92	0,6	-	-	+	-
Методы машинного обучения	28	0,72-0,97	0,6	0,39-0,56	0,06-0,92	+	-
Разработанные методики	5	0,96	0,8	0,05	0-0,20	+	+

В качестве основных учитываемых параметров сравниваемых методов рассматривались скорость и точность обнаружения кибератак, как известных, так и неизвестных, возможность работы со стационарным и нестационарным трафиком, а также то, насколько часто происходят ложные срабатывания.

Из таблицы 4.1 видно, что разработанные методики по скорости не уступают самым лучшим известным подходам (сигнатурным методам), по точности превышают все известные методы, а также имеют более широкую

область применения (охватывают нестационарный трафик), т.е. можно сделать вывод, что эффективность обнаружения КА повысилась.

Сигнатурные методы используют заранее составленные правила. Поэтому они имеют высокую точность обнаружения известных типов кибератак. Однако они не способны обнаруживать новые, неизвестные типы атак, включая таргетированные атаки и являются медлительными из-за ограничений чтения информации с жесткого диска. Кроме того, они имеют низкие показатели по отсутствию ложного обнаружения.

Статистические методы используют накопленную статистику. По этой причине они не отличаются от сигнатурных методов по скорости обнаружения, но являются менее точными в обнаружении известных атак. В то же время в ряде случаев они способны обнаруживать неизвестные атаки. По ложному обнаружению они перспективнее, чем сигнатурные методы.

Методы машинного обучения в настоящее время являются достаточно разнообразными и хорошо развитыми. Несмотря на то, что в этих методах процессу обнаружения атак обязательно предшествует процесс обучения на контрольной выборке, мы считаем, что по скорости обнаружения атак эти методы имеют высокие показатели. Кроме того, эти методы имеют более высокую точность обнаружения известных атак и хорошую точность обнаружения неизвестных атак. При этом доля ложных срабатываний в методах машинного обучения является средней.

В тоже время как показали эксперименты, сетевой трафик СПД обладает фрактальными свойствами. Иными словами, на больших объемах этот трафик обладает свойством самоподобия.

Кроме того, эксперименты продемонстрировали, что предлагаемая проактивная система защиты СПД при обнаружении КА на основе оценки самоподобия параметров функционирования системы с использованием фрактальных показателей и прогнозирования факта воздействия кибератак путем применения предложенной структуры нейронной сети с ячейками *LSTM* обладает достаточно высокой эффективностью при обнаружении как известных, так и

неизвестных КА. Вероятность обнаружения известных КА равна 0,96, а атаки «нулевого дня» - 0,8.

Главным достоинством предлагаемого подхода является высокая скорость детектирования аномалий, вызываемых КА, возможность работы с любыми видами трафика, а также низкая вероятность ложного срабатывания. Определение коэффициента Херста производится за несколько микросекунд в зависимости от производительности вычислительной техники, а обнаружение аномалий с помощью гибридной нейронной сети позволяет максимальность снизить вероятность ложного срабатывания.

В то же время, эксперименты показали, что наиболее популярные алгоритмы машинного обучения, протестированные на сгенерированных временных рядах, отлично справляются с обнаружением аномальных выбросов. В таком случае, аномалия проявляется в виде нестационарности некоторых наблюдаемых временных рядов. Это не только мгновенные скачки амплитуды измерений, но и медленные тренды, практически невидимые за время наблюдений.

4.5 Выводы по четвертому разделу

1. Разработана методика классификации КА в сетевом трафике СПД, которая позволяет выявлять КА с использованием гибридной нейронной сети, состоящей из классификатора и автокодировщика, обученного на основе данных работы функционирования СПД, учитывающего все отклонения от ее штатной работы.

Специфика предложенной методики является то, что обнаружение КА производится с использованием автокодировщика, обученного на основе эталонных данных работы СПД, информационного обмена в ней и классификатора, учитывающего все отклонения от штатной работы сети. В процессе работы автокодировщик дополнительно обучает классификатор, т.е. в итоге получается генеративно-состязательная сеть, в которой нейронные сети учатся друг у друга.

2. Предложена архитектура и программные прототипы компонентов

системы раннего обнаружения и классификации КА в сетевом трафике СПД, которые ориентированы на мгновенное обнаружение как известных, так и неизвестных КА, с минимальным количеством ложных срабатываний, их классификацию и выбор доступных контрмер. Архитектура содержит оригинальные компоненты выявления аномалий в сетевом трафике и обучения разработанной нейронной сети, базу данных с информацией о КА с целью их классификации. В силу этого архитектура позволяет формировать наборы исходных данных для исследований и разработок в области раннего обнаружения и противодействия КА в СПД, а также для исследований и разработок решений для систем поддержки принятия решения.

3. Проведена экспериментальная оценка предложенного подхода, которая показала, что, по сравнению со многими другими подходами, одним из главных преимуществ фрактального анализа является скорость его работы, а также возможность обнаружения аномалий в нестационарном трафике. К увеличению времени расчета приводит только увеличение количества обрабатываемых параметров заголовка протокола передачи данных (длина пакета, флаги и т.д.). При этом предложенная система продемонстрировала достаточно высокую вероятность обнаружения КА, достигнув значения 0.97 для известных атак и 0.99 для ранее неизвестных атак.

Предложенные программные компоненты могут являться элементом используемых в настоящее время *IDS* и *IPS*, основной задачей которых является анализ передаваемых внутренние потоки данных, находя в них последовательности битов, которые могут представлять из себя вредоносные действия или события, а также осуществляют мониторинг системных журналов и других файлов регистрации деятельности пользователей. Она позволит повысить вероятность обнаружения неизвестных КА за счет гибридной нейронной сети, уменьшить вероятность ложного срабатывания, время и объем оперативной памяти, задействованных для анализа сетевого трафика, что нивелирует недостатки существующих *IDS* и *IPS*, основанных на жестких правилах, а также сигнатурной и аномальной технологиях.

Кроме того, следует отметить, что проведенные исследования демонстрируют пока еще только потенциальную возможность и эффективность предложенной системы к прогнозированию и обнаружению КА в сети СПД. Практическая реализация и дальнейшее совершенствование этой системы, её распространение на различные виды КА, а также ее взаимодействие с другими методами являются дальнейшими направлениями исследований.

ЗАКЛЮЧЕНИЕ

Итоги выполненного исследования. В диссертационной работе решена научная задача, заключающаяся в разработке модели и методик выявления аномалий и классификации компьютерных атак в сетевом трафике сети передачи данных на основе применения фрактального анализа и методов машинного обучения.

Решенная задача имеет важное значение для совершенствования моделей, методик и средств раннего выявления аномалий в сетевом трафике сетей передачи данных, находящихся под воздействием как известных, так и неизвестных компьютерных атак, а также прогнозирования факта их воздействия.

Основные научные результаты, составляющие **итоги** выполненного исследования:

1) Проанализированы существующие модели воздействия компьютерных атак на сети передачи данных.

2) Проанализированы существующие алгоритмы выявления компьютерных атак, существующих систем мониторинга и методик противодействия компьютерным атакам в сетях передачи данных.

3) Разработана аналитическая модель выявления аномалий в сетевом трафике сетей передачи данных в условиях компьютерных атак.

4) Разработана методика раннего обнаружения аномалий в сетевом трафике сетей передачи данных.

5) Разработана методика классификации компьютерных атак в сетевом трафике сетей передачи данных.

6) Разработаны архитектуры и программные прототипы компонентов системы раннего обнаружения и классификации компьютерных атак в сетевом трафике сетей передачи данных.

7) Проведена экспериментальная и теоретическая оценка эффективности предложенных модели, методик и архитектуры, а также сравнение с существующими методиками.

Научная новизна результатов исследования заключается в том, что:

разработанная аналитическая модель выявления аномалий в сетевом трафике сетей передачи данных в условиях компьютерных атак, в отличие от известных, описывает не только стационарный, но и нестационарный сетевой трафик и обосновывает выбор фрактального метода выявления аномалий в интересах дальнейшей классификации компьютерных атак в зависимости от типа трафика;

разработанная методика раннего обнаружения аномалий в сетевом трафике сетей передачи данных, в отличие от известных, способна проводить прогнозирование и обнаружение не только известных, но и неизвестных компьютерных атак на раннем этапе их проявления благодаря совместному применению методов фрактального анализа и искусственной нейронной сети *LSTM*-типа (рекуррентные нейронные сети с долгой краткосрочной памятью), в результате чего существенно сокращается время выявления аномалий и уменьшается количество ложных срабатываний;

разработанная методика классификации компьютерных атак в сетевом трафике сетей передачи данных отличается от известных тем, что в ней обнаружение компьютерных атак производится с использованием генеративно-состязательной сети, которая производит дообучение классификатора на скрытых латентных представлениях, полученных автокодировщиком, исходя из анализа данных о функционировании сетей передачи данных;

предложенная архитектура и программные прототипы компонентов системы раннего обнаружения и классификации компьютерных атак в сетях передачи данных отличаются от известных тем, что они ориентированы на раннее обнаружение как известных, так и неизвестных компьютерных атак с минимальным количеством ложных срабатываний за счет реализации методов фрактального анализа и искусственной нейронной сети *LSTM*-типа.

Все выносимые на защиту результаты являются новыми и получены соискателем самостоятельно. Совокупное применение разработанных модели и методик, по сравнению со многими другими подходами, позволит увеличить скорость обнаружения компьютерных атак в 14 раз путем выявления аномалий в

трафике любого вида. Также предложенная система продемонстрировала достаточно высокую вероятность обнаружения компьютерных атак, достигнув значения 0.96 для известных атак и 0.8 для ранее неизвестных атак. Даны рекомендации по использованию результатов исследования для повышения защищённости сетей передачи данных от компьютерных атак.

Разработанные методики и модель направлены на сокращение времени выявления аномалий и классификации компьютерных атак в сетях передачи данных, и могут быть применены на предприятиях промышленности при выполнении научных исследований.

Рекомендации. Разработанные алгоритмы могут быть использованы в существующих системах глубокого анализа сетевого трафика (*IDS* и *IPS*), системах обнаружения атак, поскольку они представляют собой инструмент моментального выявления аномалий в сетевом трафике.

Перспективы дальнейшей разработки темы. В качестве перспектив дальнейшей разработки темы можно указать исследования, связанные с интеграцией предлагаемой системы с другими известными системами защиты и имеющимися в арсенале систем компьютерной безопасности методами детектирования атак, а также апробацию разработанного программного компонента обнаружения аномалий в сетевом трафике на основе принципов фрактального анализа данных на принципиально других типах сетей передачи данных.

Соответствие специальности. Полученные результаты соответствуют паспорту специальности 2.3.6 – «Методы и системы защиты информации, информационная безопасность».

СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

ОС – операционная система;

ПО – программное обеспечение;

КА – компьютерная атака;

СПД – сети передачи данных;

ПЭВМ – персональная электронно-вычислительная машина;

ЛВС – локальные вычислительные сети;

УС – узел связи;

АРМ – автоматизированное рабочее место;

АТС – автоматическая телефонная станция;

ИС – информационная система;

ЭМВОС – эталонная модель взаимодействия открытых систем;

ССОП – сети связи общего пользования;

МСЭ – межсетевой экран;

СУБД – система управления базами данных.

СПИСОК ЛИТЕРАТУРЫ

1. Шелухин О. И. Обнаружение вторжений в компьютерные сети (сетевые аномалии): учебное пособие / О. И. Шелухин, Д. Ж. Сакалема, А. С. Филинова; под редакцией О. И. Шелухина. — Москва: Горячая линия Телеком, 2018. - 220 с.

2. Шелухин О. И., Тенякшев А. М., Осин А. В. Фрактальные процессы в телекоммуникациях. – Закрытое акционерное общество Издательство Радиотехника, 2003.

3. . Шелухин О. И., Осин А. В. Мультифрактальные свойства трафика реального времени // Электротехнические и информационные комплексы и системы. – 2006. – Т. 2. – №. 3, С. 36-44.

4. Doukhan P., Oppenheim G., Taqqu M. (ed.). Theory and applications of long-range dependence. – Springer Science & Business Media, 2002.

5. Dang T., Sonkoly B., Molnar S. Fractal analysis and modeling of VoIP traffic. In Proceedings of the 11th International Telecommunications Network Strategy and Planning Symposium (NETWORKS 2004), Vienna, Austria, 13–16 June 2004; IEEE: Vienna, Austria; pp. 123–130.

6. Gers F., Schraudolph N., Schmidhuber J. Learning precise timing with LSTM recurrent networks // Journal of Machine Learning Research. 2002, Vol. 3, pp. 115-143.

7. Зегжда Д.П., Зегжда П.Д., Калинин М.О. Универсальный метод обнаружения кибератак на глобальные информационные системы поддержки цифровой экономики // Методы и технические средства обеспечения безопасности информации -2019. - № 28. - С. 48-49.

8. Молдовян А.А., Молдовян Н.А. Способы и алгоритмы псевдовероятностного шифрования с разделяемым ключом // Труды СПИИРАН. – 2018. – № 6 (61). – С. 5.

9. Зайцева Е.А., Зегжда Д.П., Полтавцева М.А. Использование графового представления и прецедентного анализа для оценки защищенности компьютерных систем // Проблемы информационной безопасности. Компьютерные системы -2019. - № 2. - С. 136-148.

10. . Васильев Ю.С., Зегжда П.Д., Зегжда Д.П. Обеспечение безопасности автоматизированных систем управления технологическими процессами на объектах гидроэнергетики // Известия Российской академии наук. Энергетика. – 2016. – № 3. – С. 49-61.

11. Радько Н.М., Скобелев И.О. Риск-модели информационно-телекоммуникационных систем при реализации угроз удаленного и непосредственного доступа – Москва.: РадиоСофт, 2011. – 229 с.

12. Антипов С.Г., Фомина М.В. Проблема обнаружения аномалий в наборах временных рядов // Программные продукты и системы – 2012. – № 2. – С. 78- 82.

13. Котенко Д.И., Котенко И.В., Саенко И.Б. Методы и средства моделирования атак в больших компьютерных сетях: состояние проблемы // Труды СПИИРАН. 2012. № 3 (22). С. 5–30.

14. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения.

15. ГОСТ Р 51275-2006. Защита информации. Объект информатизации. Факторы, воздействующие на информацию. Общие положения.

16. ГОСТ Р 53109-2008. Система обеспечения информационной безопасности сети связи общего пользования. Паспорт организации связи по информационной безопасности.

17. ГОСТ Р 53110-2008. Система обеспечения информационной безопасности сети связи общего пользования. Общие положения.

18. ГОСТ Р 53111-2007. Устойчивость функционирования сети связи общего пользования. Требования и методы проверки.

19. ГОСТ РВ 51987-2002. Информационная технология. Комплекс стандартов на автоматизированные системы типовые требования и показатели качества функционирования информационных систем. Общие положения.

20. ГОСТ Р ИСО 7498-2-99. Информационная технология. Взаимосвязь открытых систем базовая эталонная модель. Часть 2. Архитектура защиты информации.

21. ГОСТ Р ИСО/МЭК 27000-2012. Информационная технология. Методы и средства обеспечения безопасности. Системы менеджмента информационной безопасности. Общий обзор и терминология. М.: Стандартинформ, 2014. 22 с.

22. ГОСТ Р 53114-2008. Защита информации. Обеспечение информационной безопасности в организации. Основные термины и определения. М.: Стандартинформ, 2018. 20 с.

23. Доктрина Информационной безопасности. Указ Президента РФ от 5 декабря 2016 г. N 646 «Об утверждении Доктрины информационной безопасности РФ». – URL: https://demo.garant.ru/#/document/71556224/paragraph/1/doclist/1042/showentries/0/highlight/доктрина информационной безопасности:0 (дата обращения: 26.01.2021).

24. Указ Президента РФ от 15.01.2013 № 31с «О создании государственной системы обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации».

25. Приказ Минкомсвязи Российской Федерации от 25.08.2009 № 109 «Об утверждении требований по обеспечению целостности, устойчивости функционирования и безопасности информационных систем общего пользования».

26. Приказ Мининформсвязи Российской Федерации от 27.09.2007 № 113 «Об утверждении требований к организационно-техническому обеспечению устойчивого функционирования сети связи общего пользования».

27. Федеральный закон «Об информации, информационных технологиях и о защите информации» от 27 июля 2006 г. № 149-ФЗ. – М.: Кнорус, 2014. – 20 с.

28. Федеральный закон «О безопасности» от 28 декабря 2010 г. № 390-ФЗ. – М.: Кнорус, 2011. – 16 с.

29. ГОСТ Р ИСО/МЭК 27005-2010. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности. М.: Стандартинформ, 2011. 51 с. 70. РС БР ИББС-2.2-2009.

Методика оценки рисков нарушения информационной безопасности. М., 2009. 23 с. 71.

30. ГОСТ Р ИСО/МЭК 27006-2008. Информационная технология. Методы и средства обеспечения безопасности. Требования к органам, осуществляющим 111 аудит и сертификацию систем менеджмента информационной безопасности. Введ. 2009-10-01. М.: Стандартинформ, 2010. 40 с.

31. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы: Учебник для вузов. 4-е изд. — СПб.: Питер, 2010. – 944 с.

32. Гриняев С.Н. Поле битвы – киберпространство: Теория, приемы, средства, методы и системы ведения информационной войны. – Мн.: Харвест, 2004. – 448 с.

33. Сухопаров М.Е., Семенов В.В., Лебедев И.С. Модель поведения для классификации состояния информационной безопасности автономного объекта // Проблемы информационной безопасности. Компьютерные системы -2019. - № 4. - С. 26-34.

34. Королев В.И. Методология построения модели угроз безопасности территориально-распределенных объектов / В. И. Королев // Технология техносферной безопасности: интернет-журнал. –2013. – № 2 (48).

35. Саенко И. Б. Модель угроз ресурсам ИТКС как ключевому активу критически важного объекта инфраструктуры / И. Б. Саенко, О. С. Лаута, М. А. Карпов, А. М. Крибель // Электросвязь. – 2021. – № 1. – С. 36–44.

36. Ageev S., Kotenko I., Saenko I., Kopchak Y. Abnormal Traffic Detection in Networks of the Internet of Things Based on Fuzzy Logical Inference // Proceedings of the IEEE International Conference on Soft Computing and Measurements (SCM). 2015. pp.5–8.

37. Al-Jarrah M., Khalaf G., Amin S. PIN Authentication Using Multi-Model Anomaly Detection in Keystroke Dynamics // Proceedings of the 2019 2nd International Conference on Signal Processing and Information Security (ICSPIS), Dubai, United Arab Emirates, 2019, pp. 1-4.

38. Brezigar-Masten A., Masten I. CART-based selection of bankruptcy predictors for the logit model // Expert Systems with Applications. 2012. vol. 39. no. 11. pp. 10153–10159.

39. Ju X., Chen V.C.P.; Rosenberger J.M., Liu F. Fast knot optimization for multivariate adaptive regression splines using hill climbing methods // Expert Systems with Applications. 2021. no. 171. p. 114565.

40. Ju X., Rosenberger J.M., Chen V.C.P., Liu F. Global optimization on non-convex two-way interaction truncated linear multivariate adaptive regression splines using mixed integer quadratic programming // Information Sciences. 2022. no. 597. pp. 38–52.

41. Ju X., Liu F., Wang Li., Lee W.-J. Wind farm layout optimization based on support vector regression guided genetic algorithm with consideration of participation among landowners // Energy Conversion and Management. 2019. no. 196. pp. 1267–1281.

42. Leland W.E., Taqqu M.S., Willinger W., Wilson D.V. On the self-similar nature of Ethernet traffic // SIGCOMM Comput. Commun. 1993. vol. 23. no. 4. pp. 183–193.

43. Singh Gulshan M.B., Sharma B., Grover M., Gupta P. TSA: Self-Train Self-Test Algorithm // Proceedings of the 2020 IEEE International Conference for Innovation in Technology (INOCON). 2020. pp. 1–5.

44. Yu Z., Jiang Z., Tan L., Liu H., Yang Q. Rescaled Range Analysis of Vessel Traffic Flow in the Yangtze River // Proceedings of the 2019 5th International Conference on Transportation Information and Safety (ICTIS). 2019. pp. 1–4.

45. Shaukat S., Ali A., Batool A., Alqahtan, F., Khan J.S., Ahmad A. J. Intrusion Detection and Attack Classification Leveraging Machine Learning Technique // Proceedings of the 2020 14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab, 17-18 November 2020; IEEE: New York, USA, 2020, pp. 198-202.

46. Nurul A.H., Zaheera Z.A., Puvanasvaran A.P., Zakaria N.A., Ahmad R. Risk assessment method for insider threats in cyber security: A review // International

Journal of Advanced Computer Science and Applications (ijacsa) 2018, 9(11), pp.16–19.

47. Zhe W.; Wei C., Chunlin L. DoS attack detection model of smart grid based on machine learning method // Proceedings of the 2020 IEEE International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 28-30 July 2020; IEEE: New York, USA, 2020, pp. 735-738.

48. Karataş G., Akbulut A. Survey on Access Control Mechanisms in Cloud Computing // Journal of Cyber Security and Mobility. 2018, 7(3), pp. 1–36.

49. Lopez J., Rubio J. Access control for cyber-physical systems interconnected to the cloud // Comput. Netw. 2018, 134(C), pp. 46–54.

50. Clincy V., Shahriar H. Web Application Firewall: Network Security Models and Configuration // Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). 2018, pp. 835-836.

51. Visoottiviseth V., Sakarin P., Thongwilai J. Choobanjong, T. Signature-based and behavior-based attack detection with machine learning for home IoT devices // Proceedings of the 2020 IEEE REGION 10 CONFERENCE (TEN-CON), Osaka, Japan, 16-19 November 2020; IEEE: New York, USA, 2020, pp. 829-834.

52. Amma N.G.B., Selvakumar S., Velusamy R.L. A Statistical Approach for Detection of Denial of Service Attacks in Computer Networks // IEEE Transactions on Network and Service Management. 2020, 17(4), 2511–2522.

53. Климов С.М. Методы и модели противодействия компьютерным атакам. – Люберцы.: Каталист, 2008. – 316 с.

54. Raimundo M., Okamoto Jr. Application of Hurst Exponent (H) and the R/S Analysis in the Classification of FOREX Securities. International Journal of Modeling and Optimization 2018, 8, 116–124.

55. Sánchez-Granero M., Fernández-Martínez M., Trinidad-Segovia J., Introducing fractal dimension algorithms to calculate the Hurst exponent of financial time series. Eur. Phys. J. B. 2012, 85, article 86.

56. Grillo D., Lewis A., Pandya R. Personal Communication Services and Teletraffic Standardization in ITU-T. In The Fundamental Role of Teletraffic in the

Evolution of Telecommunications Networks, Proceedings of the 14th International Teletraffic Congress - ITC 14, Antibes Juan-les-Pins, France, 6-10 June, 1994, J. Labetoulle and J.W. Roberts, Eds.; Elsevier Science, 1994; pp. 1–12.

57. Strelkovskaya I., Solovskaya I., Makoganiuk A. Spline-Extrapolation Method in Traffic Forecasting in 5G Networks. Journal of Telecommunications and Information Technology 2019, 3, 8–16.

58. Carvalho P., Abdalla H., Soares A., Solis P., Tarchetti P. Analysis of the influence of self-similar traffic in the performance of real time applications. Available online. – URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.599.4041 &rep=rep1&type=pdf (дата обращения: 15.07.2020).

59. Ruoyu Ya., Wang Yi., Hurst Parameter for Security Evaluation of LAN Traffic. Information Technology Journal 2012, 11, 269–275.

60. Ably P., Flandrin P., Taqqu M., Veitch D. Self-Similarity and long-range dependence through the wavelet lens. In Theory and Applications of Long Range Dependence; Boston: Birkhauser Press, 2002; pp. 345–379.

61. Минькович Т.В. Информационные технологии : понятийно - терминологический аспект / Т.В. Минькович // ОТО. 2012. – Т. 2. – С. 371–389.

62. Расторгуев С.П. Литвиненко М.В. Информационные войны в сети Интернет / под ред. Михайловского А.Б. – М.: АНО «Центр стратегических оценок и прогнозов», 2014. – 128 с.

63. Макаренко С.И., Чукляев И.И. Терминологический базис в области информационного противоборства / С.И Макаренко, И.И. Чукляев // Вопросы кибербезопасности. 2014. – № 1 (2). – С. 13–21.

64. Михайлов А. П., Маревцева Н. А. Модели информационной борьбы //Математическое моделирование. – 2011. – Т. 23. – №. 10. – С. 19-32.

65. ISO, «ISO/IEC TR 27019:2013: Information security management guidelines based on ISO/IEC 27002 for process control systems specific to the energy utility industry», 2013.

66. Kendrick D., Groom L., Stewart J. Cluster randomised controlled trial evaluating an injury prevention program. Injury Prevention 13(2):93-8, 2016.

67. Fang X., Misra S., Xue G., Yang D. Managing Smart Grid Information in the Cloud: Opportunities, Model, and Applications. Article in IEEE Network July 2017. DOI: 10.1109/MNET.2012.6246750.

68. Indrajeet Prasad. Smart Grid Technology: Application and Control. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 3, Issue 5, May 2014.

69. Müller K. Verordnete Sicherheit - das Schutzprofil für das Smart Metering Gateway - Eine Bewertung des neuen Schutzprofils. Datenschutz und Datensicherheit, Vol. 35, No. 8, pp. 547–551, 2014.

70. Federal Office for Information Security, Protection Profile for the Security Module of a Smart Metering System, V.1.0, March, 2015.

71. Anwar A., University D., Mahmood A. Cyber Security of Smart Grid Infrastructure. In book: The State of the Art in Intrusion Prevention and Detection Publisher: CRC Press, January 2014, DOI: 10.1201/b16390-9.

72. Bale J., Sediyono E. Facilitated Risk Analysis Process. Risk management in information technology using facilitated risk analysis process (FRAP). Academic information systems of Satya Wacana Christian University, 2015

73. Tankard C. Advanced persistent threats and how to monitor and deter them. Network Security, Vol. 2011, No. 8, pp. 16–19, 2011.

74. Tushar P. Parikh, dr. Ashok R. Patel. Cyber security: Study on Attack, Threat, Vulnerability. International Journal of Research in Modern Engineering and Emerging Technology Vol. 5, Issue: 6, June: 2017.

75. James P. Sterbenz et al. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. Computer Networks, Vol. 54, No. 8, June 2010, pp. 1245-1265.

76. Imed el Fray. A comparative study of risk assessment methods, MEHARI & CRAMM with a new formal model of risk assessment (FoMRA) in information systems. Conference: Proceedings of the 11th IFIP TC 8 international conference on Computer Information Systems and Industrial Management, Sep. 2012.

77. Amril Syalim. Comparison of Risk Analysis Methods: MEHARI, MAGERIT, NIST800-30 and Microsoft’s Security Management Guide. сайт – URL: http://itslab.inf.kyushu-u.ac.jp

78. Mehari – Overview. Club de la Securit ´ e de l’Information Franc¸ais ´(CLUSIF). – 2010.

79. Microsoft security center of excellence. сайт – URL: http://www.microsoft.com/rus/technet/security.

80. Cyber Security Trends You Can’t Ignore in 2021, сайт – URL: https://purplesec.us/cyber-security-trends-2021/, (дата обращения: 24.04.2021).

81. Cybersecurity Statistics and Trends for 2021, сайт – URL: https://www.varonis.com/blog/cybersecurity-statistics/, last accessed 2021/04/24.

82. Clincy V., Shahriar H. Web Application Firewall: Network Security Models and Configuration. In: Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), pp. 835-836, 2018.

83. Baddar Sh.Al-H., Merlo A., Migliardi M. Anomaly Detection in Computer Networks: A State-of-the-Art Review. J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl. 5 (2014) 29-64.

84. Bodström T., Hämäläinen T. State of the Art Literature Review on Network Anomaly Detection with Deep Learning. In: Internet of Things, Smart Spaces, and Next Generation Networks and Systems. Lecture Notes in Computer Science, vol. 11118, Springer, Cham, pp. 64-76, 2018.

85. Pathan Al-S.Kh. The State of the Art in Intrusion Prevention and Detection. Auerbach Publications, 2016.

86. Ahmed M., Mahmood A., Hu J. A survey of network anomaly detection techniques. J. Netw. Comput. Appl. 60(C) (2016), 19-31.

87. Himeur Y., Ghanem Kh., Alsalemi A., Bensaali F., Amira A. Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives. Applied Energy 287 (2021) 116601.

88. Branitskiy A., Kotenko I. Analysis and Classification of Methods for Network Attack Detection. SPIIRAS Proceedings 2(45). (2016), С. 207-244.

89. Chai J., Zhu H. Detecting anomalies in data center physical infrastructures using statistical approaches. Journal of Physics: Conference Series 1176 (2) (2019) 022056.

90. Nawir M., Amir A., Yaakob N., Lynn O. Effective and efficient network anomaly detection system using machine learning algorithms. Bulletin of Electrical Engineering and Informatics 8(1) (2019) 46-51.

91. Salman T., Bhamare D., Erbad A. Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments. In: Proceedings of the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, pp. 97-103, 2017.

92. Chan P., Lippmann R. Machine Learning for Computer Security. Journal of Machine Learning Research **7** (2006) 2669-2672.

93. Arslan B., Gunduz S., Sagiroglu S. A review on mobile threats and machine learning based detection approaches. In:. Proceedings of the 2016 4th Int. Symp. on Digital Forensic and Security (ISDFS), pp.7-13, 2016.

94. Shamili A., Bauckhage C., Alpcan T. Malware detection on mobile devices using distributed machine learning. In: Proceedings of the 2010 20th Int. Conf. on Pattern Recognition (ICPR), IEEE, pp. 4348-4351, 2010.

95. Sahs J., Khan L. A machine learning approach to Android malware detection. In: Proceedings of the 2012 European Intelligence and Security Informatics Conference (EISIC), pp. 141-147, 2012.

96. Joseph A., Laskov P., Roli F., Tygar J. Machine Learning Methods for Computer Security (Dagstuhl Perspectives Workshop 12371). Dagstuhl Manifestos 3(1) (2013) 1-30.

97. Ford V., Siraj A. Applications of Machine Learning in Cyber Security. In: Proceedings of the 27th International Conference on Computer Applications in Industry and Engineering (CAINE-2014), New Orleans, Louisiana, USA, pp. 27-32, 2014.

98. Xiao L., Wan X., Lu X., Zhang Y, Wu D. IoT Security Techniques Based on Machine Learning: How Do IoT Devices Use AI to Enhance Security? IEEE Signal Processing Magazine 35(5) (2018) 41-49.

99. Kotenko I., Saenko I., Branitskiy A. Detection of Distributed Cyber Attacks Based on Weighed Ensemble of Classifiers and Big Data Processing Architecture. In: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Paris, France, pp. 1-6, 2019.

100. Radhakrishnan C., Karthick K., Asokan R. Ensemble Learning based Network Anomaly Detection using Clustered Generalization of the Features. In: Proceedings of the 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, pp. 157-162, 2020.

101. Avramenko V, Kotenko I., Malikov A., Saenko I., Combined Neural Network Model for Diagnosing Computer Incidents. In: Russian Advances in Artificial Intelligence: selected contributions to the Russian Conference on Artificial intelligence (RCAI 2020). CEUR Workshop Proceedings (CEUR-WS.org), vol. 2648, pp. 280-294, 2020.

102. Saenko I., Skorik F., Kotenko I. Combined neural network for assessing the state of computer network elements // In: Advances in Neural Computation, Machine Learning, and Cognitive Research IV. NEUROINFORMATICS 2020. Studies in Computational Intelligence, vol. 925. Springer, Cham, pp. 256-261, 2021.

103. Hoglund A., Hatonen K., Sorvari S. A Computer Host-Based User Anomaly Detection System Using the Self-Organizing Map. In: Proceedings of the IEEE-INNSENNS International Joint Conference on Neural Networks, vol. 5, pp. 411-416, 2000.

104. Wang W., Guan X., Zhang X., Yang L, Profiling program behavior for anomaly intrusion detection based on the transition and frequency property of computer audit data. Computers & Security 25(7) (2006) 539-550.

105. Saenko I., Skorik F., Kotenko I. Application of Hybrid Neural Networks for Monitoring and Forecasting Computer Networks States. In: Advances in Neural Networks. Lecture Notes in Computer Science, vol. 9719, pp. 521-530, 2016.

106. Gnosh A., Michael C., Schatz M. A Real-Time Intrusion Detection System Based on Learning Program Behavior. In: Proceedings of the 3rd International

Workshop on Recent Advances in Intrusion Detection (RAID ’00), vol. 1907, pp. 93–109, 2000.

107. Sherry L. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In: Proceedings of the 2016 Integrated Communications Navigation and Surveillance (ICNS), pp. 5C2-1-5C2-8, 2016.

108. Ergen T, Kozat S. Unsupervised Anomaly Detection with LSTM Neural Networks. IEEE Transactions on Neural Networks and Learning Systems 31(8) (2020) 3127-3141.

109. Provotar O., Linder Y., Veres M. Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders. In: Proceedings of the 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), pp. 513-517, 2019.

110. Cherdo Y., Kerret P., Pawlak R. Training LSTM for Unsupervised Anomaly Detection Without A Priori Knowledge. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4297-4301, 2020.

111. Elsayed M., Le-Khac N., Dev S., Jurcut A. Network Anomaly Detection Using LSTM Based Autoencoder. In: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, pp. 37–45, 2020.

112. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов / В. Г. Олифер, Н. А. Олифер. – 5-е изд. – СПб. : 2016. – 821 с.

113. Ивницкий В. А. Теория сетей массового обслуживания. В. А. Иваницкий. – М. : Физматлит, 2004. – 772 с.

114. Дуплякин В. М. Выбор закона распределения входного потока заявок при моделировании системы массового обслуживания торгового предприятия / В. М. Дуплякин, Ю. Н. Княжева // Вестник Самарского государственного аэрокосмического университета. – 2012. – Т. 1. – Вып. 37. – С. 102–111.

115. Цициашвили Г. Ш. Стационарные потоки в ациклических сетях массового обслуживания / Г. Ш. Цициашвили, М. А. Осипова // ДВМЖ. – 2016. Т. 16. – Вып. 2. – С. 223–228.

116. Kotenko I., Saenko I., Lauta O., Kribel A. An approach to detecting cyber attacks against smart power grids based on the analysis of network traffic self-similarity. Energies. 2020. Т. 13. № 19. С. 5031.

117. Kotenko I., Saenko I., Lauta O., Karpov M. Methodology for management of the protection system of smart power supply networks in the context of cyberattacks. Energies. 2021. Т. 14. № 18.

118. Котенко И. В. Анализ процесса самоподобия сетевого трафика как подход к обнаружению кибератак на компьютерные сети / И. В. Котенко, А. М. Крибель, О. С. Лаута, И. Б. Саенко // Электросвязь. – 2020. – № 12. – С. 54–59.

119. Крибель А. М. Методика обнаружения коллизий сетевого трафика / А. М. Крибель // Известия Тульского государственного университета. Технические науки. – 2021. – № 12. – С. 182–190.

120. Kribel A., Saenko I., Kotenko I. Detection of Anomalies in the Traffic of Information and Telecommunication Networks Based on the Assessment of its Self-Similarity // Proceedings - 2020 International Russian Automation Conference, RusAutoCon 2020, 2020, стр. 713–718, 9208147.

121. Kotenko I., Saenko I., Lauta O., Kribel A. Ensuring the survivability of embedded computer networks based on early detection of cyber attacks by integrating fractal analysis and statistical methods. Microprocessors and Microsystemsthis link is disabled, 2022, 90, 104459.

122. Kotenko I., Saenko I., Kribel A., Lauta O. A technique for early detection of cyberattacks using the traffic self-similarity property and a statistical approach. Proceedings - 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2021. 29. 2021. С. 281-284.

123. Kotenko, I., Lauta, O., Kribel, K., Saenko, I. LSTM neural networks for detecting anomalies caused by web application cyber attacks. Frontiers in Artificial Intelligence and Applicationsthis link is disabled, 2021, 337, стр. 127–140.

124. Kotenko I., Saenko I., Lauta O., Kribel A. An approach to detecting cyber attacks against smart power grids based on the analysis of network traffic self-similarity // Energies, 2020, 13(19), 5031.

125. Крибель А. М. Метод обнаружения аномалий в сетевом компьютерном трафике на основе нейронной сети с использованием LSTM / А. М. Крибель, О. С. Лаута, А. В. Филин, А. С. Фень // Электросвязь. – 2021. – № 12. – С. 43–48.

126. Котенко И. В. Метод раннего обнаружения кибератак на основе интеграции фрактального анализа и статистических методов / И. В. Котенко, И. Б. Саенко, О. С. Лаута, А. М. Крибель // Первая миля. – 2021. – № 6 (98). – С. 64–71.

127. Kotenko I., Saenko I., Kribel A., Lauta O. A technique for early detection of cyberattacks using the traffic self-similarity property and a statistical approach // Proceedings - 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2021, 2021, С. 281–284, 9407132.

128. Kotenko, I., Saenko, I., Lauta, O., Kribel, A. Ensuring the survivability of embedded computer networks based on early detection of cyber attacks by integrating fractal analysis and statistical methods // Microprocessors and Microsystemsthis link is disabled, 2022, 90, 104459.

129. Карпов М. А. Подход к управлению системой защиты информационно-телекоммуникационной сети специального назначения / М. А. Карпов, О. С. Лаута, М. А. Коцыняк, А. М. Крибель // Известия Тульского государственного университета. Технические науки. – 2020. – № 7. – С. 216–226.

130. Унтеров Д. С. Разработка метода обнаружения аномалий сетевого трафика на границе ЛВС предприятия : магистерская дисс. : 09.04.01 / Д. С. Унтеров ; Санкт-Петербургский политехнический университет Петра Великого, Институт компьютерных наук и технологий ; науч. рук. С. И. Городецкая. – СПб., 2016. – 80 с.

131. Messier G., Finvers I. Traffic Models for Medical Wireless Sensor Networks // IEEE Communications Letters. January 2007. Vol. 11, no. 1. P. 13–15. DOI: 10.1109/LCOMM.2007.061291.

132. Wang Q., Zhang T. Source traffic modeling in wireless sensor networks for target tracking // In Proc. of the 5th ACM International Symposium on Performance Evaluation of Wireless Ad-Hoc, Sensor, and Ubiquitous Networks (PE-WASUN'08). 2008. P. 96–100.

133. Vybornova A., Koucheryavy A. Traffic Analysis in Target Tracking Ubiquitous Sensor Networks // 14th International Conference, NEW2AN 2014 and 7th Conference, ruSMART 2014, Springer International Publishing, August 27–29, 2014.Vol. 8638. P. 389–398.

134. Kutuzov D., Osovsky A., Starov D., Stukach O. Processing of the Gaussian Traffic from IoT Sources by Decentralized Routing Devices // 2019 International Siberian Conference on Control and Communications (SIBCON). Proceedings. IEEE. 18–20 April 2019. Tomsk, Russia. DOI: 10.1109/SIBCON.2019.8729617.

135. Сетевая телеметрия Cisco против киберугроз // Блог компании CISCO. Информационная безопасность: сайт – URL: http://habrahabr.ru/company/cisco/blog/229073.

136. Androutsopoulos l., J. Koutsias, K.Y. Chandrinos, G. Paliouras, and C.D. Spyropoulos.2000a. An Evaluation of Naive Bayesian Anti-Spam Filtering. Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning, Barcelona, Spain, pages 917.

137. Лаута О. С. Киберустойчивость информационно-телекоммуникационной сети / О. С. Лаута, М. А. Коцыняк, А. М. Кудрявцев, И. А. Кулешов. – СПб. : Бостон-спектр, 2015. – 150 с.

138. Орлов Ю. Н. Методы статистического анализа литературных текстов / Ю. Н. Орлов, К. П. Осминин. – М. : Editorial URSS, Книжный дом «Либроком», 2012. – 312 с.

139. Котенко И. В. Программный компонент обнаружения аномалий сетевого трафика на основе принципов фрактального анализа данных / И. В. Котенко, И. Б. Саенко, О. С. Лаута, А. М. Крибель // Свидетельство о регистрации программы для ЭВМ 2021680188, Зарегистрировано в Реестре программ для ЭВМ 07.12.2021.

140. Баранов С. Н. Разработка и сертификация программного обеспечения для авиационных бортовых систем и оборудования : учеб. пособие. – СПб. : Изд-во ГУАП, 2017. – 175 с.

141. Стандарт ИСО 8402–94. Управление качеством и обеспечение качества – Словарь. – 29 с.

142. Липаев В. В. Тестирование компонентов и комплексов программ : учебник. – М. : Синтег, 2010. – 392 с.

143. Баранов С. Н., Тележкин А. М. Метрическое обеспечение программных разработок // Труды СПИИРАН. – 2014. – Т.5. – № 36. – С. 5-27.

144. Микони С. В., Соколов Б. В., Юсупов Р. М. Квалиметрия моделей и полимодельных комплексов. – 2018.

145. Соколов Б. В., Юсупов Р. М. Концептуальные основы оценивания и анализа качества моделей и полимодальных комплексов // Известия РАН. Теория и системы управления. – 2004. – № 6. – С. 5–16.

146. ГОСТ 28195-89. Оценка качества программных средств. Общие положения. М.: Издательство стандартов, 1989. 38 с.

Приложение А. Список опубликованных научных трудов соискателя ученой степени

Публикации в зарубежных изданиях из баз данных WOS и Scopus:

1. Kotenko, I., Saenko, I., Lauta, O., Kribel, A. A proactive protection of smart power grids against cyberattacks on service data transfer protocols by computational intelligence methods // Sensors 2022, 22, 7506.

2. Kotenko, I., Saenko, I., Lauta, O., Kribel, A. An approach to detecting cyber attacks against smart power grids based on the analysis of network traffic self-similarity // Energies, 2020, 13(19), 5031.

3. Kribel, A., Saenko, I., Kotenko, I. Detection of Anomalies in the Traffic of Information and Telecommunication Networks Based on the Assessment of its Self-Similarity // Proceedings - 2020 International Russian Automation Conference, RusAutoCon 2020, 2020, pp. 713–718, 9208147.

4. Kotenko, I., Saenko, I., Kribel, A., Lauta, O. A technique for early detection of cyberat-tacks using the traffic self-similarity property and a statistical approach // Proceedings - 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP 2021, 2021, pp. 281–284, 9407132.

5. Kotenko, I., Saenko, I., Lauta, O., Kribel, A. Ensuring the survivability of embedded computer networks based on early detection of cyber attacks by integrating fractal analysis and statistical methods // Microprocessors and Microsystemsthis link is disabled, 2022, 90, 104459.

6. Kotenko, I., Saenko, I., Lauta, O., Kribel, A. Anomaly and cyber attack detection technique based on the integration of fractal analysis and machine learning methods // Informatics and Automationthis link is disabled, 2022, 21(6), pp. 1328–1358.

Публикации в рецензируемых журналах из списка ВАК (по специальности 2.3.6):

7. Панков А.В., Крибель А.М., Лаута О.С., Васильев Н.А. Метод по совершенствованию информационно- аналитической работы на основе

комплексирования результатов распознавания состояний объектов контроля с использованием методов машинного обучения // Наукоемкие технологии в космических исследованиях Земли. 2022. Т. 14. № 2. С. 27-35.

8. Перов Р.А., Лаута О.С., Крибель А.М., Федулов Ю.М. Комплексная методика обнаружения кибератак на основе интеграции фрактального анализа и статистических методов // Наукоемкие технологии в космических исследованиях Земли. 2022. Т. 14. № 2. С. 44-51.

9. Перов Р.А., Лаута О.С., Крибель А.М., Федулов Ю.В. Метод выявления аномалий в сетевом трафике // Наукоемкие технологии в космических исследованиях Земли. 2022. Т. 14. № 3. С. 25-31.

В других изданиях (в рецензируемых журналах из списка ВАК):

10. Котенко И.В., Саенко И.Б., Лаута О.С., Крибель А.М. Метод раннего обнаружения кибератак на основе интеграции фрактального анализа и статистических методов // Первая миля. 2021. № 6 (98). С. 64-71.

11. Крибель А.М., Лаута О.С., Филин А.В., Фень А.С. Метод обнаружения аномалий в сетевом компьютерном трафике на основе нейронной сети с использованием LSTM // Электросвязь. 2021. № 12. С. 43-48.

12. Саенко И.Б., Лаута О.С., Карпов М.А., Крибель А.М. Модель угроз ресурсам ИТКС как ключевому активу критически важного объекта инфраструктуры // Электросвязь. 2021. № 1. С. 36-44;

13. Котенко И.В., Крибель А.М., Лаута О.С., Саенко И.Б. Анализ процесса самоподобия сетевого трафика как подход к обнаружению кибератак на компьютерные сети // Электросвязь. 2020. № 12. С. 54-59.

14. Крибель А.М. Методика обнаружения коллизий сетевого трафика // Известия Тульского государственного университета. Технические науки. 2021. № 12. С. 182-190.

15. Крибель А.М., Перов Р.А., Лаута О.С., Сычужников В.Б. Методика обнаружения компьютерных атак с помощью фрактального анализа и методов машинного обучения // Известия Тульского государственного университета. Технические науки. 2022. № 5. С. 166-178.

16. Крибель А.М., Перов Р.А., Лаута О.С., Скоробогатов С.Ю. Модель выявления аномалий в сетевом трафике сети передачи данных в условиях компьютерных атак // Известия Тульского государственного университета. Технические науки. 2022. № 5. С. 228-239.

Свидетельство о государственной регистрации программ для ЭВМ:

17. Котенко И. В., Саенко И. Б., Лаута О. С., Крибель А.М. Программный компонент обнаружения аномалий сетевого трафика на основе принципов фрактального анализа данных // Свидетельство о регистрации программы для ЭВМ 2021680188, 07.12.2021.

Приложение Б. Модель угроз ресурсам сети передачи данных

Таблица Б 1 – Модель угроз серверу

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
Порты сетевых служб сервера									
Сканирование сетевых сервисов Обнаружение периферийных устройств Обнаружение параметров конфигурации сети	Bash History	Эксфильтрация через альтернативный протокол Эксфильтрация через канал управления C2 Эксфильтрация через альтернативную сетевую среду Эксфильтрация через альтернативную физическую среду	Внешние удаленные сервисы	Regeorg	Фишинговый сервис Рассылка фишинг-email	Распространенные порты	Подключение через прокси Собственный криптографический протокол Многократное проксирование Многоуровневое шифрование Средства удаленного доступа	Проброс портов SSH тунелирование	Удаление подключений к сетевым ресурсам Скрытие конечного адреса соединения Запасные каналы Многоступенчатые каналы
Блок управления сервером баз данных									
Секретные ключи Обнаружение учетных записей Обнаружение общих сетевых ресурсов	Перехват пути Дампинг учетных данных	Эксплойты для получения учетных данных	Удаленный сеанс DCShadow Метод грубой силы или полный перебор	Sql injections	-	SQL	Протоколы управления БД	Выполнение с помощью локального планирования задач	Удаление подключений к базе данных

Оперативная память								
Обнаружение процессов	Дампинг учетных данных Захват ввода	-	EWM-инъекции Чтение файлов с помощью логических смещений файловой системы Process Doppelganging	-	-	cold boot attack уязвимость Rowhammer	-	-
Операционная система								
Bash History Секретные ключи Обнаружение учетных записей Обнаружение файлов и каталогов Раскрытие парольной политики Обнаружение групп доступа Обнаружение учетных записей Обнаружение информации о системе	Дампинг учетных данных Учетные данные в Реестре Автома тизированны й сбор Данные из хранилищ информации	Эксплойты для получения учетных данных Форсированная аутентификация Отравление LLMNR/NBT-NS Kerberoasting DLL-библиотеки фильтров паролей Автоматизированная эксфильтрация	Внешние удаленные сервисы. Выполнение с помощью эксплойтов. Windows Remote Management. Аппаратные закладки. Удаленный сеанс. Модификация файлов ~/.bash_profile и ~/.bashrc. Windows Remote Management. Перехват вызовов функций Windows API. Выполнение через загрузчик модулей Windows. Драйверы LSASS. Mshta. Regsvcs/Regasm.	Выполнение с помощью стороннего ПО для администрир ования сети Провайдеры времени	-	Windows Remote Management	Протокол удаленного рабочего стола Удаленные сервисы Связь через съемные носители	Выполнение с помощью локального планирования задач Создание учетных записей Logon-скрипты Новые службы Автозапуск с помощью ключа Run Keys и папки «Автозагрузка» Winlogon Helper DLL Прокси-выполнение кода через	Скрытые файлы и папки Port Knocking Отключение средств защиты Переменная HISTCONTROL Маскарадинг

Обнаружение сетевых подключений Обнаружение системных сервисов			Regsvr32. Rundll32. Windows Management Instrumentation. CMSTP. Модификация ключа AppCert DLLs. Злоупотребление подсистемой совместимости приложений. Модификация компонентов Windows Authentication Package. Перехват ссылок и связей Component Object Model Hijacking. Перехват поиска DLL. IFEO-инъекции. Расширения и загружаемые модули ядра. Модификация существующих служб. Вспомогательные DLL утилиты Netsh. Захват SIP и Trust Provider. Security Support Provider.					подписанные бинарники/сценарии	

			Слабости разрешений параметров служб в реестре. Windows Management Instrumentation Event Subscription. Манипуляции с маркерами доступа. Модификация исполняемых файлов приложений «специальные возможности Windows». Непрямое выполнение команд. Установка корневого сертификата. Модификация реестра. NTFS-атрибуты файла. Выдалбливание процесса/ инъекция кода в процесс. Hooking. Pass the Hash. Pass the Ticket. Протокол удаленного рабочего стола.						

Программное обеспечение									
Секретные ключи Обнаружение программных средств обеспечения безопасности	Дампинг учетных данных Перехват двухфакторной аутентификации	Эксплойты для получения данных	Эксплойты публичных приложений. Выполнение через подписанные бинарники, сценарии. InstallUtil. Выполнение с помощью стороннего ПО для администрирования сети. Выполнение через API. Создание заданий BITS. Эксплуатация уязвимостей средств защиты	Скриптинг Web Shell Упаковка софта Общедоступный Webroot	Выполнение через доверенные утилиты разработчиков софта	-	Удаленное копирование файлов	Руткиты	Деобфускация/дешифровка файлов или информации Обфускация файлов или информации Пробел после имени файла Timestomp Web-сервис
Аппаратная составляющая									
Секретные ключи	-	-	Аппаратные закладки, компрометация цепи поставок	Гипервизор	Доверительные отношения	Rootkit	Буткиты	-	Резервный доступ

Таблица Б 2 - Модель угроз криптомаршрутизатору.

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
Секретные ключи	-	-	Аппаратные закладки, компрометация цепи поставок	-	Доверительные отношения	-	Внутренний нарушитель	-	-

Таблица Б 3 – Модель угроз межсетевому экрану

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
Порты сетевых служб межсетевого экрана									
Сканирование сетевых	Прослушивание сети	Эксфильтрация через альтернативный протокол	Внешние	Порты сетевых служб маршрутизатор	-	Распространенные сервисы	Подключение через прокси	Проброс портов	Скрытие конечного адреса соединения

сервисов Обнаружение периферийных устройств Обнаружение параметров конфигурации сети		Эксфильтрация через канал управления C2 Эксфильтрация через альтернативную сетевую среду Эксфильтрация через альтернативную физическую среду	удаленные сервисы Метод грубой силы или полный перебор	а			Собственный криптографический протокол Многократное проксирование Многоуровневое шифрование Средства удаленного доступа		Запасные каналы Многоступенчатые каналы
Оперативная память									
Обнаружение процессов	Дампинг учетных данных Захват ввода	-	-	-	-	-	Cold boot attack Уязвимость Rowhammer	-	-
Операционная система									
Секретные ключи. Обнаружение учетных записей. Обнаружение файлов и каталогов. Раскрытие парольной политики. Обнаружение групп доступа. Обнаружение информации о системе. Обнаружение системных сервисов	Дампинг учетных данных Захват ввода Автоматизированный сбор	Эксплойты для получения учетных данных Автоматизированная эксфильтрация	Эксплойты для повышения привилегий	Выполнение с помощью стороннего ПО для администрирования сети	-	TTY	Удаленные сервисы	Выполнение с помощью локального планирования задач Создание учетных записей	Скрытые файлы и папки Port Knocking
Программное обеспечение									
Секретные ключи Обнаружение учетных записей	Дампинг учетных данных	Эксфильтрация данных из ПО	Эксплойты для ПО	Скриптинг	-	-	Удаленное копирование файлов	Inject в ПО	Деобфускация/дешифровка файлов или информации
Аппаратная составляющая									
Секретные ключи	-	-	Аппаратные закладки, компрометация	Гипервизор	Доверительные отношения	Rootkit	Буткиты	-	Резервный доступ

			цепи поставок						

Таблица Б 4 – Модель угроз ПЭВМ

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
Порты сетевых служб ПЭВМ									
Сканирование сетевых сервисов Обнаружение периферийных устройств Обнаружение параметров конфигурации сети	Прослушивание сети	Эксфильтрация через альтернативный протокол Эксфильтрация через канал управления C2 Эксфильтрация через альтернативную сетевую среду Эксфильтрация через альтернативную физическую среду	Внешние удаленные сервисы Метод грубой силы или полный перебор	Черви, трояны, матрешки	Подмена сетевых служб	Распространенные порты	Подключение через прокси Собственный криптографический протокол Многократное проксирование Многоуровневое шифрование Средства удаленного доступа	Проброс портов	Удаление подключений к сетевым ресурсам Запасные каналы Скрытие конечного адреса соединения Многоступенчатые каналы
Оперативная память									
Обнаружение процессов	Обнаружение процессов	Дампинг учетных данных Захват ввода	EWM-инъекции Чтение файлов с помощью логических смещений файловой системы Process Doppelganging		-	-	Cold boot attack Уязвимость Rowhammer	-	-
Операционная система									
Секретные ключи Обнаружение учетных записей Обнаружение файлов и каталогов Раскрытие парольной политики Обнаружение групп	Дампинг учетных данных Учетные данные в Реестре Захват ввода Автоматизирован	Эксплойты для получения учетных данных Форсированная аутентификация Отравление LLMNR/NBT-NS Kerberoasting DLL-библиотеки фильтров паролей Автоматизированная	Внешние удаленные сервисы. Выполнение с помощью эксплойтов. CMSTP. Удаленный сеанс. Windows Management Instrumentation. Элементы панели управления.	Интерфейс командной строки PowerShell, Провайдеры времени	Целевые фишинговые вложения, ссылки и сервисы Валидные учетные записи Выполнение пользователем/	Windows Remote Management	Протокол удаленного рабочего стола Удаленные сервисы Связь через съемные носители	Распространение с помощью съемных медиаустройств Выполнение с помощью локального планирования задач Создание учетных записей Logon-скрипты	Скрытые файлы и папки Деобфускация/дешифровка файлов или информации Отключение средств защиты Маскарадинг

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
доступа Обнаружение информации о системе Обнаружение сетевых подключений Обнаружение системных сервисов	ный сбор Данные со съемных носителей	эксфильтрация	Выполнение через загрузчик модулей Windows. Драйверы LSASS. Regsvcs/Regasm. Regsvr32. Rundll32. Выполнение через подписанные бинарники, сценарии. Windows Remote Management. Модификация исполняемых файлов приложений «специальные возможности Windows». Модификация ключа AppCert DLLs. Windows Remote Management. Mshta. Перехват вызовов функций Windows API. Злоупотребление подсистемой совместимости приложений. Модификация компонентов Windows Authentication Package.		модификация ярлыков			Новые службы Автозапуск с помощью ключа Run Keys и папки «Автозагрузка» Winlogon Helper DLL Прокси-выполнение кода через подписанные бинарники/сценарии	

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
			Перехват ссылок и связей Component Object Model Hijacking. Перехват поиска DLL. IFEO-инъекции. Модификация существующих служб. Вспомогательные DLL утилиты Netsh. Модификация Port Monitors в Диспетчере печати. Захват SIP и Trust Provider. Security Support Provider. Слабости разрешений параметров служб в реестре. Windows Management Instrumentation Event Subscription. Манипуляции с маркерами доступа. Модификация исполняемых файлов приложений «специальные возможности Windows». Непрямое выполнение						

Разведка			Проникновение			Выполнение		Закрепление	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
			команд. Установка корневого сертификата. Модификация реестра. NTFS-атрибуты файла. Выдалбливание процесса. Hooking. Pass the Hash. Pass the Ticket						
Программное обеспечение									
Секретные ключи Обнаружение учетных записей Обнаружение программных средств обеспечения безопасности	Дампинг учетных данных	Эксфильтрация данных из ПО	Эксплоиты публичных приложений. Теневая загрузка. Протокол Dynamic Data Exchange. Выполнение через API. InstallUtil. Выполнение через подписанные бинарники, сценарии. Создание заданий BITS. Расширения браузеров. Выполнение через API. Эксплуатация уязвимостей средств защиты	Скриптинг Упаковка софта	Выполнение через доверенные утилиты разработчиков софта	PowerShell	Удаленное копирование файлов	Автозапуск в офисных приложениях Тиражирование через съемные носители	Timestomp Обработка XSL-скриптов
Аппаратная составляющая									
Сканирование парольной политики	Сбор информации о железе,	-	Аппаратные закладки компрометаци	Гипервизор	Доверительные отношения	-	Буткиты	Подмена mac-адресов	Резервный доступ Rootkit

Разведка			**Проникновение**			**Выполнение**		**Закрепление**	
Обнаружение	Сбор данных	Эксфильтрация	Эксплуатация	Вредоносный код	Социальная инженерия	Командование	Управление	Распространение	Скрытие
	файлах, ключах		я цепи поставок						

Приложение В. Копия свидетельства о государственной регистрации программы для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ

RU2021680188

ФЕДЕРАЛЬНАЯ СЛУЖБА
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ

ГОСУДАРСТВЕННАЯ РЕГИСТРАЦИЯ ПРОГРАММЫ ДЛЯ ЭВМ

Номер регистрации (свидетельства):
2021680188
Дата регистрации: 07.12.2021
Номер и дата поступления заявки:
2021669850 07.12.2021
Дата публикации и номер бюллетеня:
07.12.2021 Бюл. № 12

Автор(ы):
Котенко Игорь Витальевич (RU),
Саенко Игорь Борисович (RU),
Лаута Олег Сергеевич (RU),
Крибель Александр Михайлович (RU)
Правообладатель(и):
Федеральное государственное бюджетное учреждение науки "Санкт-Петербургский Федеральный исследовательский центр Российской академии наук" (RU)

Название программы для ЭВМ:
Программный компонент обнаружения аномалий сетевого трафика на основе принципов фрактального анализа данных

Реферат:
Программа предназначена для обнаружения аномалий в сетевом трафике с помощью фрактального анализа и нейронных сетей

Язык программирования: Python

Объем программы для ЭВМ: 8,9 Кб

Приложение Г. Копии актов о реализации результатов исследования

ИНТЕЛТЕХ INTELTECH

Публичное акционерное общество «Информационные телекоммуникационные технологии» (ПАО «Интелтех»)

ул. Кантемировская д. 8, Санкт-Петербург,
Россия, 197342 Тел. (812) 295-50-69,
Факс (812) 542-18-49

www.inteltech.ru E-mail: intelteh@inteltech.ru
ОКПО 07503490, ОГРН 1027801525608,
ИНН/КПП 7802030605/781401001

_____№_____

На №_____от_____

Утверждаю
Первый заместитель
генерального директора
по научной работе
_____И.А. Кулешов

АКТ
реализации
результатов научных исследований
Крибеля Александра Михайловича

Комиссия в составе: председателя – заместителя генерального конструктора Харченко О.В., КВН, доцента, членов комиссии: начальника отдела Машкина И.В., КВН, доцента; начальника отдела Лапицкого В.Ф., КТН, доцента, составила настоящий акт о том, что научно-технические предложения по разработке системы раннего обнаружения и классификации КА в сетевом трафике сетей передачи данных, разработанные Крибелем А.М., были реализованы в рамках опытно-конструкторской работы «Опорник».

Комиссия отмечает практическую значимость и новизну указанных результатов.

Председатель комиссии: О.В. Харченко

Члены комиссии: И.В.Машкин

В.Ф. Лапицкий

МИНИСТЕРСТВО ОБОРОНЫ
РОССИЙСКОЙ ФЕДЕРАЦИИ
(МИНОБОРОНЫ РОССИИ)
Федеральное государственное казенное
военное образовательное учреждение
высшего образования
**ВОЕННАЯ ОРДЕНОВ
ЖУКОВА И ЛЕНИНА
КРАСНОЗНАМЕННАЯ
АКАДЕМИЯ СВЯЗИ
ИМЕНИ МАРШАЛА
СОВЕТСКОГО СОЮЗА
С.М.БУДЕННОГО**
г. Санкт-Петербург
Тихорецкий пр. 3, 194064
Тел.247-98-63

УТВЕРЖДАЮ

Начальник Военной академии связи
генерал-лейтенант С.Костарев

«22» апреля 2022 г.

Акт

о внедрении результатов

диссертационной работы Крибеля Александра Михайловича

на тему «Выявление аномалий и классификация компьютерных атак в сети передачи данных на основе применения фрактального анализа и методов машинного обучения»

Мы, нижеподписавшиеся, председатель комиссии - начальник 32 кафедры, к.т.н, доцент Митрофанов М.В., члены комиссии: доцент 32 кафедры, к.т.н Васюков Д.Ю; старший преподаватель 32 кафедры, д.т.н Лаута О.С. составили настоящий АКТ о том, что результаты диссертационной работы Крибеля Александра Михайловича на тему «Выявление аномалий и классификация компьютерных атак в сети передачи данных на основе применения фрактального анализа и методов машинного обучения» были внедрены в практическую деятельность Военной академии связи и НИРы по следующим направлениям:

№ п.п.	Результаты диссертационной работы
1	2
1.	Модель выявления аномалий в сетевом трафике сети передачи данных в условиях компьютерных атак.

2.	Методика раннего обнаружения аномалий в сетевом трафике сети передачи данных
3.	Методика классификации компьютерных атак в сетевом трафике сети передачи данных
4.	Архитектура системы раннего обнаружения и классификации компьютерных атак в сетевом трафике сети передачи данных

Настоящий акт составлен в 3-х экземплярах.

ПРЕДСЕДАТЕЛЬ КОМИССИИ:

М. Митрофанов

ЧЛЕНЫ КОМИССИИ:

Д. Васюков

О. Лаута