Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт кибербезопасности и защиты информации

Работа допущена к защите
Директор Института
кибербезопасности и защиты
информации, д.т.н., проф.
_____ Д.П. Зегжда
«___»_____2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

ОБНАРУЖЕНИЕ СЕТЕВЫХ АТАК НА КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ НЕЙРОЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

по направлению подготовки (специальности)
10.04.01 Информационная безопасность

Направленность (профиль)
10.04.01_01 Математические методы компьютерной безопасности

Выполнил
студент гр. 4841001/90101 А.Д. Фатин

Руководитель
доцент ИКиЗИ,
к.т.н. Е.Ю. Павленко

Санкт-Петербург
2021

РЕФЕРАТ

На 94 с., 20 рисунков, 14 таблиц.

КЛЮЧЕВЫЕ СЛОВА: МНОГОМЕРНЫЕ ВРЕМЕННЫЕ РЯДЫ, IoT, НЕЙРОЭВОЛЮЦИОННЫЕ АЛГОРИТМЫ, NEAT, КИБЕРФИЗИЧЕСКИЕ СИСТЕМЫ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

Тема выпускной квалификационной работы: «Обнаружение сетевых атак на киберфизические системы с использованием нейроэволюционных алгоритмов».

Данная работа посвящена детектированию сетевых атак на киберфизические системы (КФС) средствами нейроэволюционных алгоритмов.

Объектом исследования настоящей выпускной квалификационной работы являются КФС разной степени гетерогенности.

Предметом исследования является процесс функционирования КФС в штатном состоянии и в состоянии проведения на целевую систему сетевых атак.

Задачи, решенные в ходе исследования:

1. Исследование наиболее распространенных способов представления данных, циркулирующих в КФС. Выделение основных преимуществ, недостатков и областей применения каждого способа.
2. Исследование наиболее распространенных методов детектирования сетевых атак на КФС, использующих проанализированные способы представления данных. Выделение основных преимуществ, недостатков и областей применения каждого метода.
3. Разработка метода детектирования сетевых атак на КФС, основанного на использовании нейроэволюционных алгоритмов, с учетом проведенного ранее анализа.
4. Создание программного средства, реализующего предложенный метод детектирования сетевых атак на КФС, с использованием современных программных технологий.

5. Экспериментальные исследования реализованного программного средства, оценка точности его работы.

В результате работы создан и реализован метод детектирования сетевых атак на КФС. Также проведена оценка точности метода. Принцип работы заключается в выявлении отклонений между текущими значениями состояния КФС и предсказанными результатами. Предсказание выполняется нейроэволюционным алгоритмом семейства NEAT.

THE ABSTRACT

94 pages, 20 figures, 14 tables.

KEYWORDS: MULTI-DIMENSIONAL TIME SERIES, IoT, NEUROEVOLUTION, NEAT, CYBERPHYSICAL SYSTEMS, INFORMATION SECURITY

The subject of the graduate qualification work is "Detection of network attacks on cyber-physical systems using neuroevolutionary algorithms."

This work is devoted to the detection of network attacks on cyber physical systems (CPS) by means of neuroevolutionary algorithms.

The object of this work is CPS of varying degrees of heterogeneity.

The subject of this work is the process of functioning of the CPS in the normal state and in the state of carrying out network attacks on the target system.

Tasks solved during the study:

1. Investigation of the most common ways of presenting data circulating in the CPS. Highlighting the main advantages, disadvantages and areas of application of each method.

2. Investigation of the most common methods for detecting network attacks on CPS, using the analyzed methods of data presentation. Highlighting the main advantages, disadvantages and areas of application of each method.

3. Development of a method for detecting network attacks aimed at CPS, based on the use of neuroevolutionary algorithms, taking into account the analysis carried out earlier.

4. Creation of a software tool that implements the proposed method for detecting network attacks on CPS, using modern software technologies.

5. Experimental studies of the implemented software tool, assessment of the accuracy of its work.

As a result of the work, a method for detecting network attacks on CPS was created and implemented. The accuracy of the method was also assessed. The principle of operation is to identify deviations between the current values of the state of the CPS and the predicted results. The prediction is performed by a neuroevolutionary algorithm of the NEAT family.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

IoT	Internet of things
IIoT	Industrial Internet of things
КФС	Кибер физическая(ие) система(ы)
АСУ ТП	Автоматизированная система управления технологическим процессом
SCADA	Supervisory Control And Data Acquisition
DMZ	Demilitarized Zone
NEAT	NeuroEvolution of Augmenting Topologies
CPPN	Compositional pattern-producing networks
НСС	Низкоуровневая составляющая системы
ВСС	Высокоуровневая составляющая системы
NEAT	Neuroevolution of augmenting topologies
DoS	Denial of Service
DDoS	Distributed Denial of Service
MSE	Mean squared error
ДВП	Дискретное вейвлет-преобразование
DWT	Discrete wavlet transform
WTMM	Wavelet Transform Modulus Maxima
MF-DFA	Multifractal Detrender Fluctuation Analysis
SA	Segregated Approach
ICA	Iteratively Coupled Approach
FCA	Fully Coupled Approach
HF	Hazard function
SM	Survival models
RNN	Reccurent Neural Network
GRU	Gated Reccurent Unit
LSTM	Long / Short Term Memory
DTN	Delay-tolerant networking

AE	Auto Encoder
NTM	Neural Turing Machine
PSO	Particle swarm optimization (algorithm(s))
ABC	Artificial bee colony (optimization / algorithm(s))
ACO	Ant colony optimization (algorithm(s))
НОД	Наибольший общий делитель
GCD	Greatest common divisor
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
ROC	Receiver Operating Characteristic
AUROC	Area Under Receiver Operating Characteristic

ВВЕДЕНИЕ

В текущих реалиях использование концепции промышленного интернета вещей в среде киберфизичеких систем (КФС), в том числе в системах SCADA и АСУ ТП, не вызывает сомнений. Рассматривая данную область, одной из первичных задач встает вопрос используемых способов представления данных, циркулирующих в КФС, возможные преимущества, недостатки и области применения данных способов.

В работах [1-3] рассматриваются новые способы представления данных КФС, однако существует огромное множество моделей и методов удовлетворения запросов конечного пользователя в области информационной безопасности КФС.

В данной работе проводится подробный анализ и сравнение существующих способов описания данных КФС, методов детектирования сетевых атак, направленных на КФС, разбор основополагающих подходов и решений в сфере безопасности КФС, а также даются рекомендательные дополнения к уже существующим подходам с рассмотрением новых.

Объектом исследования настоящей выпускной квалификационной работы являются КФС разной степени гетерогенности.

Предметом исследования является процесс функционирования КФС в штатном состоянии и в состоянии проведения на целевую систему сетевых атак.

В ходе написания настоящей магистерской диссертации были решены следующие задачи:

1. Исследовать наиболее распространенные способы представления данных, циркулирующих в КФС. Выделить основные преимущества, недостатки и области применения каждого способа.

2. Исследовать наиболее распространенные методы детектирования сетевых атак на КФС, использующие проанализированные способы представления данных. Выделить основные преимуществ, недостатки и области применения каждого метода.

3. Разработать метод детектирования сетевых атак на КФС, основанный на использовании нейроэволюционных алгоритмов, с учетом проведенного ранее анализа.

4. Создать программное средство, реализующее предложенный метод детектирования сетевых атак на КФС, с использованием современных программных технологий.

5. Экспериментально исследовать реализованное программное средство, оценить точность его работы и сформировать рекомендации по применению.

Данная работа делится на три части:

- в первой части будут рассмотрены способы представления данных КФС, их область применения, положительные и отрицательные стороны. Проводится сравнения с аналогами, подводятся итоги, краткие выводы и даются рекомендации по реализации и выбору конечных способов;
- вторая часть будет выполнена по плану первой, однако уже будут представлены методы детектирования сетевых атак, базирующиеся на способах представления данных КФС, описанных в первой части.
- третья часть включает в себя непосредственно создание, реализацию и оценку точности метода детектирования сетевых атак на киберфизические системы средствами нейрогенетических алгоритмов.

Говоря о методах детектирования сетевых атак, стоит упомянуть, что любая КФС функционирует на основе двух типов потоков – физического (низкоуровневого) и логического (высокоуровневого) [4, 5]. Анализ низкоуровневой составляющей системы (НСС), состоящий в обработке данных, полученных от измерительных приборов, датчиков и сенсоров, позволяет оценивать корректность выполнения системных процессов. Он также позволяет изучать в режиме реального времени возникновение и проявление аномального поведения на самых ранних этапах за счет отсутствия высокоуровневых абстракций и простоты доступа к первоначальным данным. Аналогичным образом, совместно с анализом НСС, выполняется анализ высокоуровневой составляющей системы (ВСС). Он обусловлен необходимостью учета логики

операций, в том числе детектирования аномального поведения в логическом пространстве, когда физические параметры остаются в корректном состоянии.

Как было отмечено ранее, на текущий момент существует большое множество различных методов, подходов и реализаций детектирования сетевых атак на КФС [6-13], однако предпочтение обычно отдаётся либо использованию методов анализа на базе статистических инструментов [7-9], либо применению методов машинного обучения [10-13].

Методы машинного обучения в большинстве случаев применяются вместе с многомерными временными рядами, чей математический аппарат позволяет достичь высокой степени достоверности получаемых результатов, высокой скорости реагирования, низкой величины ошибок первого и второго рода. Это происходит за счёт работы с непрерывно генерируемыми данными в НСС, которые представляются в виде многомерных временных рядов. Далее, агрегируя временные ряды в многомерные, данный подход позволяет более полно охарактеризовать поведение КФС в динамике и упростить дальнейшую обработку массивов данных.

В практической части данной работы рассматривается применение нейроэволюционного NEAT-алгоритма с использованием гиперкуба для анализа многомерных временных рядов, описывающих состояние КФС, на предмет выявления аномальных состояний.

Верификация метода проводится на наборе данных TON_IOT DATASETS [14]. Топология системы представляет собой структуру Интернета вещей (Internet of Things, IoT). Данные являются релевантными, проверенными и корректными, что позволяет использовать их для анализа и оценки точности рассматриваемого подхода.

Набор данных включает в себя состояния и передаваемые данные каждого из 7 устройств сети: каждое устройство оперирует двумя основными переменными и двумя второстепенными (загруженность и текущее значение состояния). Рассматриваемый период работы системы включает 1 период 48 часов функционирования в нормальном состоянии и 3 периода по 48 часов, в

течение которых дискретно проводились атаки разных типов на систему, в том числе атаки типа DoS, DDoS, Backdoor.

Данные собирались как с НСС, так и с ВСС, а именно, в конечную выборку данных вошли 4 базиса: состояние каждого объекта, мера загруженности каждого объекта, физические данные, измеренные объектом, и конечный получатель данных.

Полученные в ходе научно-исследовательской деятельности результаты были представлены на конференциях «Методы и технические средства обеспечения безопасности информации», «РусКрипто'2021», «Неделя науки – 2021 Санкт-Петербургского государственного технологического института» и «Algorithms and solutions based on computer technology», а также опубликованы в журналах «Automatic Control and Computer Sciences» (Q3, Q4) и «Проблемы информационной безопасности. Компьютерные системы», входящем в перечень рецензируемых научных изданий ВАК, в 2020 и 2021 гг. Также результаты научно-исследовательской деятельности удостоены получением стипендии Президента Российской Федерации по приоритетным направлениям и именной стипендии Владимира Потанина 2020/2021 г.

1 ИССЛЕДОВАНИЕ БЕЗОПАСНОСТИ КФС

Затрагивая тему безопасности киберфизических систем, необходимо кратко ознакомиться с основными понятиями, положениями, принципами и лучшими практиками в данной сфере. Краткому экскурсу в данную сферу и последующими дискуссиям на тему возможных подходов для обнаружения нарушений безопасности в КФС и посвящен текущий раздел.

1.1 Основные понятия и принципы КФС

Термин «Киберфизические системы» введен в 2006 г. директором национального научного фонда США Хелен Джилл для обозначения комплексов, состоящих из физических объектов, управляющих подсистем и исполнительных механизмов (контроллеров). Данные системы разделяются на несколько различных типов. Принципиально можно выделить следующие:

- IoT – объединяют в сеть устройства разного вида для возможности функционирования системы без участия человека;
- АСУ ТП – Автоматизированная система управления технологическим процессом;
- робототехнические системы критического назначения, позволяющие выполнять задачи, требующие абсолютной точности, без участия человека;
- беспилотные летательные аппараты и автомобили;
- медицинские устройства, использующиеся для поддержания жизни и здоровья человека;
- умные сети производства, трансформации и распределения электроэнергии, которые используют коммуникационные сети и технологии для сбора данных об эффективности технологического процесса; позволяют повысить уровень автоматизации, сократить расходы на служебный персонал, снизить уровень рисков, а также повысить устойчивость производства и распределения электроэнергетических ресурсов в режиме реального времени и с минимальным участием человека.

Приведенный список систем объединяет подход на основе интеграции физического пространства и киберпространства, идущий рука об руку с высокой степенью риска и серьезными (иногда необратимыми) последствия в случае нарушения безопасности работы КФС. Исходя из вышесказанного, можно сделать вывод, что киберфизические системы имеют огромное число вариаций, потому в рамках данной работы предпочтение отдавалось преимущественно системам типа SCADA, IoT, IIoT и АСУ ТП, однако, не умаляя общности, в данной работе термин «киберфизические системы» будет включать не только выделенные типы систем, но и также все, перечисленные ранее. Примеры разборов большинства систем, используемых в них способов описания циркулирующих данных и методов детектирования сетевых атак будут подробно рассмотрены в разделе 2 и в разделе 3.

1.1.1 Общий принцип работы

Системы управления и передачи данных используются в многих различных секторах и критических инфраструктурах, включая производство, распространение и транспортировку.

Типичная структура КФС содержит средства удаленной диагностики, множественные контуры управления и дублирования, (обычно) пользовательский интерфейс ввода-вывода, логирования и журнализации, средства обслуживания и диспетчерского управления. Зачастую реализация данных средства выполняется на множестве сетевых протоколов с использованием многоуровневых сетевых парадигм и архитектур.

Контуры управления обычно используют данные, получаемые с актуаторов, сенсоров и программируемых логических контроллеров. Под сенсорами в данном контексте понимается устройство, измеряющее некоторые (обычно) физического величины, свойства и/или параметра и затем отправляющее полученные данные фиксированных дискретных переменных в логические контроллер для последующей обработки. Далее логический контроллер обрабатывает полученные данные и генерирует необходимые

команды, базируясь на конечных алгоритмах и механизмах управления и принятия решений. Полученные команды поступают на вход актуаторов, которые используются в управлении контролируемыми процессами.

Конечный наблюдатель (обычно инженер-оператор) могут взаимодействовать с интерфейсами ввода-вывода и отображения информации для получения текущих данных о состоянии системы. Также возможна настройка вручную заданных значений или запланированных событий, изменение алгоритмов управления и/или функционирования. Пользовательский интерфейс дополнительно может использоваться для коммуникации с соседними предприятиями или удаленного управления последними. Утилиты диагностики и обслуживания используются для предотвращения, идентификации и восстановления из аномальной работы или сбоев.

Зачастую контуры управления выполняются в модульной иерархии: они могут быть как вложенными, так и выполнены каскадно, при этом заданные может существовать косвенная зависимость используемых значений между контурами, если система выполнена не в одноуровневом исполнении процесса. Контуры нижнего уровня обычно функционируют непрерывно с начала технического процесса и вплоть до его окончания, а время выполнения данного контура может составлять он нескольких машинных тактов до нескольких дней. На Рисунке 1 приведена типичная логическая структура КФС и операции, выполняющиеся на ней [15].

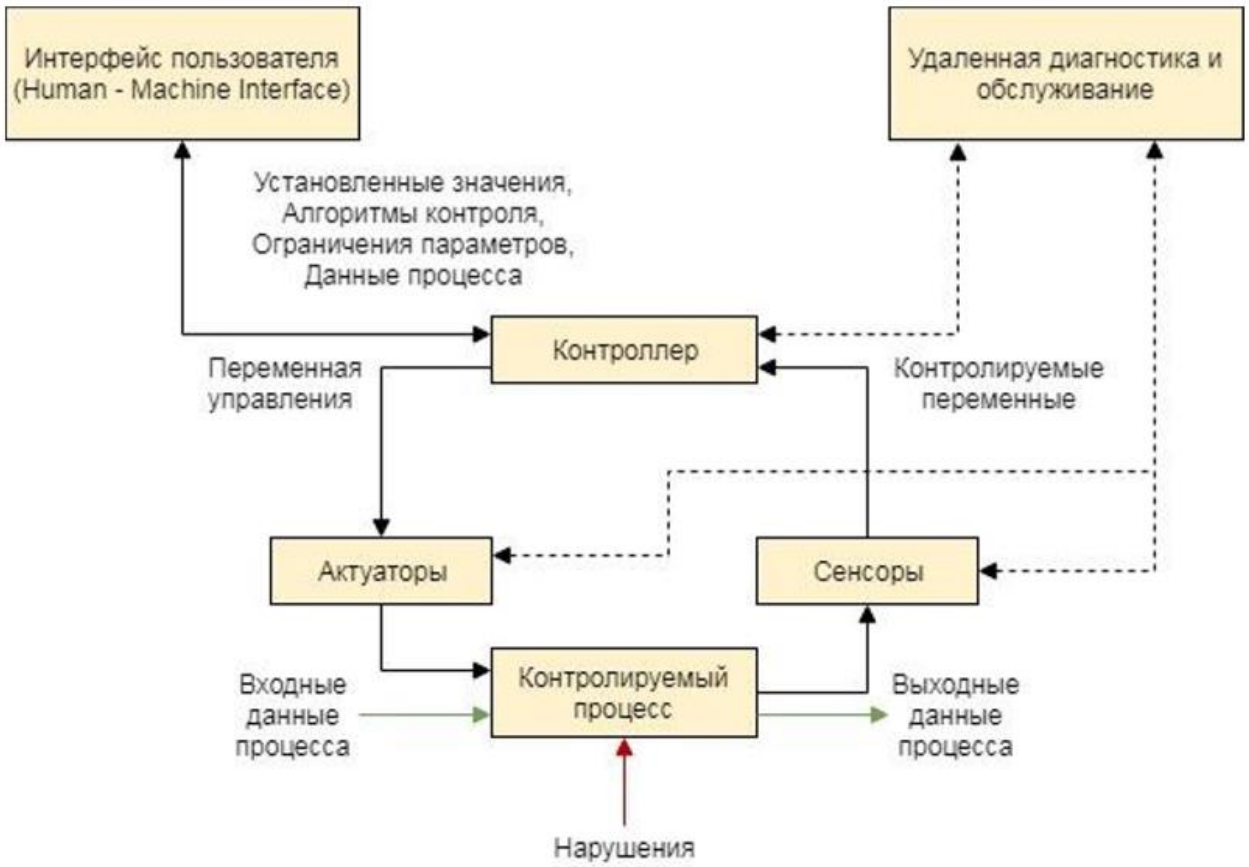

Рисунок 1 – Логическая структура КФС

1.1.2 Требования, предъявляемые к КФС

КФС в общем случае достаточно сильно отличаются от традиционных систем управления: они включают в себя, зачастую, большее количество дублирующих механизмов, средств принятия решений и восстановления ввиду отсутствия рабочего персонала или желания автоматизировать процесс. Дополнительно стоит отметить весьма различающиеся требования к целевым задачам, возможным рискам, их оценке и прогнозированию. Упомянутые системы могут иметь различные требования как к производительности и надежности, так и используемым операционным системам и приложениям, которые можно считать нестандартным в типичной сетевой среде. Средства безопасности должны быть реализованы таким образом, чтобы поддерживать целостность системы не только в течение нормальной запланированной работы,

но и в случае возникновения аномальных состояний, например, наличии кибератак.

На данный момент в КФС доступные Ethernet устройства и IP устройства заменяют ранее использованные проприетарные технологии, что увеличивает вероятность возникновения уязвимостей и инцидентов кибербезопасности. В данных системах внедряются новые решения для продвижения возможностей корпоративного подключения и удаленного доступа. Интеграция физического и кибернетического пространства добавляет новые возможности, но обеспечивает меньшую изоляцию системы от внешнего мира. Несмотря на то, что существующие решения для IT систем позволяют предотвратить большую часть угроз, КФС требуют особые меры по обеспечению безопасности [15].

1.1.3 Архитектура безопасности КФС

Разрабатывая логическую структуру функционирования и/или взаимодействия КФС стоит соблюдать (по возможности) принцип изолирование: отделение сети конечной системы от корпоративной, сети предприятия или глобальной. Функциональная составляющая циркулирующих команд в сетях различается – доступ в глобальную сеть, почтовые сервера, обмен данными и удаленный доступ обычно носят легитимный характер в сети предприятия, но явно не должны иметь места в КФС специально назначения. В сети предприятия допустимо отсутствие строгих процедуры контроля конфигурации сетевого оборудования, политик обновления и функционирования программного обеспечения и прочего. Однако, если допустить сетевой трафик КФС в сеть предприятия или в глобальную сеть, могут возникнуть дополнительные риски возникновения аномалий в работе КФС: возможность появления сетевых атак, нелегитимного доступа и изменения функционирования и прочих.

Практические соображения, такие как стоимость установки общего доступа подключения к сети Интернет или поддержание однородной сетевой инфраструктуры, часто означают, что требуется соединение между КФС и

корпоративными сетями. Это соединение представляет значительную угрозу безопасности и должно быть защищено. Если сети должны быть подключены, настоятельно рекомендуется разрешить только минимальные (по возможности, одиночные) подключения и устанавливать соединение через брандмауэр и DMZ (Demilitarized Zone). Серверы, обрабатывающие данные КФС, доступ к которым возможен из сети предприятия, рекомендуется размещать в DMZ. Рекомендуется использовать именно этот способ по причине ограничения доступа политикой безопасности, возможности конфигурирования портов и прочих необходимых настроек.

Из вышесказанного можно выделить следующие основные компоненты архитектуры безопасности КФС:

1. Сегментация сети. Разделение КФС на домены безопасности и отделение ее от других сетей, таких как корпоративная сеть. Каждая сеть разделяется на критические части после проведенного анализа операционного риска. Сегментация включает в себя разделение сети на более мелкие сети. Таким образом устанавливаются домены безопасности, которые обычно определяются как управляемые одним и тем же органом, обеспечивающие одинаковую политику и имеющие одинаковый уровень доверия. Сегментация может свести к минимуму метод и уровень доступа к конфиденциальной информации, доступ к КФС и конфигурации оборудования, а также значительно усложнить ее для злоумышленника.

2. Защита границ доменов безопасности. Устройства пограничной защиты управляют потоком информации между взаимосвязанными доменами безопасности, чтобы защитить КФС от кибератак, несанкционированных логических и физических ошибок и аварий. Передача информации между системами, представляющими разные домены безопасности с разными политиками безопасности, создает риск нарушения самих политики. Устройства пограничной защиты являются ключевыми компонентами конкретных архитектурных решений, которые обеспечивают соблюдение заданных политик безопасности. Организации могут изолировать компоненты

КФС и бизнес-системы, выполняющие различные задачи или бизнес-функции. Такая изоляция ограничивает несанкционированные потоки информации между компонентами системы, дополнительно давая возможность создать более многоуровневую защиту для целевых устройств и подсистем. Дополнительно стоит отметить возможность резервирования подсистем, выполняя их полное и независимое дублирование. Последнее, в свою очередь, возможно только при независимом функционировании подсистем или в разных условиях и сетях, не связанных друг с другом, или в системах разного уровня.

3. Межсетевые экраны. Данные программные комплексы позволяют выполнять администрирование сетевых потоков внутри целевых системы на основе выбранных политик безопасности.

4. Системы обнаружения вторжений. Системам обнаружения вторжений, реагирования на возникшие инциденты и системам детектирования аномального состояния особо внимание будет уделено непосредственно в разделе 3 и разделе 4.

5. Эшелонированная архитектура. Многоуровневая стратегия, включающая два или более различных перекрывающихся механизма безопасности, также известна как углубленная защита, желательна, чтобы минимизировать влияние отказа в каком-либо одном механизме. Стратегия глубокоэшелонированной архитектуры включает использование межсетевых экранов, создание демилитаризованных зон, возможности обнаружения вторжений наряду с эффективными политиками безопасности, программами обучения, механизмами реагирования на инциденты логической и физической безопасности.

6. Аутентификация и авторизация.

7. Мониторинг, журналирование и аудит. Данные элементы архитектуры необходимы для понимания текущего состояния КФС и подтверждения того, что система работает нормально, корректно и в штатном режиме.

8. Обнаружение аномальной работы, реагирование и восстановление системы [15].

1.1.4 Проблемы безопасности КФС

Использование вышеописанных механизмов безопасности позволяет разделять сеть предприятия на зоны с различной политикой безопасности, контролировать потоки информации, передающиеся из одной зоны в другую, обнаруживать несанкционированный доступ к сети предприятия, аномальные состояния и сетевые атаки. Также благодаря введение эшелонированной защиты предоставляет достаточно большой уровень безопасности благодаря избыточности. Однако некоторые угрозы безопасности КФС могут предоставлять возможность успешной атаки даже при наличии всех ранее перечисленных механизмов безопасности. Среди таких угроз: человеческий фактор, уязвимости существующих реализаций протоколов, уязвимости межсетевых экранов и оборудования для пограничной защиты, уязвимости нулевого дня и прочее. То есть злоумышленник имеет потенциальную возможность получить полный доступ к терминалу оператора и манипулировать входными значениями для нарушения нормального протекания физических и логических процессов системы.

Таким образом, возможны атаки, при которых ни один из механизмов безопасности не детектирует нарушение – единственными данными, на основе которых можно обнаружить аномалии, являются показатели текущего состояния процессов. Так как восстановление физических процессов является крайне сложным и дорогостоящим, то обнаружить аномалию необходимо как можно раньше.

1.2 Возможные подходы для обнаружения нарушений безопасности в КФС

Обнаружение аномалий в протекании процессов на киберфизической системе предприятия может основываться на способах представления данных внутри системы. Создание полноценной модели сложных физических процессов является весьма нетривиальной задачей: такой подход требует

глубокого понимания системы и ее реализации, а, следовательно, индивидуального подхода к каждой системе отдельно или к некому набору типичных систем. На практике часто ограничиваются методами детектирования аномальных состояний КФС, базирующихся на способах представления данных и их обработке. Непосредственно исследованию способов представления информационных потоков в КФС и посвящен раздел 2.

2 СПОСОБЫ ПРЕДСТАВЛЕНИЯ ДАННЫХ, ЦИРКУЛИРУЮЩИХ В КФС

По описательной природе способы представления данных в КФС принципиально можно разделить на использующие:

- многомерные временные ряды без преобразования с последующим анализом;
- сжатые, агрегированные или обработанные иными способами многомерные временные ряды;
- фрактальное представление топологии системы;
- графовые структуры разных видов.

Здесь и далее перечисляются основные способы описания данных в КФС и постановки задач, а также причины использования этих способов с выделением их преимуществ, недостатков и основных областей применения.

2.1 Многомерные временные ряды без преобразования с последующим анализом

В работах [2, 16, 17] предлагаемый подход рассматривает данные с НСС АСУ ТП: с актуаторов, ПЛК (программируемых логических контроллеров), сенсоров. Авторы данных работ выполняют преобразования поступаемых из КФС данных, преобразуются последние в многомерные временные ряды. Использование многомерных временных рядов обосновано следующим принципом: данный метод сохраняет большую информативность для дальнейшего анализа за счет сохранения связей между устройствами.

Для детектирования отклонений в процессах функционирования КФС средствами обученной модели прогнозирования многомерного временного ряда выполняется предсказание будущего состояния системы. На вход модели поступают показания текущего состояния, а на выходе получается предсказанный результат. Далее вычисляется ошибка – разность между реальным значением состояния КФС и предсказанным с помощью обученной модели. Если величина ошибки находится выше порогового значения, система фиксирует детектирование аномального состояния.

Согласно источнику [2], многомерный временной ряд представляет собой следующую совокупность:

$$X = \{X^{(1)}, X^{(2)}, \dots, X^{(m)}\},$$

где каждое значение в момент времени t_i представлено вектором:

$$X = \{x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}\}.$$

Первоначально данные, полученные от компонентов системы, нормируются: приводятся к единому виду и единому масштабу:

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}.$$

Предсказание следующего значения временного ряда представляет собой построение модели:

$$\hat{y} = model(x_{t-1}, \dots, x_{t-n}).$$

Для вычисления ошибки выполняется следующий набор действий:

- вычисляется разница между предсказанным $\hat{y}$ и наблюдаемым y значением для каждого признака:

$$e_t = |y - \hat{y}|;$$

- далее можно выявлять аномалии на основании условия

$$_{max}\ e_t > T,$$

где T – некоторый порог.

Используя данный подход, необходимо отметить, что все атаки, проводимые на системы, являются продолжительными во времени. Для фиксирования данного положения и повышения точности детектирования учитываются все зафиксированные максимальные ошибки (всплески) в некотором фиксированном по размеру окне времени:

$$Err_i = \sum_{t=i-w\ max}^{i} e_t > T.$$

В результате исследования в [16] частоты ошибок первого и второго рода равны 0,1 и 0,14 соответственно.

В работе [2] точность метода на тренировочных данных составила 90%, а на тестовом множестве − 82%. Значение MSE (среднеквадратичной ошибки) составило 0,06 и 0,03 соответственно.

Преимущества способа:

- высокая точность краткосрочного прогнозирования и, как следствие, хорошее выявление атак, укладывающихся в одно или несколько окон при отстройке на конкретных, заранее известных временных параметрах;
- вариативность применяемых анализаторов данных.

Недостатки способа:

- необходимость ручного подбора гиперпараметров предсказателя, например на основе подхода поиска по решетке;
- вариативная ширина окна выборки данных и, как результат, высокая вероятность ошибок для атак с сильно различимой длительностью: при узком окне высока вероятность пропустить длительные атаки, и наоборот, при достаточно широком окне возрастает вероятность пропустить кратковременные атаки;
- в общем случае данный подход показал не самую высокую точность выявления, однако один из самых высоких показателей универсальности.

Область применения:

Системы, имеющие достаточные вычислительные ресурсы для временного хранения и обработки многомерных временных рядов без преобразования и требующие глубокого анализа взаимосвязей с допустимым пренебрежением топологии.

2.2 Сжатые, агрегированные или обработанные иными способами многомерные временные ряды

Наиболее часто используемыми инструментами для обработки сырых временных рядов являются адаптивный алгоритм фильтра Калмана и дискретное вейвлет-преобразование DWT (ДВП). В большинстве случаев агрегирование и сжатие информации, полученной от киберфизической системы, выполняется данными решениями не только ввиду их вычислительной лёгкости и простоты автоматизации, но и за счёт удобства взаимодействия полученных результатов со статистическими инструментами

обработки данных на последующих этапах функционирования анализатора системы.

Далее приводится подробный обзор перечисленных методов с попарным выделением их преимуществ и недостатков.

2.2.1 Адаптивный алгоритм фильтра Калмана

Для прогнозирования состояния компонентов системы в [18] предлагается использовать математический аппарат фильтра Калмана [19, 20], нашедшего широкое применение в задачах предсказания координат движущегося в пространстве тела.

Показания компонентов системы представляются в виде хаотичной траектории движения некоторого тела с переменной скоростью в одномерном пространстве с использованием классических физических уравнений пути, скорости и ускорения материальной точки. Алгоритм применения фильтра Калмана для вычисления будущих значений представлен на Рисунке 2.

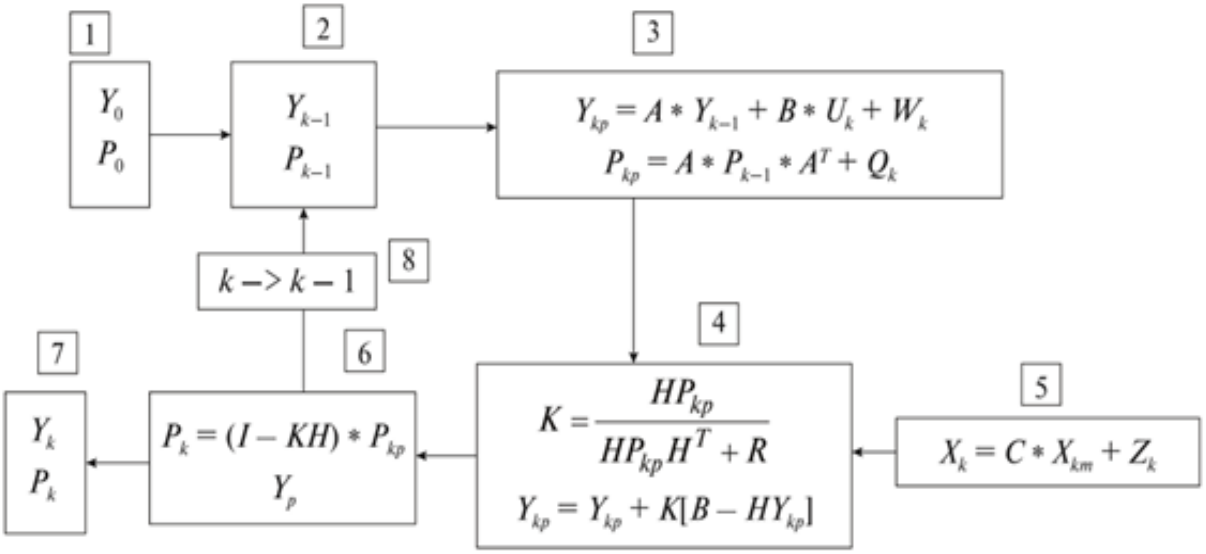

Рисунок 2 – Алгоритм применения фильтра Калмана

В блоке 1 описываются начальные оценочные значения: Y_0 – координата тела и P_0 – ковариационная матрица ошибок; в блоке 2 представлены значения тех же параметров на *k-1* шаге. В блоке 3 показаны уравнения оценки текущего состояния тела на основе состояния из предыдущего шага. Матрица Q – шум,

принимаемый за нулевую матрицу, исходя из физики задачи. Матрица ковариации P_{kp} обновляется на основе предыдущей с использованием матрицы эволюции процесса A. В блоке 4 описывается вычисление Калманова усиления: под переменной H понимается вспомогательная матрица, используемая в приведении усиления Калмана к матрице нужной конечной размерности; R – ошибки измерений. В блоке 5 используются данные от компонентов системы с вычисленными значениями скоростей, эти данные представлены матрицей C и шумовой матрицей Z. Блоке 6 описывает шаги получения финального результата фильтра. Блок 7 содержит значения Y_k и P_k, необходимые при расчетах. В блоке 8 описывается итерирование фильтра необходимое количество раз исходя из того, что вычисленные для шага k значения станут новыми значениями для шага k-1.

Показания датчиков системы представляются в виде временных рядов, фильтр Калмана применялся к каждому из рядов отдельно, полученные показания представляются в виде единого многомерного ряда, который в последствии используется для анализа данных и последующего предсказания состояния на следующем шаге.

2.2.2 Дискретное вейвлет-преобразование

Иными подходами руководствовались авторы в работе [21]. Ими предлагается проведение анализа трафика, полученного в КФС, средствами дискретного вейвлет-преобразования последовательности данных, полученных из инспектируемых пакетов. Дополнительно авторами выполняется статистический анализ различных параметров сетевых пакетов, полученных из КФС.

Вейвлет-преобразование, в общем случае, является обобщением спектрального анализа, и базируется на разложении получаемых данных на величины и конечные базисы, более удобные для конечного анализа и обработки. В отличие от Фурье-преобразования, вейвлет-преобразование имеет возможность анализа данных в частотно-временной области.

Ознакомившись с математическим аппаратом данного преобразования, становится понятно, что любую последовательность можно разложить на вейвлет-функции и базисные масштабирующие функции. Непосредственно само преобразование выполняется для разложения исследуемых последовательностей на две последовательности коэффициентов: $coeffA$ и $coeffD$. Коэффициенты аппроксимации $coeffA$ связаны с масштабирующей функцией, а коэффициенты детализации $coeffD$, согласно работе [22], связанны с базисной вейвлет-функцией. В случаях, когда уровень разложения принимает величину больше единицы, необходимо во всех последующих шагах выполнить вейвлет-преобразование полученных на каждом уровне коэффициентов аппроксимации $coeffA$.

После всех проведенных операций, на выходе алгоритма на уровне разложения j будем иметь набор коэффициентов $[coeffA_j, coeffD_j, coeffD_{j-1}, coeffD_{j-2}, \dots, coeffD_2, coeffD_1]$, который в последствии используется для анализа данных и последующего предсказания состояния системы на следующем шаге.

Преимущества обоих способов:

- агрегация, сжатие или обработка трафика с приведением к одноразмерным величинам упрощают и стандартизируют анализ;
- повышение производительности анализатора;
- отсутствие расходов на хранение данных ввиду использования только текущего состояния системы;
- возможность использовать разные способы решения, не обязательно завязанные на статистическом анализе;
- абстрагирование от топологии системы путем замещения сетевой структуры многомерными временными рядами с последующей обработкой;
- согласно [18], достаточная эффективность работы как с линейными, так и с нелинейными процессами.

Недостатки обоих способов:

- в общем случае статистические инструменты обработки данных не учитывают топологию системы и ее физические свойства, что может весьма сильно сказаться на точности анализа за счет снижения информативности;
- сложность унификации настройки анализатора одновременно на кратковременные и долговременные атаки за счет физической невозможности статистического аппарата.

Область применения:

Пониженные требования к ресурсоемкости и унификации данных дополнительно стимулируют использование данного подхода в гомоморфных системах с малой степенью изменчивости, например в самоподобных и/или линеаризованных структурах, не сильно завязанных или не завязанных вовсе на собственной топологии. Наибольшую эффективность данный подход должен показать в системах, сильно подверженных статистически предсказуемым флуктуациям трафика, например в сетевых магистралях.

2.3 Фрактальное представление топологии системы

В [23] авторами используется метод мультифрактального анализа данных для выявления аномалий в трафике магистральных сетей. Согласно мнению авторов, данный подход в должной мере позволяет выявлять сетевые неполадки или атаки. В качестве сигнальных метрик используются значения характеристик мультифрактального спектра.

Под мультифрактал обычно понимают совокупность или множество фракталов, каждый из которых имеет свою конечную размерность и может быть ею охарактеризован. Под обыкновенным фракталом следует понимать множество или набор данных, который удовлетворяет неким критериям самоподобия и/или самовоспроизводимости. Описание мультифрактала в конечном счете сводится к цикличному применению последовательно чередующихся алгоритмов, повторно применяющих шаблоны фиксированной фрактальной размерности [24]. Если сетевой трафик можно охарактеризовать фрактально, то можно говорить о возможности его разделения на части,

удовлетворяющие своим собственным критериям самоподобия, как обычные фракталы.

Анализ трафика магистральных сетей на наличие мультифрактальных свойств реализуется вычислением функции мультифрактального спектра $f(\alpha)$ над полученными данными, обычно сведенными к временным рядам или агрегированным иным шаблонным образом. Данные параметры подбираются индивидуально под нужны конечного пользователя и, как следствие, могут сильно разниться в зависимости от атак, которые необходимо детектировать. Выделенные параметры в дальнейшем (обычно) обрабатываются посредством использования статистических инструментов. Анализируя значения α и $f(\alpha)$, получаемые в подвижном окне, можно судить об изменение информации, содержащейся в исследуемых временных рядах. В свою очередь, изменение мультифрактальных свойств или потеря возможности удовлетворить критерии самоподобия сигнализирует об изменении состояния трафика, что может характеризовать наличие атак или неполадок в конечной КФС. Каноничный вид мультифрактального спектра Лежандра, на котором обычно выполняется анализ, представлен на Рисунке 3.

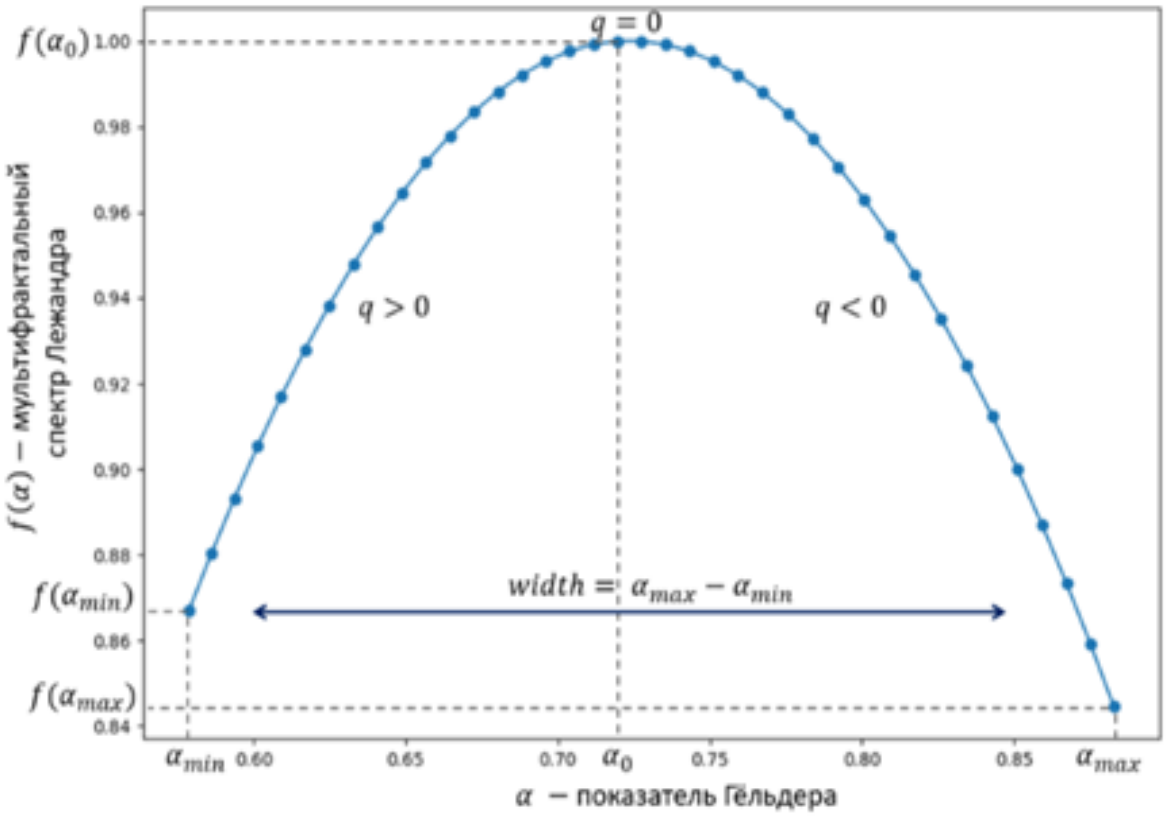

Рисунок 3 – Мультифрактальный спектр Лежандра

Из работы [25] достоверно известно, что по левой ветви мультифрактального графа можно судить о глобальных тенденциях изменения исследуемых данных, а по правой можно оценить изменение локальной составляющей подграфа. Дополнительно стоит отметить тот факт, что мультифрактальный анализ не сводится только к анализу ветвей: можно также проанализировать ширину мультифрактального спектра *width* для детектирования аномалий в сетевом трафике. Подобный подход нашел отражение в работе [26]. Исследование ширины спектра показывает высокую точность детектирования атак, так как описываемый параметр характеризует степень мультифрактальности множества – сужение спектра свидетельствует о повышении степени однородности исследуемого объекта.

Исходя из вышесказанного, в качестве исследуемых метрик, характеризующих наличие или отсутствие аномалий в сетевом трафике, авторами предлагается использовать следующие значения:

- *width* – ширина спектра;
- *widthright* – ширина правой «ветви»;
- *width*left – ширина левой «ветви»;
- *highleft* – высота левой «ветви»;
- *highright* – высота правой «ветви».

Дополнительно стоит отметить, что в работе [26] авторы проводят вычисления ширины мультифрактального спектра, используя метод максимумов модулей вейвлет-преобразования WTMM (Wavelet Transform Modulus Maxima). Однако в работе [27] отмечается, что оценки характеристик, полученные данным методом, являются менее устойчивыми и имеют явно больший разброс значений в сравнении с методом детрендированного флуктуационного анализа MF-DFA (Multifractal Detrender Fluctuation Analysis). Исходя из вышесказанного, авторами рассматриваемой статьи было принято решение ширину мультифрактального спектра сетевого трафика вычислять с помощью метода MF-DFA [28, 29].

Преимущества способа:

- возможность детектирования как кратковременных, так и долговременных атак на систему.

Недостатки способа:

- необходимость дополнительной обработки предварительных данных статистическими инструментами для повышения надежности работы механизма;
- малый набор доступных инструментов, используемых для решения задач, описываемых данным способом, а именно – почти единоличное главенство критериев самоподобия, упоминаемых далее;
- математическая сложность улучшения и видоизменения, ограниченная статистическими преобразованиями и критериями самоподобия;
- сложность детектирования новых аномалий, слабо завязанных на уже выбранных параметрах.

Область применения:

Данный способ представления данных в КФС в должной мере подходит для детектирования разнящихся по времени существования атак в сетях с высокой степенью самоподобия. Степень эффективности работы напрямую зависит от возможности свести топологию системы к фрактальной и качеству выбора заранее определенных параметров. Ограниченность способа не позволяет ему в должной мере конкурировать с прочими описательными подходами, однако данное решение, при условии выполнения всех необходимых рекомендаций, должно показывать наилучший результат в своей нише.

2.4 Графовые структуры разных видов

По причине обилия существующих графовых структур здесь и далее будут рассматриваться наиболее популярные и эффективные топологии в области описательных способов представления данных в киберфизических системах.

В частности, будут рассмотрены:

- классические графы;
- динамические графы;
- событийные графы;
- сигнальные графы.

2.4.1 Классические графы

Данный тип графовых структур наиболее общий, простой и разносторонний. Как следствие, данные графы являются базисом для большинства последующих графов или вспомогательным устройством в иных описательных методах. Иногда, впрочем, используются некие модификации стандартных графовых структур, которые сложно отнести к новому виду за счет малого количества изменений. Данные структуры, ввиду своей простоты, хорошо себя зарекомендовали во многих сферах применения, в том числе и в рассматриваемой.

В работе [30] рассматривается возможность использовать классические гравофые структуры для моделирования сетевой инфраструктуры сложных крупномасштабных объектов (в том числе критического назначения. Также авторы дополняют классические графы целевой функцией объекта и унарными операциями над графом, отражающими кибератаки.

Методом исследования является представление сетевой инфраструктуры исследуемого объекта в виде ориентированного графа, описание целевой функции как множества маршрутов на графе и систематизация кибератак в виде унарных операций над графом.

В результате исследования авторы получили графовую модель, описывающую сетевые инфраструктуры сложных крупномасштабных объектов с учетом выполняемой ими целевой функции.

Также авторами представлены различные типы целевых функций (ввиду большого объема не показанные здесь), описаны возможные проводимые кибератаки на сетевую инфраструктуру исследуемых объектов.

Полнота, четкость и корректность разработанной модели подтверждается сформулированной и доказанной теоремой.

Преимущества способа:

- простота и наглядность с сохранением принципиальности системы;
- малая требовательность к ресурсам;
- обширная существующая алгоритмическая база;
- идейное продолжение сетевой структуры и, как результат, наибольшая рентабельность при анализе топологии сетевых КФС.

Недостатки способа:

- недостаточная вариативность – либо приходится добавлять новые операции/функции/способы отображения, либо видоизменять и дополнять используемый математический аппарат;
- появление сложных зависимостей в случае гетероморфности (разнородности) структуры – например, *tg*-графы, в которых весьма велика

важность направления обмена данными, тип и количество существующих реберных связей.

Область применения:

Повсеместное применение элементов данной топологии в структурах, не сильно завязанных на сложных связях и/или отображениях, стимулирует и дальше использовать подход, описанный в данном методе, однако сам по себе способ описания не является исчерпывающей и зачастую требует дополнений и/или видоизменений.

2.4.2 Динамические графы

Динамический граф есть набор или упорядоченное во времени множество классических графов, в котором операция перехода от *i*-го графа к *i+1*-му графу определяется функцией времени и в простейшем случае является простым линейным отображением массива событий во времени на массив графов.

Данная графовая структура пока не нашла широкого применения в обеспечении безопасности КФС и детектировании аномалий, однако при должном подходе и наличии в системе достаточных вычислительных ресурсов должна себя хорошо зарекомендовать как простейший способ, позволяющий реализовать механизм цикличности обработки данных анализатором, который описывается далее.

Дополнительно стоит сказать, что хоть подобный подход и может требовать дополнительные ресурсы в системе на сохранение и обработку «слепков» состояния системы в графовом отображении, на практике подобное решение сводится к хранению списка изменений графовой топологии для минимизации расходуемого пространства на хранителях информации.

Зачастую элементы динамического графа встречаются в виде идейного продолжения классических графовых структур. Так, например, в работе [1] вводятся временные параметры генерации сообщения и дискретизируют многомерные временные ряды по временной оси, позволяя в первом представлении обобщить структуру до типичного динамического графа,

дополняя последнюю функцией значимого периода функционирования устройства $\varphi(t_i)$.

Преимущества способа:

- дополнительная степень свободы за счет использования оси времени;
- возможность более детализированного анализа и обучения анализатора на основе временных параметров;
- возможность буферизации данных для цикличного анализа системы.

Недостатки способа:

- повышенная требовательность к вычислительным ресурсам системы;
- необходимость обучать анализатор, так как другие решения малоприменимы для подобного подхода;
- необходимость хранить большие объемы данных (частично решается сохранением не «слепков» системы, а списков изменений в графе).

Область применения:

Системы, требующие повышенной надежности и/или имеющие сильную привязку ко времени работы и обладающие достаточными вычислительными ресурсами, в должной мере подходят для реализации подобной описательной системы.

Отдельно стоит отметить, что подобный подход достаточно подробно и просто позволяет описать цикличность работы анализирующего устройства, требуя при этом не самые высокие показатели производительности системы.

Является так называемым промежуточным вариантом между классическими и сигнальными графовыми структурами, вследствие чего удобоприменим в достаточно широких областях.

2.4.3 Событийные графы

В работах [31, 32] авторами рассматривается графово-событийная модель представления данных в КФС. Такой подход позволяет выполнить анализ поведения программ на основе событий, генерируемых в процессе функционирования системы. Также авторами представлена архитектура

системы, перечислен список событий, отслеживание которых выполняется на соответствующих им уровнях. Дополнительно проанализированы метрики, позволяющие оценивать подобие полученного графа и структуры графов заданных приложений. Приводятся результаты экспериментов, иллюстрирующие эффективность и точность разработанного подхода.

Стоит упомянуть, что в контексте информационной безопасности под событием понимается любое изменение состояния информационной системы, отражающееся на состоянии её безопасности.

В целях решения проблемы безопасности состояния КФС авторами в работе [31] определяется множество уровней ответственности, на которых предлагается разместить агентов. Под агентами авторы понимают программные компоненты или их реализации, способные собирать данные о происходящих событиях вышележащего уровня.

Хотя в общем случае решения, используемые в связке с данным способом описания данных в КФС, не всегда сводятся к оценкам самоподобия, данный подход применяется, основываясь на трех критериях самоподобия системы с алгоритмическим базисом реализации в структурах с повышенной сложностью топологии:

- нахождение максимального общего подграфа двух графов;
- нахождение максимального общего подграфа и минимального общего надграфа двух графов;
- функция вычисления расстояния.

Все три оценки рассматриваются далее.

Преимущества способа:

- обработка разноуровневых событий и целостность картины происходящих действий в системе;
- фрагментация системы на разные уровни ответственности;
- отсутствие обучаемого анализатора за счет введения критерия самоподобия (вариативно).

Недостатки способа:

- необходимость составления списка событий и, как следствие, открытость к уязвимостям, использующим новые триггеры/подходы;
- необходимость разработки агентов отдельно под каждую зону ответственности и дополнительные проверки покрытия системы всеми зонами ответственности, что влечет излишние расходы на согласование работы агентов в частности и описательной модели в целом;
- возможная косвенность анализа: в наше время задача определения изоморфизма двух графов является не решенной. Обычно анализ выполняется непосредственно по косвенный признакам.

Область применения:

Сложноструктурированные многоуровневые системы, имеющие явно нелинейную топологию, например КФС с гибридным исполнением сетевой инфраструктуры (такие как операционные системы, физико-биологические системы связи и взаимодействия, встраиваемые устройства, имплантаты и так далее), являются идеальными кандидатами для применения данного способа описания циркулирующих данных ввиду сложности стыковки разных логических и/или исполнительных уровней посредством использования других методов в частности и своей физической природы в целом.

2.4.4 Сигнальные графы

Анализируя безопасность КФС, стоит упомянуть сигнальные графы, так как последние имеют особое значение для физических систем и их моделирования. Под сигнальными графами понимают взвешенные ориентированные графы. Вершины таких графов логически отражают некоторые переменные, которые, в свою очередь, описывают состояния систем и подсистем. Вес каждой из таких вершин задает функцию времени и/или некоторые величины, отражающие соответствующую переменную (обычно состояние подсистемы КФС). В свою очередь, дуги сигнальных графов характеризуют связи между переменными: вес каждой такой дуги представляет

собой отношение (численное или функциональное), описывающее передачу сигнала от одной вершины к другой.

Стоит также отметить, что использование сигнальных графов является весьма распространенной практикой в теории цепей и механизмов, а также в оценке рисков, расчете возможного количества отказов системы в единицу времени и так далее. В следствии вышесказанного возможность применения сигнальных графов в области анализа безопасности и выявления аномалий в КФС весьма перспективна.

Отдельно стоит заострить внимание на максимальной близости способов описания физических процессов в сигнальных графах к их реальным истокам и, как результат, удобство использования данного способа в сферах безопасности КФС за счет четкого разделения высокоуровневой (логической) и низкоуровневой (физической) составляющих системы (ВСС и НСС соответственно) и операций, проводимых в системе, с последующей обработкой и решением задач разных уровней связности, как будет показано далее.

Преимущества способа:

- нативное отображение физических процессов на множество зависимостей в графовой структуре, удобство применения за счет готового математического аппарата, по сей день успешно применяемого в физических и технических областях связи;
- возможность логического разбиения и/или параллельной обработки/отображения двух разных типов данных, например ВСС и НСС;
- отсутствие обязательной привязки ко времени и, как результат выполнения данного условия, пониженные требований к вычислительным ресурсам системы.

Недостатки способа:

- первоначальная сложность описания системы за счет близости к физическим процессам;

- повышенная ресурсоемкость (при условии дополнительного использования цикличности эпох работы заданных физических устройств и/или анализатора в случаях необходимости повышенных требований к безопасности системы, как отмечается далее);

- как и в случае со всеми прочими графовыми структурами, возможность легкого расширения, масштабирования и дополнения структуры под нужды конечного пользователя.

Область применения:

Сложные КФС, безопасность которых в том числе должна основываться на многоплановом/многоуровневом анализе двух и более параметров с совместной обработкой данных, акцентирующих внимание на физике процессов, должны быть идеальными кандидатами для применения данной методологии. В качестве примеров можно рассмотреть системы АСУ ТП, IIoT, SCADA и так далее.

2.5 Необходимость учета высокоуровневой (логической) и низкоуровневой (физической) составляющих системы

Исходя из вышеописанных положительных и отрицательных сторон каждого из способов представления данных в КФС, не сложно заметить, что практически все сводятся к анализу высокоуровневой составляющей системы (ВСС), то есть к анализу циркулирующих пакетов в КФС с передачей наборов команд, и намного реже «сырого» представления физических данных элементов этой системы, то есть к низкоуровневой составляющей системы (НСС).

Очевидно, такой механизм в подавляющем большинстве случаев показывает высокую степень надежности, однако в должной мере не покрывает физическую составляющую атак на КФС.

Так, например, не используя дополнительный механизм проверки НСС, в большинстве случаев удается определить по ВСС изменения в системе за счет

отображения физических явлений на множество ВСС, но, опять же, такое отображение не всегда имеет место.

Исходя из сказанного, логично предположить необходимость введения второй составляющий анализатора системы, а именно – анализ «физического» или «низкоуровневого» трафика.

Опять же, раздельный анализ НСС и ВСС не сильно облегчает задачу, хоть и позволяет точнее проводить разграничение и с большей степенью надежности выявлять аномалии. Данный механизм следует дополнить цикличным анализом или набором действующих циклов физических величин, введя некую периодичность в работе анализатора.

Подобный механизм, его реализация и применимость описываются в разделе «Реализация цикличности анализатора на основе нейронных сетей».

2.6 Степень связности решения задачи с физической точки зрения

Следующим шагом логично рассмотреть степень связности модулей анализатора или самих анализаторов. Как принято в моделировании физических процессов, задачи обработки данных, акцентирующие внимание на связности, обычно делятся на 3 типа:

- Segregated Approach (SA) или несвязанное/раздельное решение;
- Iteratively Coupled Approach (ICA) или частичная/итеративная связность решения;
- Fully Coupled Approach (FCA) или полная связанность решения.

2.6.1 Segregated Approach

SA системы обычно применяются при исследовании и/или моделировании ряда задач, акцентирующих внимание на каких-либо конкретных явлениях, чей связностью с побочными явлениями можно пренебречь или рассчитать отдельно, не учитывая дополнительные зависимости с побочными явлениями.

В нашем случае подобное решение не имеет места, так как требуется учитывать в должной мере обе составляющие, которые практически всегда являются связанными как минимум односторонним отображением типа HCC → BCC.

2.6.2 Iteratively Coupled Approach

Подход, описывающий ICA решения, обычно основан на последовательном итеративном решении ряда задач. На *i*-м шаге выполняется решение задачи X. Полученные данные поступают в задачу *Y*, из *Y* в *Z*. После окончания последовательных расчетов на *i*-ом шаге данные из задачи *Z* поступает снова в задачу *X*, а шаг *i* становится *i+1*.

В случае ICA решения возникает проблема, схожая с описанной в части SA подхода. Так, например, сложность принятия решений связными модулями может быть нивелирована созданием промежуточного блока, принимающего сигналы от обоих модулей, анализирующего данные и реализующего выбор по обоим типам исходных данных, однако, опять же, в своем роде данный анализатор представляет 3 модуля: BCC, HCC и решатель. Видоизменение любого из модулей просто и легко делается за счет модульности разработки, однако все равно придется переписывать все модули, так как, например, модуль BCC может иметь новые данные, которые считаются аномалией только при возникновении схожих флуктуаций в модуле HCC, а HCC модуль будет считать эти флуктуации нормальными, так как не была обновлена его прошивка.

К тому же итеративный подход сильно сказывается на производительности за счет невозможности распараллеливания рекуррентных вычислений.

2.6.3 Fully Coupled Approach

Решение FCA модели – наиболее простое, удобное для доработки и использования, демонстрирующее максимальную точность. По сути, оно представляет собой единый анализатор, который, несомненно, так же сложно

переписывать, как и ICA, но его проще сертифицировать, обновлять зависимые базы данных (если они есть) и использовать на практике, так как нет необходимости в согласовании модулей. Также такой подход меньше нагружает систему, что зачастую является критическим критерием при выборе подхода к решению в реальных задачах промышленного масштаба, однако за это конечному пользователю придется расплатиться малой энергоэффективностью анализатора.

2.7 Итог по выбору способов представления данных

Исходя из всего вышесказанного, при рассмотрении КФС, к которым предъявляются повышенные требования в сфере информационной и физико-технической безопасности, рекомендуется акцентировать внимание на FCA связности задачи за счет необходимости учета обоих уровней связи.

В качестве способов, наиболее подходящих принципу FCA, можно выделить событийные графы, сигнальные графы и их сочетание с многомерными временными рядами. Первые рекомендуется использовать в случае повышенной сложности топологии и/или возможности проблем обработки стыковочных зон системы за счет гетероморфности структур другими методами; иначе же, в случаях рассмотрения классических КФС типа АСУ ТП, IIoT и SCADA, рекомендуется применять сигнальные графы ввиду их большей приспособленности к реализации цикличной обработки данных, описываемой в следующей статье. Для полноты анализа, сохранения принципа глубины связей обрабатываемых данных и распространения обратных связей рекомендуется использовать перечисленные графовые структуры с многомерными временными рядами.

Для большей наглядности основные преимущества и недостатки способов представления данных в КФС, дополнительные примечания и прочие матераиалы приведены в Таблице 1 и Таблице 2.

Таблица 1 - Основные черты способов представления данных в КФС

Задача\Способ представления	Временные ряды	Алгоритм Калмана	ДВП	Фракталы	Графы
Вариативность	+	+	+	-	+
Краткосрочные атаки	+	+	+	+	+
Долгосрочные атаки	+/-	+	-	+	+
Ручная настройка	+	-	-	+	+/-
Агрегация данных	+/-	+	+	+	+/-
Производительность	+/-	+	+	+	+/-
Учет нелинейных процессов	+/-	+	-	-	+/-
Учет топологии системы	+/-	-	-	+	+
Унификация задачи	+	-	-	-	+/-

Таблица 2 - Основные черты графовых структур

Задача \ Вид графов	Классические	Динамические	Событийные	Сигнальные
Вариативность	+	+	+	+
Краткосрочные атаки	+	+	+	+
Долгосрочные атаки	+	+	+	+
Ручная настройка	создание нового способа	-	+	+
Агрегация данных	возможна	возможна	-	возможна
Производительность	+	-	+	-
Учет нелинейных процессов	-	+	-	+
Учет топологии системы	+	+	+	+
Унификация задачи	+	+	+/-	+

3 МЕТОДЫ ДЕТЕКТИРОВАНИЯ СЕТЕВЫХ АТАК НА КФС

Анализируя механизмы, средства и математический аппарат, используемые в методах детектирования сетевых атак, направленных на КФС, принципиально можно выделить следующие подходы:

- оценка критериев самоподобия системы;
- предсказание состояния системы на основе статистических инструментов;
- предсказание состояния системы на основе машинного обучения.

Здесь и далее приводятся основные методы детектирования сетевых атак, основанные на исследованных в прошлом разделе способах представления данных, с выделением их преимуществ, недостатков и областей применения.

3.1 Оценка критериев самоподобия системы.

Большая часть описанных ранее способов легко и удобно применяется в случаях КФС со слабо выраженной гетероморфностью структуры, согласно описанным ранее преимуществам, логично использовать механизм оценки критериев самоподобия структуры, так как, как отмечалось авторами ранее, при должном соблюдении весьма жестких условий, выигрыш от использования подобных решений зачастую перевешивает строгость и жесткость структуры системы.

Так, например, в работе [1] авторами в качестве критерия самоподобия используется показатель Херста:

$$\frac{R}{S} = (\frac{N}{2})^H,$$

где H – показатель Херста, R – размах первых N значений ряда, S – стандартное отклонение.

Из [33, 34] известно, что процесс считается самоподобным, если выполняется следующее неравенство:

$$0.5 \leq H \leq 1.$$

Полученные ранее авторами данные обрабатываются автокорреляционной функцией для полученных временных рядов.

Аппроксимировав автокорреляционную функцию с помощью метода наименьших квадратов, в рассматриваемом примере авторы получили, что коэффициент наклона (взятый со знаком плюс) для нормального процесса равен $\beta = 0.33$, а для процесса с нарушениями $\beta = 0.65$. Существует оценка, связывающая коэффициент наклона β и показатель Херста следующим соотношением:

$$\beta = 2 - 2H$$

Таким образом в рассматриваемом примере показатель Херста отражает очевидные нарушения в работе системы.

Иными подходами руководствуются авторы работ [31, 32]. Как ранее упоминалось, используется 3 критерия самоподобия:

- нахождение максимального общего подграфа двух графов:

$$\Delta_1(G_1, G_2) = 1 - \frac{|MCS(G_1,G_2)|}{|G_1|+|G_2|-|MCS(G_1,G_2)|},$$

где $|MCS(G_1, G_2)|$ - максимальный общий подграф графов G_1и G_2; $|G|$- число вершин графа G; значение знаменателя соотношения представляет собой эквивалент объединения в теории множеств;

- нахождение максимального общего подграфа и минимального общего надграфа двух графов:

$$\Delta_2(G_1, G_2) = |msc(G_1, G_2) - |MCS(G_1, G_2)|,$$

где $|msc(G_1, G_2)|$ - минимальный общий надграф графов G_1и G_2;

- функция вычисления расстояния:

$$\Delta_3(G_1, G_2) = 1 - \frac{|MCS(G_1,G_2)|}{\max\{|G_1|,|G_2|\}},$$

где $|msc(G_1, G_2)|$ - минимальный общий надграф графов G_1и G_2.

Данный подход обоснован использованием метрик, основанных на графовых структурах, и полностью опирается на самоподобие в графовом смысле, нежели показатель Херста из предыдущей работы, оперирующий временными рядами.

Преимущества решения:

- высокая скорость обработки результатов;

- малая требовательность к вычислительным ресурсам системы;
- высокая точность оценки в малые интервалы времени – положительные результаты в случаях детектирования аномалий краткой длительности.

Недостатки решения:

- слабый анализ или полное отсутствие анализа в низкоуровневой составляющей системы (НСС);
- сложность детектирования длительных аномалий при условии их плавного появления и относительно невысокой скорости роста.

Область применения:

Данное решение находит свое место во всех, названных ранее способах описания данных, допускающих слабую гетероморфность системы и не требующих повышенной чувствительности к анализу НСС, а зачастую является единственным оптимальным, как, например, в случаях фрактального представления топологии системы.

3.2 Предсказание состояния системы на основе статистических инструментов

По причине существования различий между преимуществами и недостатками среди механизмов предсказания состояния КФС на основе статистических инструментов, данные подходы будут отдельно рассмотрены в следующих разделах. Особое внимание в данном анализе будет уделено сложности математического аппарата и применимости теории вероятностей вкупе с получаемой результативностью от применения данных подходов.

Анализ будет проводиться для поиска точек разладки на основе Байесовского онлайн алгоритма и использования коэффициентов множественной корреляции. Причины выбора – наибольшая популярность и наивысшая эффективность среди изученного множества инструментов статистического анализа и теории вероятностей в сфере обеспечения безопасности КФС.

Ввиду схожести подходов, их преимуществ, недостатков и областей применения, последние будут перечислены вместе, обобщая данный раздел.

3.2.1 Поиск точек разладки на основе Байесовского онлайн алгоритма

Среди методов обнаружения разладки (разладки - расхождения ожидаемой случайное величины с полученной на основе прогнозирования более чем на некую величину N) обычно принято использовать алгоритмы и методы, основанные на формуле Байеса. Так, например, в исследовании [35] авторами рассматривается возможность использования адаптированного байесовского онлайн-алгоритма для обнаружения точек разладки. Данный метод вполне логично использовать и для обнаружения сетевых атак и прочих аномалий в сетях в исследовании [3]. Авторами отмечается достаточно высокие показатели эффективности и невысокой ресурсоемкости при использовании в качестве замещения метода более «дорогих» подходов, анализирующих трафик магистральных сетей Интернет.

Байесовский онлайн-алгоритм строится на вычислении распределения длин прогона относительно поступающих данных, согласно [36]. Основа алгоритма заключается в использовании формулы Байеса:

$$\mathrm{P}(r_t|x_{1:t}) = \frac{\mathrm{P}(x_t|r_{t-1}, x_{1:t-1})\Sigma_{r_{t-1}} P(r_t|r_{t-1})P(r_{t-1}|x_{1:t-1})}{P(x_{1:t})},$$

где каждая из вероятностей, используемых в данной формуле, определяется следующим образом:

- $\mathrm{P}(x_t|r_{t-1}, x_{1:t-1})$ – вероятность того, что новые поступление данных удовлетворяют параметрам распределения в текущей длине прогона;
- $P(r_t|r_{t-1})$ – вероятность того, что длина пробега либо возрастает с приходом новых данных, либо возникает точка разладки;
- $P(r_{t-1}|x_{1:t-1})$ – значения, вычисленные на предыдущем шаге.

Авторами байесовского онлайн-алгоритма было предложено использовать семейства сопряженных априорных распределений для последующего упрощения вычислений вероятности. Непосредственно

рассматриваются следующие величины: правдоподобие данных $p(x|\theta)$, априорное распределение $p(\theta)$, правдоподобие данных $p(x)$ и апостериорное распределение: $p(\theta|x) = \frac{p(x|\theta)p(\theta)}{p(x)}$.

В тех случаях, когда формула применяется итеративно, возникает необходимость в том, чтобы апостериорное и априорное распределения были из одного и того же семейства распределений. Для такой цели вводятся семейства сопряженных априорных распределений. Если семейство априорных распределений $p(\theta|\alpha)$ при умножении на семейство правдоподобия $p(x|\theta)$ дает в результате апостериорное распределение $p(\theta|\alpha')$, то $p(\theta|\alpha)$ называют семейством сопряженных априорных распределений [37]. Параметры распределения параметров обозначаются буквой α и называются гиперпараметрами. Для нормально распределенных данных с известным средним сопряженным семейством распределений будет Гамма-распределение. Гиперпараметры апостериорного распределения вычисляются по следующим формулам:

$$\alpha = \alpha + \frac{n}{2},$$

$$\beta = \beta + \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{2}.$$

Функция вероятности для нормально распределенных данных с известным средним, в соответствии с источником [38], представляет собой распределение Стьюдента с 2α степенями свободы, нулевым средним и дисперсией, равной $\frac{\alpha}{\beta}$. В качестве вероятности $P(r_t|r_{t-1})$ предлагается брать значение функции отказов HZ (Hazard function). Данное понятие берется из класса статистических моделей, позволяющих оценить вероятность наступления событий SM (Survival models) [39]. Функция отказов может принимать различные формы, в зависимости от поставленной задачи. Предполагается, что длины пробега подчиняются экспоненциальному распределению, а потому функция принимает вид: $H(t) = 1/\lambda$, где λ – параметр экспоненциального распределения. Иными словами, вероятность

появления разладки не зависит от времени или от предыдущих точек разладки. Таким образом, возникает три возможных значения $P(r_t|r_{t-1})$:

$$P(r_t|r_{t-1}) = \begin{cases} (1-H) \text{ для } r_{t-1} = r_{t-1} + 1 \\ H, \text{для } r_t = 0 \\ 0, \text{в других случаях} \end{cases}$$

Вероятность $P(r_{t-1}|x_{1:t-1})$ представляет собой значения, вычисленные на предыдущем шаге. Иными словами, предложенный алгоритм является рекурсивным. Инициализирующее значение $P(r_0|x_0) = 1$, это говорит о том, что точка разладки в любом случае содержится в самом начале исследуемого ряда. Вероятность данных (evidence) $P(x_{1:t})$ в байесовском онлайн-алгоритме вычисляется как сумма вероятностей $P(r_{t-1}|x_{1:t})$. Таким образом, на выходе алгоритма получается матрица, содержащая вероятности длин пробега r_t для каждого момента времени t. В итоге, если наиболее вероятная длина пробега r_t принимает значение, равное 0, это сигнализирует точку разладки в момент времени t.

3.2.2 Использованием коэффициента множественной корреляции

В большинстве работ, направленных на детектирование аномалий и сетевых атак в трафике, использующих вейвлет-анализ, авторами используется оценка статистических свойств вейвлет-коэффициентов. В случае использования дискретного вейвлет-преобразования обычно используют коэффициенты детализации, поскольку последний позволяет с большей точностью детектировать атаки через локальные отклонения значений, которые в противном случае сглаживались бы на больших временных промежутках [40, 41]. Например, в исследованиях [42, 43, 44] авторами предлагается отслеживать именно изменение дисперсии коэффициентов, а, например, в исследовании [45] рассматривается использование критерия Фишера, позволяющего проверить равенство дисперсий двух фиксированных последовательностей.

Однако в работе [21] авторы предлагают использовать коэффициенты аппроксимации, которые характеризуют глобальный тренд исследуемого ряда или последовательности.

Отдельно следует отметить тот факт, что в большинстве работ, использующих математический аппарат временных рядов, анализируется статистические данные трафика: число пакетов фиксированных типов, интенсивность сетевых запросов и прочее. В рассматриваемой статье авторами предлагается анализировать конкретно ключевые поля сетевого протокола для обнаружения сетевых атак. Не сложно понять, что в зависимости от типа конкретной атаки для анализа должны быть выбраны как разные уровни модели OSI, так и разные ключевые параметры.

Для детектирования аномалий, в первую очередь, авторы выполняют разбиение первичного трафика на непересекающиеся временные окна – реализую дискретизацию данных. Из каждого окна извлекаются ключевые параметры фиксированного протокола и производится формирование временных рядов. Таким образом, проводя первичную обработку сырых данных, исследователь получает множество, составленное из окон следующего вида:

$$\text{Окно}_i = \begin{cases} \text{Временной ряд для параметра}_1 \\ \text{Временной ряд для параметра}_2 \\ \text{Временной ряд для параметра}_3 \end{cases}$$

На следующем шаге каждое полученное множество передается на вход дискретному вейвлет-преобразованию. После выполнения преобразования авторы получают коэффициенты детализации, необходимые для последующего анализа.

Чтобы охарактеризовать поведение сетевого трафика, разбитого на множества окон, авторы предлагают измерять степень зависимости различных ключевых параметров сетевого пакета. В роли такой метрики можно использовать коэффициент множественной корреляции, который для последовательностей x, y, z вычисляется по следующей формуле [46]:

$$R_{y(x,z)} = \sqrt{\frac{r_{xy}^2 + r_{zy}^2 - 2 * r_{xy} * r_{zy} * r_{xz}}{1 - r_{xy}^2}},$$

где r_{xy}, r_{zy}, r_{xz} – парные коэффициенты корреляции, которые вычисляются следующим образом:

$$r_{xy} = \frac{\sum(x_i - < x >) * (y_i - < y >)}{\sqrt{\sum(x_i - < x >)^2 * \sum(y_i - < y >)^2}}$$

Анализ коэффициента множественной корреляции $R_{y(x,z)}$ и величина его отклонения от нормального значения может указывать на наличие сетевых атак или аномалий в трафике.

Преимущества решения (практически полностью совпадают с оценкой критериев самоподобия системы):

- высокая скорость обработки результатов;
- малая требовательность к вычислительным ресурсам системы;
- высокая точность оценки в малые интервалы времени – положительные результаты в случаях детектирования аномалий краткой длительности.

Недостатки решения (практически полностью совпадают с оценкой критериев самоподобия системы):

- слабый анализ или полное отсутствие анализа НСС;
- сложность детектирования длительных аномалий при условии их плавного появления и относительно невысокой скорости роста.

Область применения:

Данное решение достаточно легко и удобно применяется со всеми ранее описанными способами представления данных, допускающими любую гомороморфность системы и не требующими повышенной чувствительности к анализу НСС. Наиболее предпочтительными для применения системами являются те, в которых соблюдаются условия отсутствия повышенных требований к производительности, допустима низкая степень гетероморфности и по каким-либо причинам малоприменимы критерии самоподобия.

3.3 Предсказание состояния системы на основе машинного обучения

В сфере безопасности КФС из машинного обучения, согласно эмпирическому опыту авторов и сложившимся практикам данной области, предпочтение обычно отдается нейронным сетям разных конфигураций (RNN, GRU, LSTM, AE, NTM и прочие) и эволюционным алгоритмам (преимущественно генетическим, но также всречаются, например, PSO, ABC, ACO и прочие). К слову, хоть нейронные сети и заняли доминирующее положение в этой области, а генетические алгоритмы сыскали меньшую популярность за счет возможных проблем с преодолением локальных экстремумов, однако, например, в работе [47] генетические алгоритмы показали свою эффективность.

Выгода от использования машинного обучения в области решения задач безопасности КФС подтверждена в теории и на практике в таких работах, как [2], [16], [17].

Ввиду большого числа видов и методов машинного обучения, стоит отметить базовые подходы прогнозирования. Реализация и непосредственное обучение модели подробно описываются в разделе 4.

Принципиально выбранная нейронная сеть или генетический алгоритм сказываются лишь на точности, скорости и требовательности к ресурсам работы анализатора за счет своих внутренних устройств и принятых решений. Акцент же, в свою очередь, отдается нейронным сетям за счет максимальной гибкости настройки анализатора для каждой конкретной теоретико-описательной и физической модели. Так, например, в работе [16] подробно описываются причины выборов конфигурации нейронной сети, слои, степень просеивания и прочие необходимые детали.

Общий подход решения сформулированных ранее задач состоит в получении данных от системы, сопоставлении/преобразовании/отображении их согласно принятой модели и последующем предсказании будущего состояния системы. Полученный результат сравнивается с нынешним состоянием. В

случае разницы результатов, превосходящей некое пороговое значение, поведение системы считается аномальным.

Преимущества решения:

- высокая вариативность применяемых конструкций и, как результат, широкий выбор между скоростью, качеством и требованиям к ресурсоемкости системы;
- возможность наиболее глубокого и надежного детектирования аномалий в НСС посредством применения цикличности анализатора, описанного далее;
- возможность реализации наиболее глубокого анализа и повышенного уровня безопасности системы.

Недостатки решения:

- первоначальная сложность настройки анализатора;
- необходимость обучения системы;
- априори повышенные требования к ресурсам системы в сравнении со всеми прочими решениями;
- невозможность переноса обученной модели на новую топологию (в отличие от большинства других подходов), необходимость переобучения.

Область применения:

Данное решение применимо со всеми названных ранее способами представления данных КФС, допускающими любую степень гомоморфности (однородности) структуры системы, однако к последней предъявляются повышенные требования в области ресурсоемкости.

В случае необходимости повышенной чувствительности к анализу НСС данное решение легко дополняется механизмом цикличности.

Наиболее удобные сферы применения те, что не подразумевают частого изменения топологии сети с точки зрения смены проектов и/или их реализации. Например, функционирующие АСУ ТП, SCADA и IIoT, то есть классические крупномасштабные КФС.

В случаях частой реконфигурации параметров сети появляются дополнительные временные и вычислительные расходы ресурсов системы на переобучение.

3.3.1 Реализация цикличности анализатора на основе нейронных сетей

Также стоит отметить, что существуют конфигурации нейронных сетей, способные предсказывать не только единичное будущее значение системы, но и, за счет буферизации временных переменных, периоды. Данный подход позволяет решить задачу цикличности анализатора, а именно обучать нейронную сеть не последовательным набором данным, дискретизированным, например, по времени, а набором данных, соответствующим определенному циклу работы множества устройств.

Очевидно, что цикл обучения, то есть. ширину выборки временного интервала стоит задавать по наибольшему времени t_k одного цикла из всех устройств, рассматриваемых в системе, если данный цикл устройства является $GCD(t_1, t_2, \dots, t_n) = t_k$ (НОД$(t_1, t_2, \dots, t_n) = t_k$, наибольшим общим делителем), в противном случае длительность цикла задается произведением длительностей циклов N устройств таким образом, чтобы к концу цикла все устройства вернулись в свое начальное физическое состояние, иначе происходит наложение многомерных кривых циклов устройств и вызывается ложное детектирование аномалии.

Такой подход позволит находить физические аномалии НСС при отсутствии проявления аномального поведения в высокоуровневой составляющей системы (ВСС) даже в тех случаях, когда аномалия на уровне НСС не была выявлена механизмами статистического анализа или критериями самоподобия. Например, можно рассмотреть процесс закалки в металлургическом цехе. Так температура, положим, индукционной печи за время цикла составляет сложную кривую ввиду определенного технического процесса, которая ни в коем разе не может быть нарушена по причинам усадки металла или прочих физических явлений, возникающих в случае отклонения

работы системы от заданного алгоритма. В случае нарушения данных ВСС продолжает считаться легитимным за счет скомпенсированного изменения величин (положим, равного цикличного отклонения от средней величины многомерной кривой, измененной неким шумом со средним значением, стремящимся к нулю, добавленным вредоносным программным обеспечением), однако физический процесс нарушается. Так, например, сохраняется условная линия тренда или усредненные за определенный период значения, однако, в случае использования цикличности анализатор способен выявить отклонение конкретных физических устройств от заданной многомерной кривой и обнаружить аномалию.

3.4 Итог по выбору методов детектирования сетевых атак

Исходя из всего вышесказанного, при рассмотрении КФС, к которым предъявляются повышенные требования в сфере информационной и киберфизической безопасности, при наличии достаточной вычислительной мощности, авторами рекомендуется акцентировать внимание на решениях, основанных на применении машинного обучения ввиду повышенной вариативности и возможности применения механизма цикличности анализатора для дополнительного глубокого анализа НСС с целью максимизирования безопасности системы.

В иных случаях, например, при невозможности выполнения критериев достаточной вычислительной ресурсоемкости системы и/или возможности допущения либо только кратковременных, либо только долговременных атак, допустимы применения как статистических инструментов решения поставленных задач, так и использования критериев самоподобия. Последние, в свою очередь, рекомендуются именно в случаях малой гетероморфности системы для большей эффективности и надежности, либо в случаях мультифрактальности, когда можно раздельно применить критерии для каждой подсистемы, либо при необходимости детектировать разные по длительности аномалии, но невозможности использовать машинное обучение.

В отдельных частных случаях, например, в случае развертывания системы с периферийные вычисления, создания DTN сетей, сетей военно-оперативного назначения и прочих особых случаев, рекомендуется использовать модификации графовых структур ввиду легкости преобразования последних. Такое решение позволит обеспечить максимальную гибкость и привязку к весьма узконаправленным задачам в системе с чрезмерно высокой гетероморфностью. В случае малой вычислительной способности или существовании большой задержки рекомендуется в таких сетях использовать статистические инструменты анализа состояний промежуточных устройств и логических узлов. В случаях чрезмерно малой вычислительной способности стоит задуматься о событийной модели поведения и уровневых агентах, описываемых, например, в работе [31].

4 РАЗРАБОТКА И РЕАЛИЗАЦИЯ МЕТОДА ДЕТЕКТИРОВАНИЯ СЕТЕВЫХ АТАК НА КФС, ОСНОВАННОГО НА ИСПОЛЬЗОВАНИИ НЕЙРОЭВОЛЮЦИОННЫХ АЛГОРИТМОВ

Проведя тщательный анализ существующих способов представления данных в КФС, их преимуществ, недостатков, областей применения, а также существующих методов детектирования сетевых атак, можно перейти к непосредственному созданию и реализации собственного метода детектирования сетевых атак на КФС.

Описываемый метод будет основываться на обработке полученных временных рядов из актуаторов и сенсоров КФС, использовании модифицированного алгоритма NEAT-гиперкуба для предсказания последующего состояния системы, и вычислении ошибки между предсказанным и реальным значениями.

Непосредственно сам алгоритм работы NEAT-гиперкуба основан на симбиозе двух других механизмов: нейронных сетей и генетических алгоритмов, выполняющих конфигурацию нейронной сети. Основные моменты реализации будут описаны в следующих разделах.

Тестирование созданного и реализованного метода детектирования сетевых атак на КФС будет выполняться на наборе данных TON_IOT DATASETS [14].

4.1 Первичная обработка данных, полученных из КФС

Изначальные данные, полученные из TON_IOT DATASETS [14] выборки, были представлены в csv файлах. Обработка выполнялась на языке Python.

Выполненная обработка и агрегация данных включала в себя следующие шаги:

1. Унификация времени представления данных по времени с шагом в 1 секунду (выполнена нормировка по времени: усреднение данных, полученных за промежуток в 1 секунду, если такие существуют, либо дублирование данных в случае отсутствия таковых за промежуток в 1 секунду).

2. Присвоение идентификационных номеров каждому из 7 устройств (id, нумерация выполнялась произвольно, однако с сохранением логической связи «отправитель-получатель»).

3. Изменение показателей состояния для тех устройств, которые не могли измерять степень своей «загруженности», но поддерживающих возможность измерять степень разряда питающего устройства (в данном случае учитывалась скорость разряда за 1 час; для удобства данный показатель был сведен к изменению процентного состояния заряда в секунду).

4.2 Выбор способа представления данных, циркулирующих в КФС

Ввиду рассмотренных ранее преимуществ и недостатков существующих методов представления данных, было решено остановиться на использовании многомерных временных рядов. Основные причины данного выбора: вариативность применяемых анализаторов данных, возможность ручной настройки гиперпараметров решателя, а также высокая степень вариативности метода – возможность применять в гетерогенных системах различного типа.

В случае использования многомерных временных рядов обычно проводится обучение нейронной сети на валидных данных для предсказания будущего состояния системы и вычисления разницы (ошибки) между предсказанным и реальным состоянием. Путём анализа ошибки проводится детектирование аномальных состояний в системе.

Как упоминалось ранее, классический многомерный временной ряд представляет собой следующую совокупность:

$$X = \{X^{(1)}, X^{(2)}, \dots, X^{(m)}\},$$

где каждое значение в момент времени t_i представлено вектором:

$$\{X^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}\}.$$

Для удобства работы и упрощения проектирования начального паттерна связей (субстрата) гиперкуба исходные данные, полученные от объектов системы, нормируются следующим образом:

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}.$$

В состав рассматриваемой IoT системы вошло 7 устройств, причем каждый имел 4 базисные составляющие, то есть размерность многомерного временного ряда составила 28.

Для удобства и высокой оперативности работы метода частота дискретизации обрабатываемых данных взята $\Delta t = 1$ с.

На Рисунках 4-7 приводятся примеры отображения нормализованных данных о состоянии системы в течение 48 часов её функционирования в нормальном состоянии и в состоянии, включающем аномальное поведение.

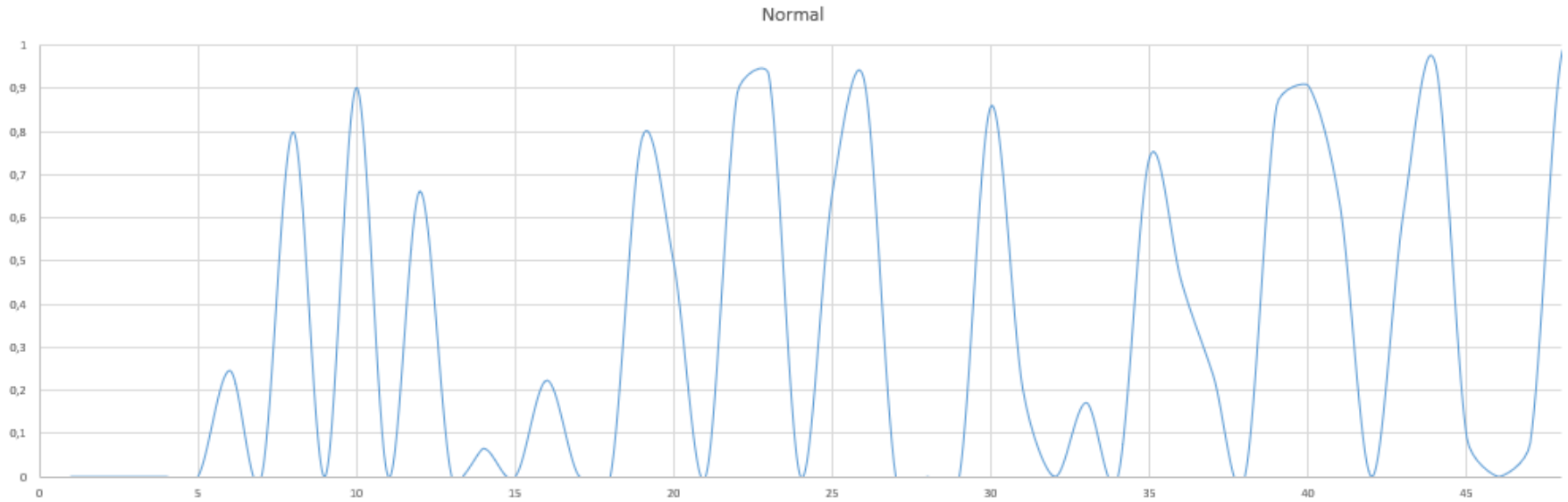

Рисунок 4 – Пример изменения данных в течение 48 часов работы системы в нормальном состоянии

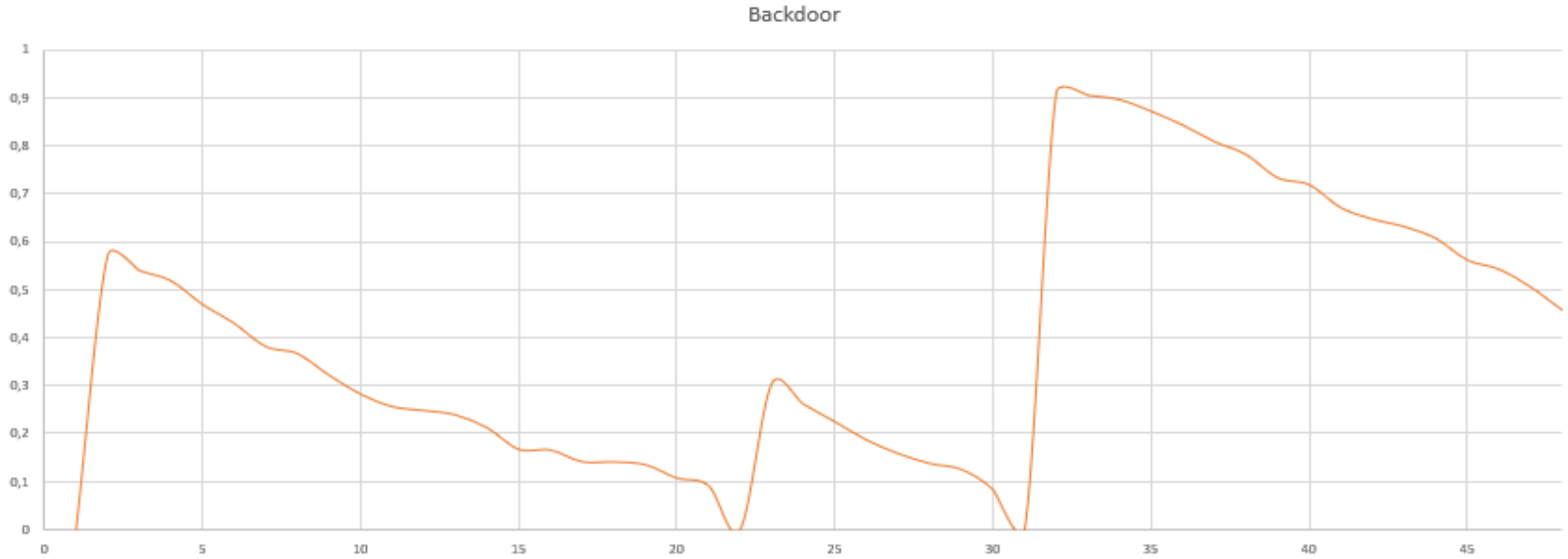

Рисунок 5 – Пример изменения данных в течение 48 часов работы системы с атаками типа Backdoor

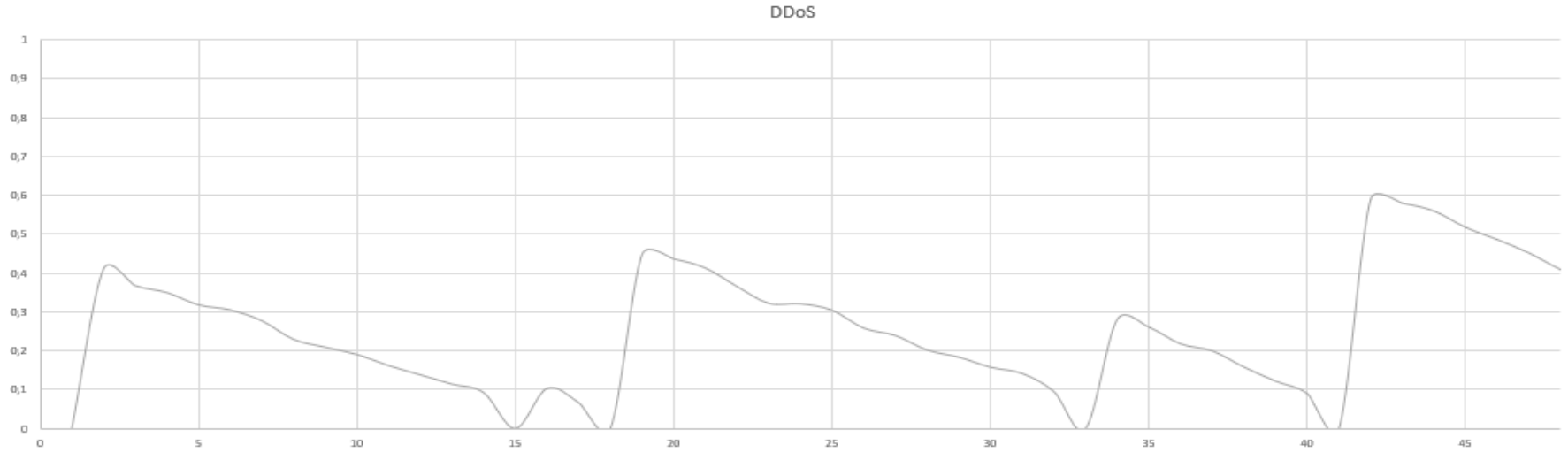

Рисунок 6 – Пример изменения данных в течение 48 часов работы с атаками типа DDoS

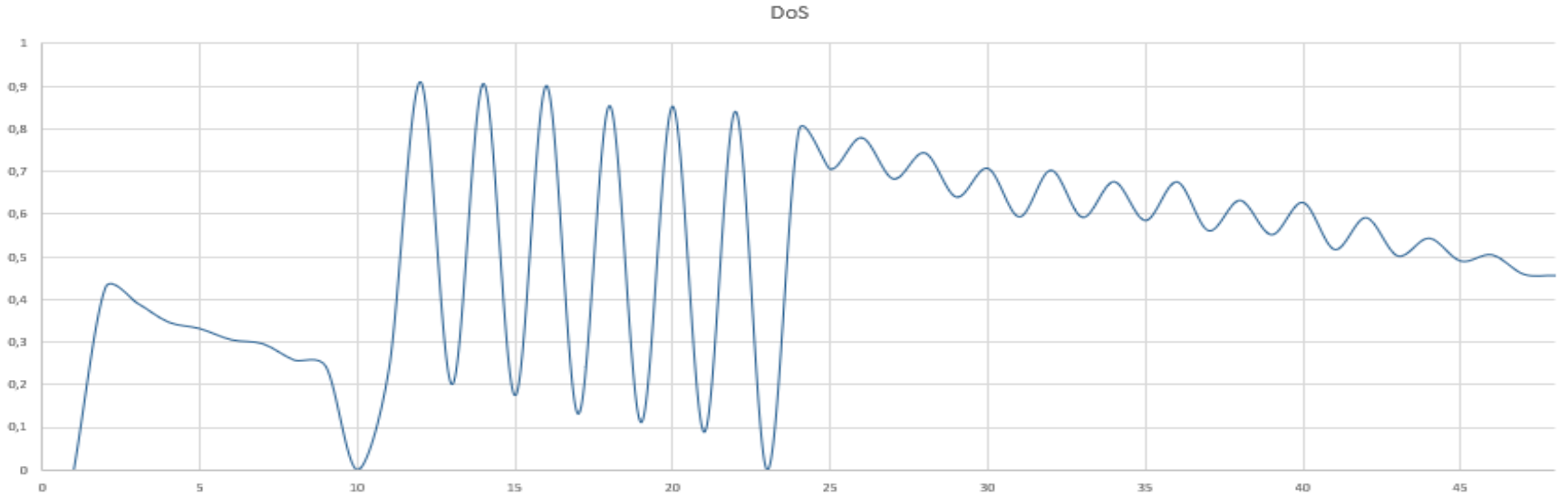

Рисунок 7 – Пример изменения данных в течение 48 часов работы системы с атаками типа DoS

4.3 Причины выбора и модификации нейроэволюционного алгоритма семейства NEAT

Исходя из проведенных ранее исследований становится очевидным, что многомерные временные ряды обычно используются с нейронными сетями ввиду того, что последние показали достаточно высокую точность детектирования сетевых атак вкупе с использованием этого способа описания данных в КФС.

Дабы избежать перечисленные ранее недостатки использования нейронных сетей, а именно первоначальную сложность настройки анализатора и сложность составления топологии нейронной сети, было решено использовать нейроэволюционный алгоритм NEAT-гиперкуб.

NEAT (NeuroEvolution of Augmenting Topology) – это генетический алгоритм для создания развивающихся нейронных сетей. Данный метод был разработан в Остине, в Техасском университете. Принцип работы алгоритма сводился к изменению весов и двухмерной структуры нейронной сети – поиску наиболее оптимального значения методами генетических алгоритмов.

Отдельно стоит отметить возможность модульного исполнения NEAT алгоритмов. Так как реализация метода сводится не только к выставлению заданных гиперпараметров, исполнитель имеет возможность конфигурировать как данные, используемые для обработки нейронной сетью, саму нейронную сеть, так и модифицировать генетическую составляющую алгоритма под свои нужды, чтобы создавать топологию именно той направленности, которую требует задача.

NEAT-гиперкуб (Hypercube-based NEAT) – это генеративное кодирование, которое развивает искусственные нейронные сети с принципами широко используемого алгоритма NEAT. Это новый метод развития крупномасштабных нейронных сетей с использованием геометрических закономерностей предметной области. Он использует сети создания композиционных шаблонов (CPPN, Compositional pattern-producing networks).

Сети создания композиционных шаблонов (CPPN) представляют собой разновидность искусственных нейронных сетей, архитектура которых определяется генетическими алгоритмами.

В то время как нейронные сети зачастую содержат именно сигмовидные и гауссовские функции активации, CPPN обычно базируются на более сложных функциях, так как первые не способны в полной мере решить задачу оптимизации. Выбор функций для канонического набора может быть смещен в сторону определенных типов шаблонов и закономерностей. Например, периодические функции, такие как синус, создают сегментированные шаблоны с повторениями, в то время как симметричные функции, такие как гауссовский, создают симметричные шаблоны. Линейные функции могут использоваться для создания линейных или фрактальных узоров. Таким образом, архитектор системы генетического искусства на основе CPPN может изменять типы генерируемых ею паттернов, выбирая набор канонических функций, которые необходимо включить.

Кроме того, в отличие от обычных нейронный сетей, сети композиционных шаблонов обычно могут быть применимы ко всем возможным входным данных, так что они могут представлять собой полную структуру. Поскольку они представляют собой композиции функций, CPPN фактически кодируют структуры с бесконечным разрешением и могут быть дискретизированы для конкретного решения с любым оптимальным разрешением.

Использование сетей CPPN удачным образом сочетается с многомерными временными рядами. Как будет показано далее, модификация алгоритма NEAT-гиперкуба – изменение размерности задач с двухмерной направленности на N-мерную позволяет в дальнейшем сильно упростить топологию конечной нейронной сети.

В общем случае алгоритм NEAT-гиперкуб работает с входной, выходной сетками и промежуточными слоями, сконфигурированными пользователем,

однако такой подход не позволяет в полной мере автоматически конфигурировать топологию конечной нейронной сети.

4.4 Модификация алгоритма NEAT-гиперкуб

Пример отображения топологии конечной нейронной сети на пространство гиперкуба и пример изменения топологии самой нейронной сети приводятся Рисунке 8.

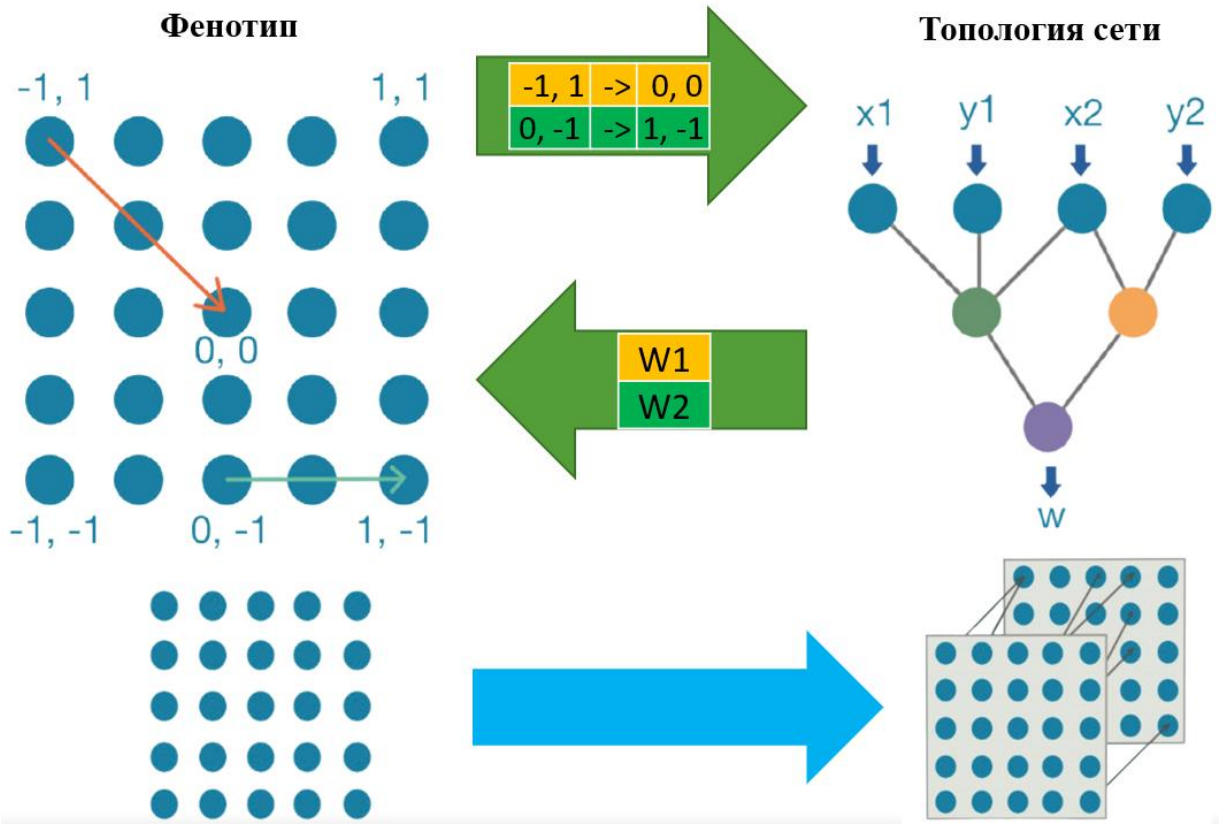

Рисунок 8 – Отображение нейронной сети на гиперкуб

Модификация алгоритма гиперкуба заключалась в добавлении возможности наращивания промежуточных слоев, которые иначе должны были быть сконфигурированы вручную. Данное изменение позволило практически полностью автоматизировать построение конечной нейронной сети.

В качестве фитнес-функции генетического алгоритма наращивания слоев была выбрана функция поиска новизны решения. Данный выбор обусловлен тем, что классические фитнес-функции в должной мере не справлялись и поиском оптимальной конфигурации промежуточных слоев.

4.4.1 Архитектура нейронной сети на основе модифицированного NEAT-гиперкуба

В качестве стартовой конфигурации нейронной сети была выбрана структура «сэндвич»: две двумерные плоские сетки с узлами входа и выхода, где один слой может выстраивать связи в направлении другого.

В качестве первичного субстрата используется форма многомерного временного ряда от времени t_0, полученного из данных 7 устройств, описанных ранее, и имеющая размерность 7 * 4 = 28.

В качестве входных данных используется многомерный временной ряд от времени t_i.

В качестве выходных данных на вершине гиперкуба получаем многомерный временной ряд будущего состояния системы во времени t_{i+1}.

Добавление любого произвольного гена узла или связи во время эволюции сети приводит к появлению нового глобального измерения вариации паттернов связей, то есть к появлению новых признаков через субстрат фенотипа. Новый способ изменения паттерна связи сводится, в конечном счете, к модификации генома структуры гиперкуба – изменению параметров методом моделируемой эволюции перестроения связей и/или узлов. Дополнительно ранее созданные связи в сети могут быть повторно использованы в качестве базиса создания нового паттерна связей для субстрата с более высоким разрешением, чем начальный, используемый для обучения. Таким образом, данный подход позволяет получить решение проблемы при любом разрешении сетки гиперкуба [48].

Вышеупомянутые свойства сделали алгоритм гиперкуба мощным инструментом для развития крупномасштабных искусственных нейронных сетей, имитирующих биологические объекты, а также позволили исправить проблему стагнации решения нейронных сетей за счет введения вариативности размещения узлов.

После модификации алгоритма нет необходимости строго задавать структуру нейронной сети, так как она может меняться во время эволюции благодаря генетической составляющей – наращиванию промежуточных слоев, изменению количества активных нейронов и существующих связей.

4.4.2 Типы используемых операторов мутации и кроссовера

На Рисунках 9-10 приводятся используемые операторы кроссовера и операторы мутации совместно с их принципами работы. Так как изменение в графе нейронной сети может быть сведено к изменению размещения вершин и изменению связей между вершинами, было решено ограничиться данным набором.

Применение операторов кроссовера и мутации в конечном счете сводится к следующим действиям:

1. Инверсия – побитовое (единичное) изменение связи, её веса и/или активности нейрона.
2. Изменение порядка – перенос существующего узла и/или связей в другую область. В конечном счёте сводится к переконфигурированию расположения связей.
3. Изменение значения – изменение веса связей и/или активности нейронов.
4. Изменение экспрессии – создание новых нейронов, связей, выстраивание дополнительных обратных зависимостей или их удаление (аналог Drop-слоя из классических нейронных сетей).
5. Одиночный, двухточечный и унифицированный кроссоверы в конечном счете позволяют «перемешивать» решения между узлами нейронной сети, то есть проводить реконфигурацию существующих связей и их весов без изменения их количества и веса.

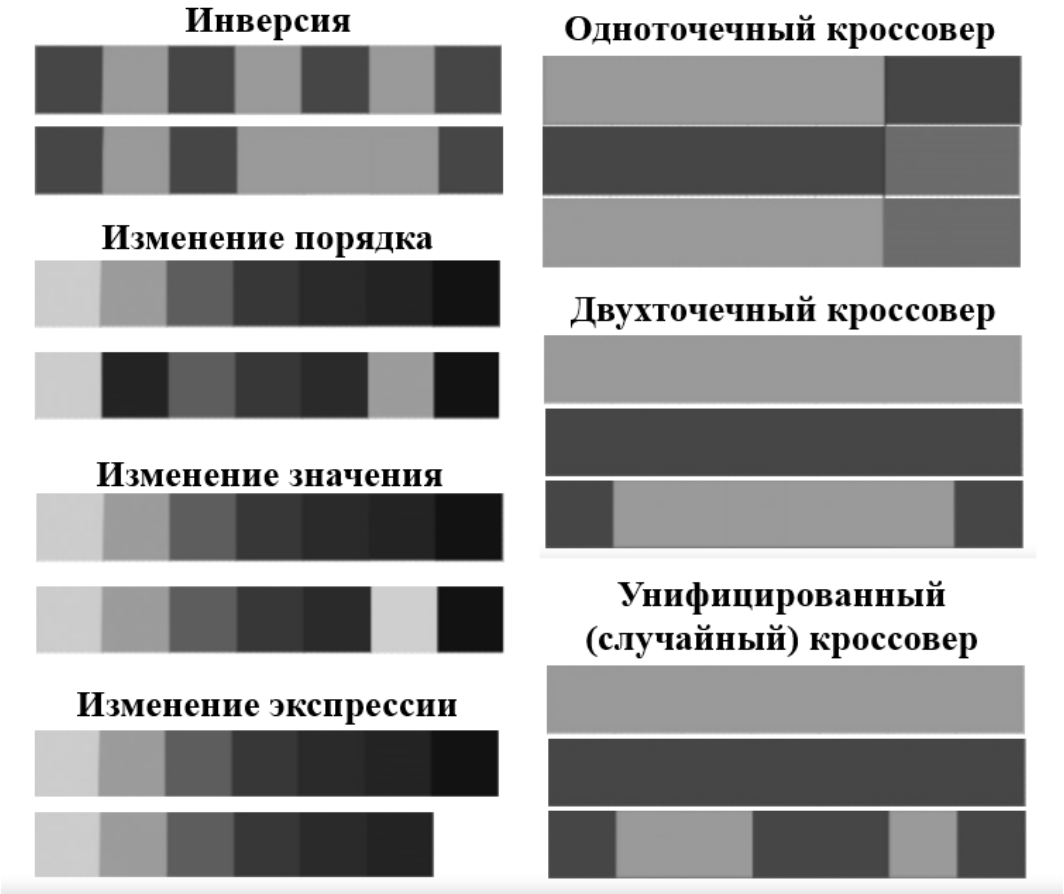

Рисунок 9 – Используемые операторы кроссовера

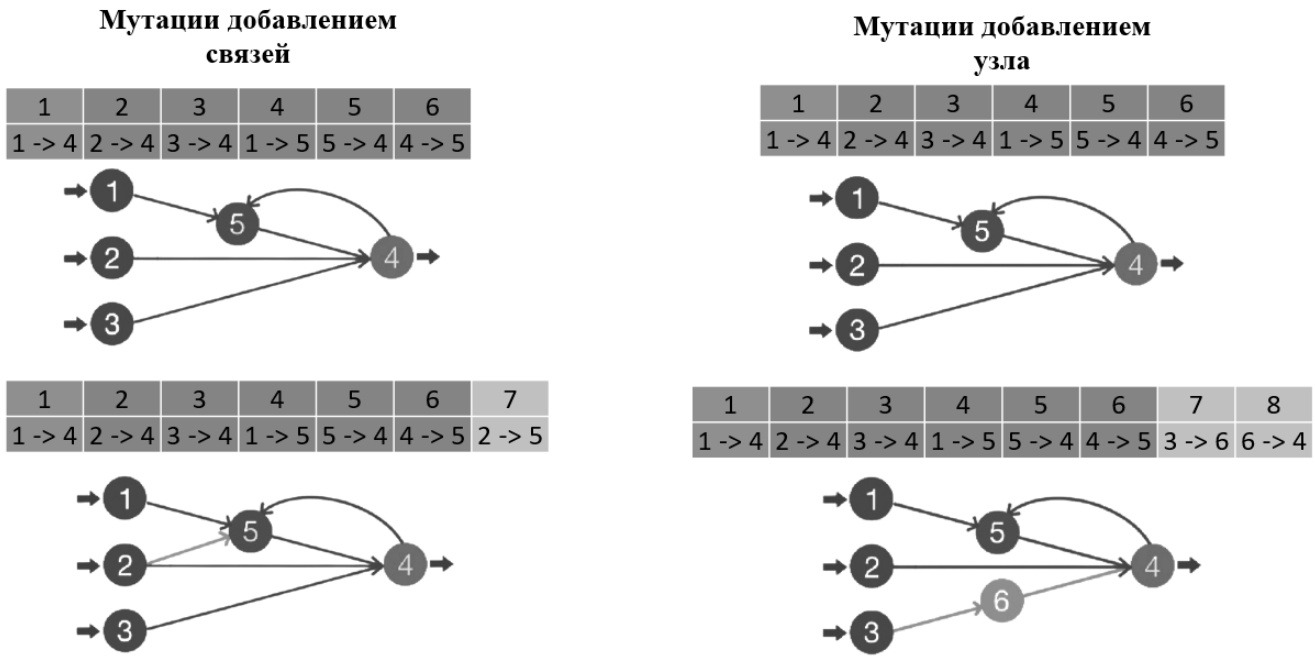

Рисунок 10 – Используемые операторы мутации

Также на Рисках 11-13 приводятся примеры осуществления реальной мутации фенотипа топологии нейронной сети. На Рисунках 11-12 приводится изначальная популяция субстрата, а на Рисунке 13 приводится полученный после мутации новый фенотип субстрата.

1	2	3	4	5	6	7	8
1 -> 4	2 -> 4	3 -> 4	1 -> 5	5 -> 4			2 -> 5

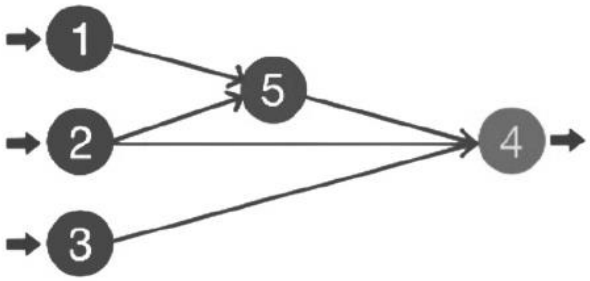

Рисунок 11 – Пример мутации фенотипа

1	2	3	4	5	6	7	8	9
1 -> 4	2 -> 4	3 -> 4	1 -> 5	5 -> 4	3 -> 6	6 -> 4		2 -> 6

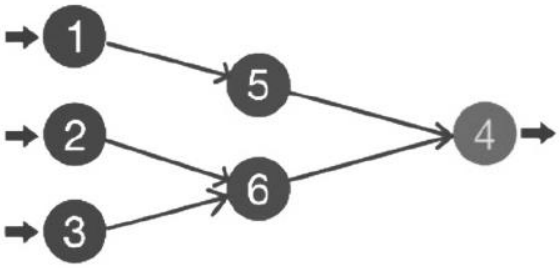

Рисунок 12 – Пример мутации фенотипа

1	2	3	4	5	6	7	8	
1 -> 4	2 -> 4	3 -> 4	1 -> 5	5 -> 4			2 -> 5	
1	2	3	4	5	6	7	8	9
1 -> 4	2 -> 4	3 -> 4	1 -> 5	5 -> 4	3 -> 6	6 -> 4		2 -> 6

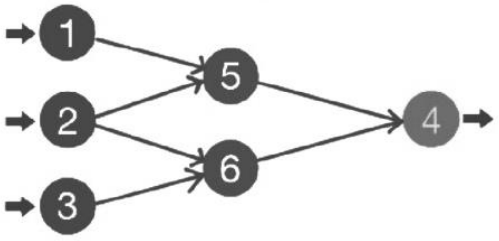

Рисунок 13 – Пример мутации фенотипа

4.5 Принцип работы метода детектирования сетевых атак

Описываемый метод основывается на обработке полученных многомерных временных рядов, составленных из данных, циркулирующих внутри КФС, предсказании будущего состояния системы средствами модифицированного нейроэволюционного алгоритма NEAT-гиперкурб и анализе возникающих ошибок – расхождения между реальными значениями состояния системы и предсказанными.

Методов включает в себя 2 этапа – подготовительный и рабочий. Подготовительный этап нацелен на автоматическое конфигурирование оптимальной топологии нейронной сети и подразумевает под собой следующие шаги:

1. Подготовка тестовых данных – нормализация и составление многомерных временных рядов.

2. Передача полученных многомерных рядов на вход нейронной сети, первично сконфигурированной пользователем.

3. Обучение нейронной сети на переданных данных и её реконфигурация генетической составляющей нейроэволюционного алгоритма до тех пор, пока не будет получена заданная точность на тестовых данных.

Рабочий этап подразумевает под собой непосредственное детектирование сетевых атак, направленных на КФС, и включает в себя следующие шаги:

1. Подготовка реальных данных функционирующей КФС – нормализация и составление многомерных временных рядов.

2. Передача полученных многомерных рядов на вход нейронной сети, оптимально сконфигурированной генетической составляющей нейроэволюционного алгоритма.

3. Предсказание будущего состояния системы нейронной сетью на основе полученных многомерных временных рядов.

4. Вычисление ошибки между предсказанным состоянием системы и реальным.

5. Фиксирование наличия или отсутствия атак на КФС на основе полученной ошибки.

Схема работы метода представлена на Рисунке 14.

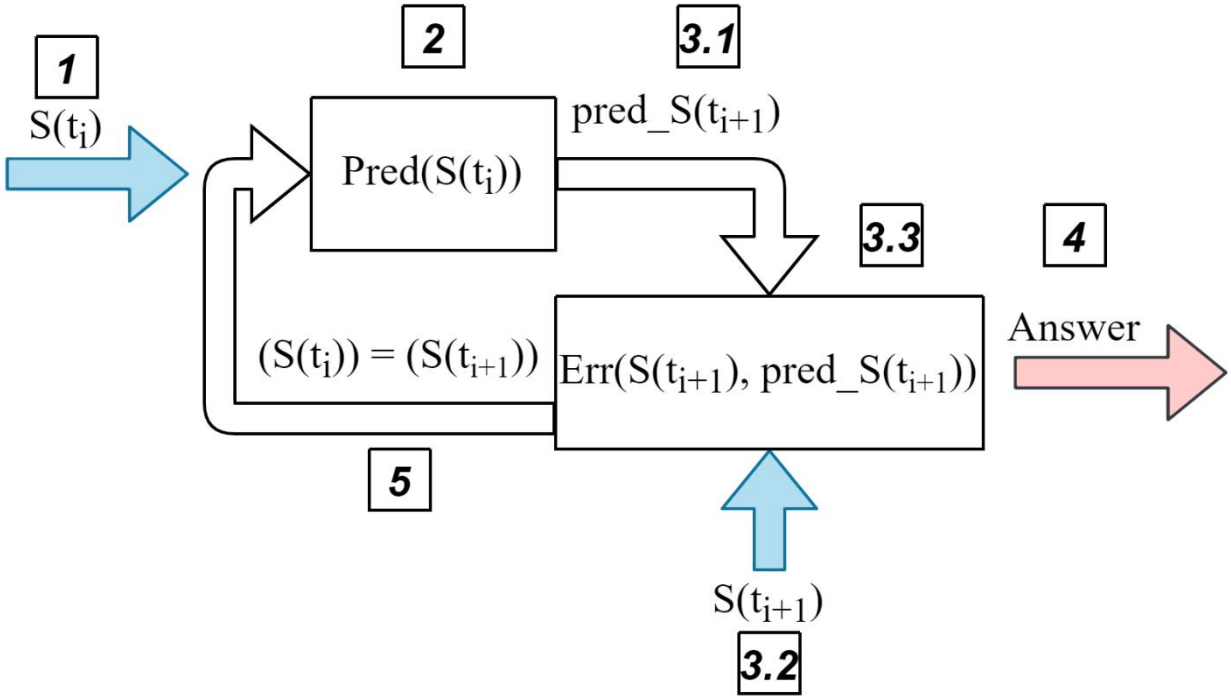

Рисунок 14 – Принципиальная схема работы метода детектирования сетевых атак

На этапе 1 сформированный многомерный временной ряд $S(t_i)$ от времени t_i подается на вход нейронной сети. На этапе 2 выполняется операция предсказания будущего ряда $pred\_S(t_{i+1})$ на основе ряда $S(t_i)$, где Pred() – функция предсказания, выполняемая нейронной сетью. На этапах 3.1 и 3.2 предсказанный многомерный ряд $pred\_S(t_{i+1})$ и многомерный ряд, полученный из реальных показателей системы $S(t_{i+1})$, поступают в блок сравнения. На этапе 3.3 вычисляется разница между показателями и происходит накопление ошибки. На этапе 4 на основе сравниваемых данных в блоке 3.3 мы получаем ответ о наличии или отсутствии атак. На этапе 5 значения многомерного временного ряда от времени t_i заменяется значениями от времени t_{i+1}, после чего алгоритм повторяется.

4.5.1 Прогнозирование состояния системы

Как уже ранее отмечалось, данные, прошедшие процедуру нормализации, должны быть подвергнуты предварительной обработке: для каждой точки временного ряда определяется спрогнозированное значение, как показано на Рисунке 15.

Для предсказания последующего значения состояния системы через временной ряд необходимо выполнить операцию:

$$\gamma_{pred} = pred(x_{t-1}, \dots, x_{t-n}).$$

Операция предсказания выполняется средствами нейронной сети, сконфигурированной алгоритмом гиперкуба.

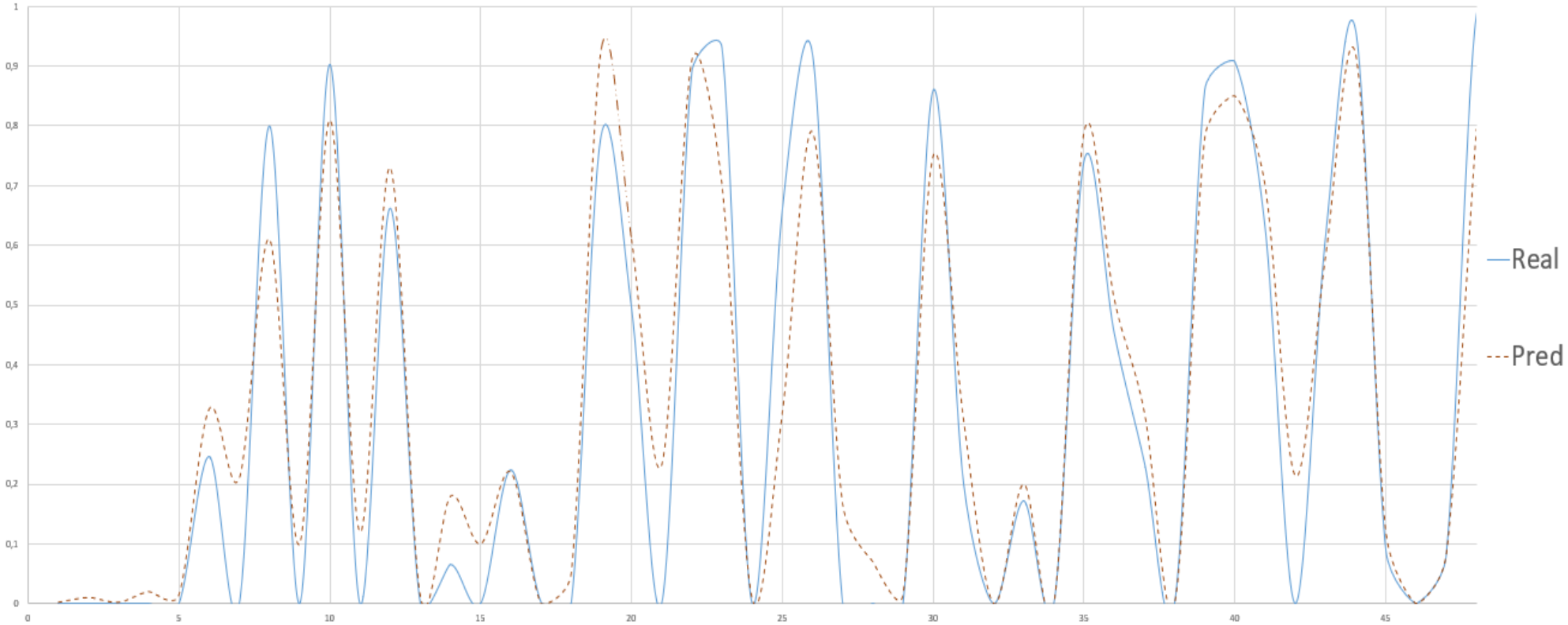

Рисунок 15 – Пример предсказания состояния системы

4.5.2 Учет ошибок предсказания состояния системы и фиксирование наличия атак на систему

Состояния системы, предсказанные нейронной сетью, могут в некоторой степени отличаться от реальных значений, поэтому необходимо учитывать ошибку – возможную разность между показателями.

Для расчета ошибки между предсказанным состоянием системы и реальным выполняется следующий ряд действий:

- вычисление разницы между предсказанным γ_{pred} и реальным γ_{real} значением:

$$err_t = |\gamma_{pred} - \gamma_{real}|;$$

- фиксирование наличия или отсутствия атаки на основе условия превышения значения ошибки реального состояний и предсказанного более чем на фиксированную величину:

$$MAX(err_t) > T,$$

где T – пороговое значение проявления аномального поведения в системе.

Однако возникает вероятность ложных срабатываний за счет кратковременных «выбросов» больших ошибок предсказания в малые промежутки времени, поэтому необходимо учесть усредненную ошибку за некоторый промежуток времени:

$$ERR_i = \langle \sum_{t=i-k}^{i} MAX(err_t) \rangle > T.$$

На Рисунках 16-19 приводятся примеры величины ошибки между предсказанным и реальным состоянием системы при наличии и отсутствии атак.

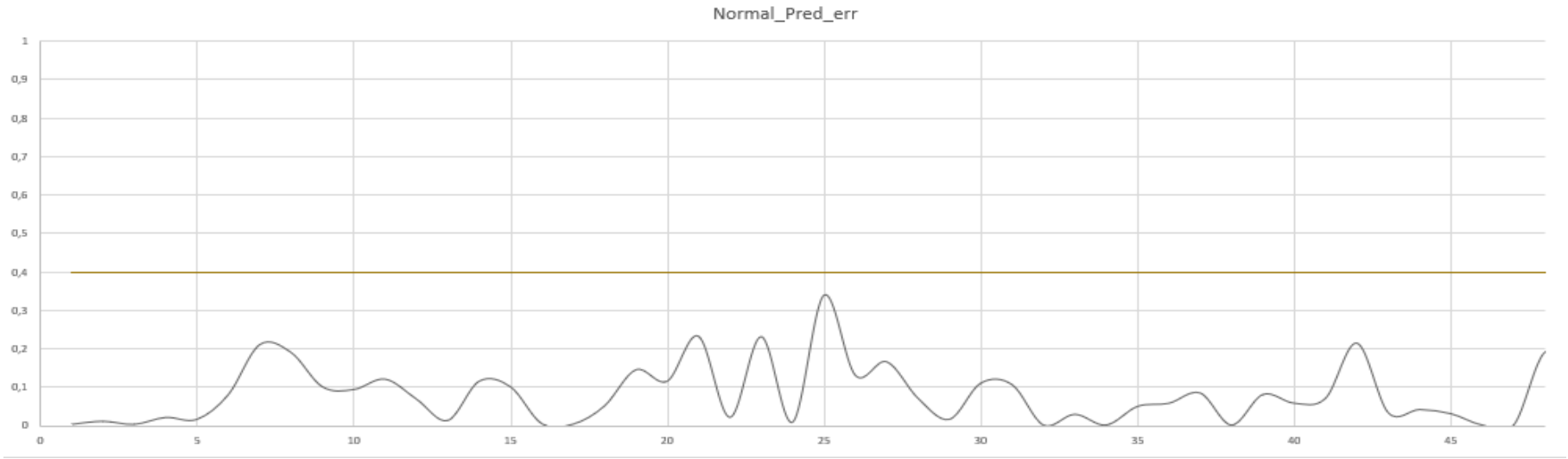

Рисунок 16 – Пример ошибки предсказания состояния системы в случае отсутствия атак

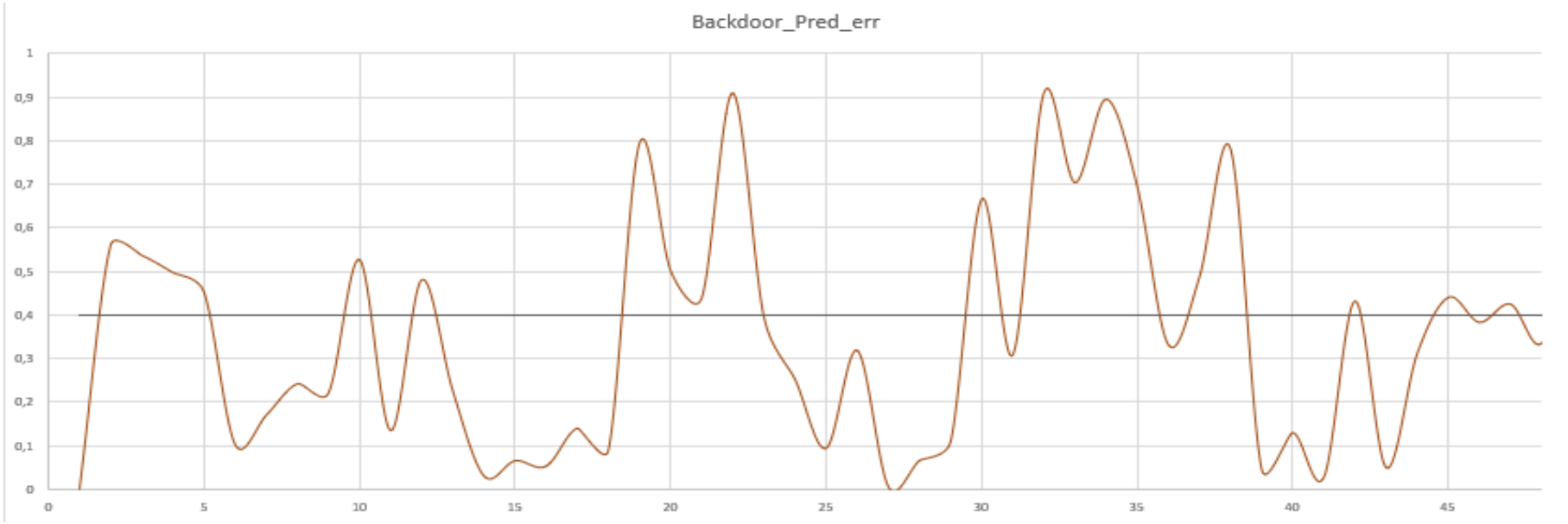

Рисунок 17 – Пример ошибки предсказания состояния системы в случае атаки типа Backdoor

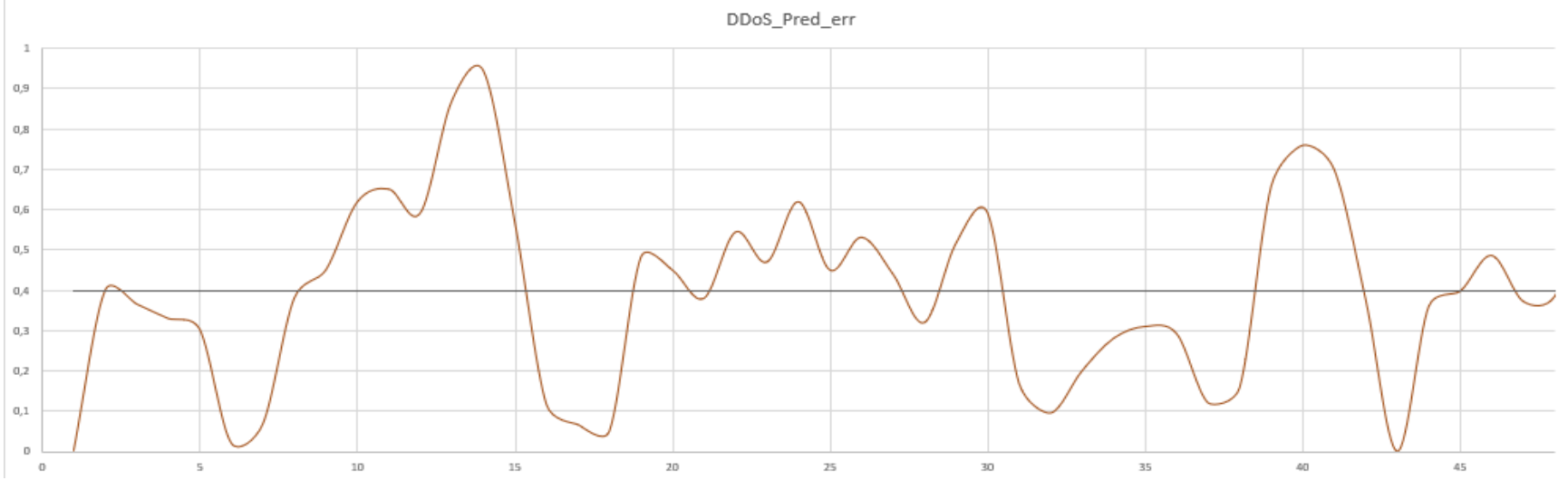

Рисунок 18 – Пример ошибки предсказания состояния системы в случае атаки типа DDoS

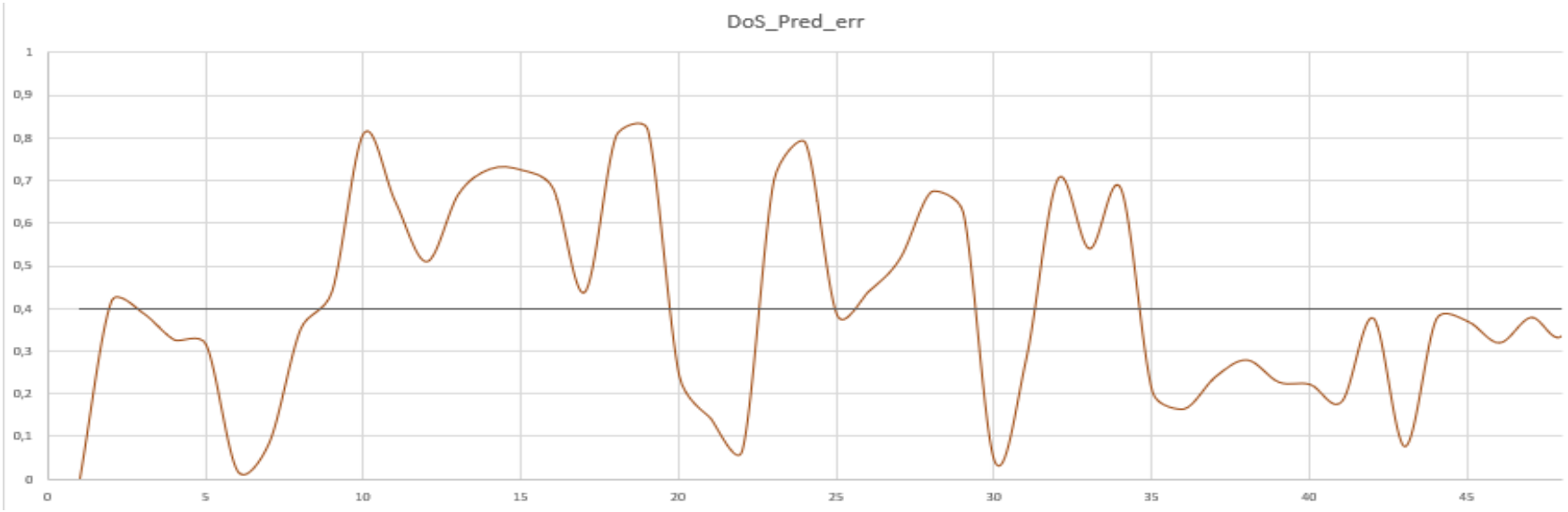

Рисунок 19 – Пример ошибки предсказания состояния системы в случае атаки типа DoS

4.6 Программная реализация разработанного метода

Программная реализация была выполнена средствами языка Python.

Первичная обработка данных выполнялась стандартной библиотекой, позволяющей работать с файлами формата «.csv».

Создание многомерных временных рядов выполнялось с помощью математической библиотеки Pandas.

Нейросеть строилась и обучалась по алгоритму модифицированного гиперкуба с использованием библиотеки NEAT-Python языка Python. Библиотека NEAT-Python использует набор гиперпараметров, которые влияют на выполнение и точность алгоритма NEAT.

В работе использовались нижеописанные гиперпараметры (приведены наиболее важные):

1. Функция активации всех узлов сети является сигмоидальной, а входы узлов агрегируются функцией суммы: activation_default = sigmoid, aggregation_default = sum. Данный выбор обусловлен стремлением выделить слабые сигналы и постараться избежать насыщения и перенасыщения от сильных сигналов.

2. Тип закодированной сети – полносвязная сеть обратного распространения: feed_forward = True, initial_connection = full_direct. В основе данного решения лежит желание оптимизировать скорость схождения нейрогенетического алгоритма топологии нейронной сети. Нежелание использовать топологию сети с обратным распространением в начальной субстрате обосновано отсутствием необходимости восстановления формы и частоты первоначальных данных. Однако в дальнейшем субстрату разрешено эволюционировать до сети с обратными связями, что и наблюдается на практике.

3. В ходе эволюции новые сетевые узлы и связи добавляются и/или удаляются с определенной вероятностью. Очевидно, вероятность добавления и удаления узлов была выставлена с более низким значением, чему у вероятности появления и удаления связей. Данное решение обосновано желанием

оптимизировать топологию сети именно средствами распространения взаимосвязанных данных, появлением и наличием обратных связей, минимизацией создания «мёртвых» узлов.

4. Вероятность добавления/удаления узла - node_add_prob = 0.05, node_delete_prob = 0.05.

5. Вероятность добавления/удаления связи - conn_add_prob = 0.3, conn_delete_prob = 0.3.

6. Все связи включены по умолчанию с очень низкой вероятностью отключения из-за мутации: enabled_mutate_rate = 0.01. Хотя топология нейронной сети и генерируется относительно произвольным образом, не стоит отказываться от операций просеивания и «drop-слоев». Введение мутации отключения произвольных узлов и/или связей позволяет избавиться от паразитных косвенных зависимостей, которые не всегда способны сказаться на предсказании состояния системы положительно. Опять же, в случае удачного возникновения обратных связей не паразитического характера, фитнес-функция не позволит погибнуть популяции со столь удачной мутацией.

Чтобы стимулировать разнообразие видов, зададим сильное влияние избыточных/непересекающихся частей родительских геномов на расстояние между геномами: параметры расстояния между геномами – compatibility_disjoint_coefficient = 1.0. Данное решение позволяет изначально создать максимально возможную псевдослучайную популяцию особей для последующего кроссовера. В противном случае, при создании идентичных особей, приходится ждать дополнительное время возникновения стартовых мутаций, необходимый для успешных операций кроссовера.

Стагнация видов может длиться до 50 поколений, а уникальные виды частично защищены от вымирания: species_fitness_func = min, max_stagnation = 50, species_elitism = 4. Выбор данных величин обусловлен желанием сохранить гладкую тенденцию схождения топологии системы. В случае занижения времени существования стагнируемых популяций, наблюдается появление скачкообразоного развития топологии системы, однако в такие моменты

решение системы сложно назвать стабильным и точным – от раза к разу сходимость системы является весьма произвольной и могут наблюдаться решения на множестве не самых валидных и оптимальных структур. Количество видов, защищенных от вымирания, в противовес длительности стагнации, было занижено для сохранения возможности существования большего количества различных особей в единицу времени.

5 Оценка точности разработанного метода

Тестирование программной реализации созданного метода выполнялось на выборке данных TON_IOT DATASETS [14].

Пороговая величина ошибки определяется эмпирически, и в данном случае (на данном исследуемом датасете) величина T была установлена в значение 0,398. При данном пороговом значении были рассчитаны следующие величины:

1. Accuracy (насколько близко результат измерения к истинному значению) = $(TP + TN) / (P + N)$.
2. Precision (насколько близки измерения одного и того же объекта друг к другу) = $TP / (TP + FP)$.
3. True Positive Rate = $TP / (TP + FN)$.
4. True Negative Rate = $TN / (TN + FP)$.
5. False Positive Rate = $FP / (FP + TN)$.
6. False Negative Rate = $FN / (FN + TP)$.
7. Positive Predictive Value $= 1 - FP / (FP + TP)$.
8. Negative Predictive Value = $TN / (TN + FN)$.
9. F1 Score = $2TP / (2TP + FP + FN)$.
10. Matthews Correlation Coefficient (MCC, Коэффициент корреляции Мэтьюса) = $\frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}$.

В данном случае:

1. TP – количество верных детектирований нормального состояния системы (True Positive).
2. TN – количество верных детектирований атак на систему (True Negative).
3. FP – количество нераспознанных атак (False Positive).
4. FN – количество нормальных состояний системы, распознанных как атаки (False Negative).
5. P – общее количество нормальных состояний КФС (Positive).

6. N – общее количество состояний КФС, включающих в себя атаки (Negative).

Далее приводятся значения для всех рассматриваемых временных промежутков в виде таблиц. Для удобства, значения разбиты по типам атак и по рассматриваемым временным интервалам. После каждого временного промежутка, а также после всех расчетов и выкладок, следуют краткие заключения о точности разработанного метода.

Таблица 3 - Полученные данные на отрезке «без атак». 48 часов

All	1209600
Positive	1209600
Negative	0
True Positive	1083802
True Negative	0
False Positive	0
False Negative	125798

Таблица 4 - Точность метода на отрезке «без атак». 48 часов

Accuracy	0,8960
Precision	1,0000
True Positive Rate	0,8960
True Negative Rate	-
False Positive Rate	-
False Negative Rate	0,1040
Positive Predictive Value	1,0000
Negative Predictive Value	0,0000
F1 Score	0,9451
Matthews Correlation Coefficient	-

На тестовом множестве сложно корректно судить о точности метода. Хотя общая точность (Precision) получилась равно единице, это не гарантирует идеальной работы метода. Данное значение обусловлено отсутствием ложный срабатываний типа FP, так как на данном множестве принципиально

отсутствовали атаки, а само множество являлось эталонным. Близость решения (Accuracy) позволяет сделать вывод, что некие ошибки (здесь – FN) на обучаемом множестве все же присутствовали – переобучение модели не произошло.

Таблица 5 - Полученные данные на отрезке «DoS атаки». 48 часов

All	1209600
Positive	394496
Negative	815104
True Positive	363748
True Negative	760838
False Positive	54266
False Negative	30748

Таблица 6 - Точность метода на отрезке «DoS атаки». 48 часов

Accuracy	0,9297
Precision	0,8702
True Positive Rate	0,9221
True Negative Rate	0,9334
False Positive Rate	0,0666
False Negative Rate	0,0779
Positive Predictive Value	0,8702
Negative Predictive Value	0,9612
F1 Score	0,8954
Matthews Correlation Coefficient	0,8433

Анализируя промежуток DoS, можно сказать, что здесь метод показал себя положительно. Данные слова подтверждаю как высокая близость решений (Accuracy), так и высокая общая точность классификации (Precision). Отдельно стоит отметить следующие величины: False Positive Rate и False Negative Rate – их значения составили менее 0,1 и находятся весьма близко друг другу. Данные показатели свидетельствуют о том, что частота ложных детектирований составляет малую долю от общей, а перекос значений в сторону FP или FN отсутствует.

Таблица 7 - Полученные данные на отрезке «DDoS атаки». 48 часов

All	1209600
Positive	486456
Negative	723144
True Positive	451619
True Negative	667022
False Positive	56122
False Negative	34837

Таблица 8 - Точность метода на отрезке «DDoS атаки». 48 часов

Accuracy	0,9248
Precision	0,8895
True Positive Rate	0,9284
True Negative Rate	0,9224
False Positive Rate	0,0776
False Negative Rate	0,0716
Positive Predictive Value	0,8895
Negative Predictive Value	0,9504
F1 Score	0,9085
Matthews Correlation Coefficient	0,8453

Как и в случае с промежутком, включающим в себя DoS атаки, метод также хорошо отработал на промежутке DDoS атак. Значение общей точности (Precision) немного увеличилось, а в остальном можно сделать выводы, аналогичные случаю с DoS – на заданном промежутке метод прекрасно справился со своей задачей.

Таблица 9 - Полученные данные на отрезке «Backdoor атаки». 48 часов

All	1209600
Positive	781609
Negative	427991
True Positive	658908
True Negative	301315
False Positive	126676
False Negative	122701

Таблица 10 - Точность метода на отрезке «Backdoor атаки». 48 часов

Accuracy	0,7938
Precision	0,8387
True Positive Rate	0,8430
True Negative Rate	0,7040
False Positive Rate	0,2960
False Negative Rate	0,1570
Positive Predictive Value	0,8387
Negative Predictive Value	0,7106
F1 Score	0,8409
Matthews Correlation Coefficient	0,5482

На данном промежутке метод показал самую низкую точность (Precision) и близость решений (Accuracy). Как не сложно заметить из False Positive Rate и False Negative Rate, практически треть атак была неверно классифицирована системной как нормальное состояние КФС. Данное поведение можно частично объяснить тем, что в исследуемом датасете под Backdoor атаками понималось повторное дублирование пакета отправителю в определенные промежутки. Так как мощность потока данных пакетов редко превышала величину нормально отправляемых повторных пакетов в случае реальной потери первичных, метод не в полной мере справился конкретно с этой атакой. Рекомендуется использовать либо дополнительные критерии отбора и детектирования на данной реализации атак, либо ввести в таком случае событийный обработчик повторной пересылки ранее полученных пакетов.

Таблица 11 - Полученные данные на отрезке «Все атаки». 144 часа

All	3628800
Positive	1662561
Negative	1966239
True Positive	1474275
True Negative	1729175
False Positive	237064
False Negative	188286

Таблица 12 - Точность метода на отрезке «Все атаки». 144 часа

Accuracy	0,8828
Precision	0,8615
True Positive Rate	0,8867
True Negative Rate	0,8794
False Positive Rate	0,1206
False Negative Rate	0,1133
Positive Predictive Value	0,8615
Negative Predictive Value	0,9018
F1 Score	0,8739
Matthews Correlation Coefficient	0,7647

Отдельно, помимо временных промежутков, включающих в себя дискретные атаки, стоит рассмотреть точность метода на всех промежутках, включающих в себя атаках.

Не сложно заметить, что общая точность (Precision) и близость решения (Accuracy) слезка уменьшились относительно тех же показателей, но в случаях DoS и DDoS атак. Данное изменение связано с более низкой точностью метода на множестве Backdoor атак. Возможные причины данного поведения – достаточно редкое дублирование отправляемых пакетов во время атаки, которое легко затерять на фоне реальной потери пакетов и легитимного дублирования.

Таблица 13 - Полученные данные на отрезке «За все время». 192 часа

All	4838400
Positive	2872161
Negative	1966239
True Positive	2558077
True Negative	1729175
False Positive	237064
False Negative	314084

Таблица 14 - Точность метода на отрезке «За все время». 192 часа

Accuracy	0,8861
Precision	0,9152
True Positive Rate	0,8906
True Negative Rate	0,8794
False Positive Rate	0,1206
False Negative Rate	0,1094
Positive Predictive Value	0,9152
Negative Predictive Value	0,8463
F1 Score	0,9027
Matthews Correlation Coefficient	0,7658

Подводя итоги по точности метода, также стоит рассмотреть данные показатели за все исследуемые временные промежутки. Агрегируя данные по всем 4 промежуткам – DoS, DDoS, Backdoor атаки и промежутку для обучения (отсутствие атак), мы получим весьма высокие показатели общей точности (Precision) и близости решений (Accuracy). Дополнительно стоит отметить, что на общем промежутке значения True Positive Rate и True Negative Rate претерпели незначительные падения на единицы процентов, что, опять же, свидетельствует о высоком уровне точности и надежности метода как на валидационном множестве, на множестве дискретных атак, так и на совокупности этих множеств.

Для получения более четкого представления о точности метода был выполнен ROC анализ. На Рисунке 20 приводится аппроксимированная ROC кривая на временном промежутке «За все время» за 192 часа.

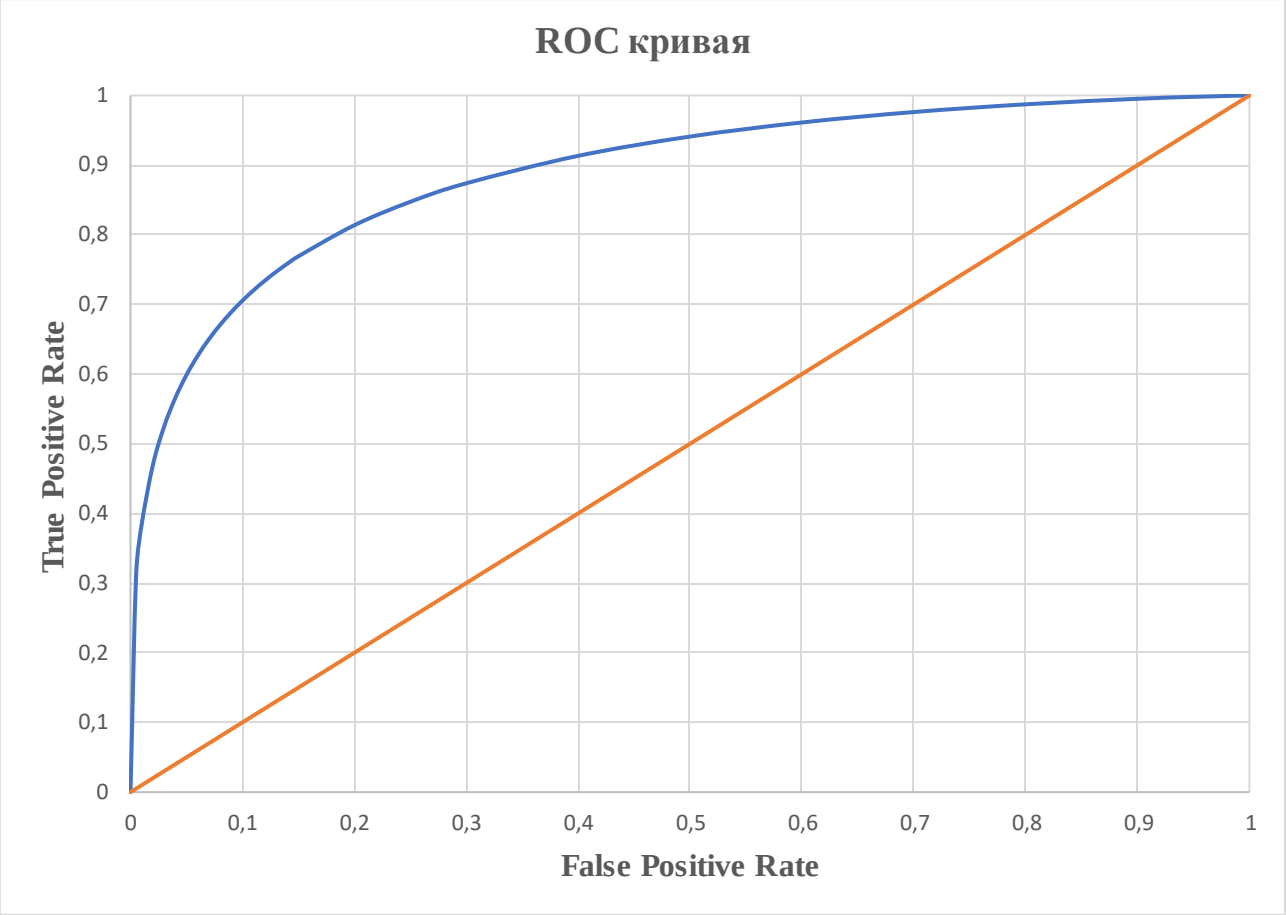

Рисунок 20 – Аппроксимированная ROC кривая для временного промежутка «За все время»

Показатель AUROC (площадь под графиком кривой) можно трактовать как эквивалентность вероятности того, что бинарный классификатор при выполнении оценки присвоит больший вес случайно выбранной положительной характеристике или показателю, нежели случайно выбранному отрицательному. В идеальных условиях данный показатель стремится к 1, а в случае равновероятностного «угадывания» на множестве – к 0,5 (данная прямая представлена на Рисунке 20 оранжевым цветом). Для данного случая показатель AUROC составил 0,89, что подчеркивает достаточную точность метода и удачно выбранную пороговую величину ошибки.

ЗАКЛЮЧЕНИЕ

Результатом настоящей диссертационной работы является создание, реализация и экспериментальные исследования реализованного метода детектирования сетевых атак, проводимых на КФС. Метод включает в себя использование нейроэволюционного алгоритма семейства NEAT: модифицированный NEAT-гиперкуб.

После проведенной модификации алгоритм позволяет практически полностью сконфигурировать целевую нейронную сеть без участия пользователя по заданным параметрам, в том числе дополнительно создавая промежуточные слои сети, ранее недоступные в первичной версии алгоритма.

Выявление сетевых атак, проводимых на КФС, осуществлялось в несколько этапов:

1. Первичная обработка данных и представление их в виде многомерных временных рядов.
2. Конфигурирование нейронной сети генетической составляющей NEAT-гиперкуба.
3. Обучение сконфигурированной нейронной сети на тестовом множестве.
4. Предсказание будущего состояния системы на основе текущих данных.
5. Расчет ошибки между предсказанным и реальным состояниями системы.
6. Сравнение полученной ошибки с минимальным пороговым значением T.

Тестирование выполнялось на наборе данных TON_IOT DATASETS [14].

Полученные общая точность (Precision; 0,9152) и близость решений (Accuracy; 0,8861), а также величины False Positive Rate (0,1206) и False Negative Rate (0,1094) свидетельствуют об отсутствии переобучения модели и высокой надежности данного метода.

Дальнейшим направлением развития темы является создание модели потока данных киберфизических систем на основе гиперкуба с возможностью самовосстановления по адаптивной графой структуре.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лаврова Д. С. Подход к разработке SIEM-системы для интернета вещей / Д. С. Лаврова // Проблемы информационной безопасности. Компьютерные системы. – 2016. – № 2. – С. 51–59.

2. Калинин М. О., Лаврова Д. С., Ярмак А. В. Обнаружение угроз в киберфизических системах на основе методов глубокого обучения с использованием многомерных временных рядов / М. О. Калинин, Д. С. Лаврова, А. В. Ярмак // Проблемы информационной безопасности. Компьютерные системы. – 2018. – № 2. – С. 111–117.

3. Зегжда П. Д. Обнаружение аномалий в сетевом трафике с использованием дискретного вейвлет-преобразования и метода разладки / П. Д. Зегжда и др. // Проблемы информационной безопасности. Компьютерные системы. – 2018. – № 4. – С. 14–21.

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

5. Lavrova D. S. An approach to developing the SIEM system for the Internet of Things. / D. S. Lavrova // Automatic Control and Computer Sciences. – 2016. – Vol. 50. – № 8. – pp. 673–681.

6. Фатин А. Д., Павленко Е. Ю. Анализ моделей представления киберфизических систем в задачах обеспечения информационной безопасности / А. Д. Фатин, Е. Ю. Павленко // Проблемы информационной безопасности. Компьютерные системы. – 2020. – № 2. – С. 109-121.

7. Tulone D., Madden S. PAQ: Time series forecasting for approximate query answering in sensor networks / D. Tulone, S. Madden // Wireless Sensor Networks: Third European Workshop, EWSN 2006, Zurich, Switzerland, 2006. – pp. 21-37.

8. Assumption-Free Anomaly Detection in Time Series / L. Wei, N. Kumar, V. Lolla, E. Keogh, Sю Lonardi, Ch. A. Ratanamahatana //SSDBM: Proceedings of the

17th international conference on Scientific and statistical database management. – 2005. – Vol. 5. – pp. 237-242.

9. Pincombe B. Anomaly Detection in Time Series of Graphs using ARMA Processes / B. Pincombe // Asor Bulletin. – 2005. – Vol. 24. – №. 4. – pp. 2-10.

10. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding / K. Hundman, V. Constantinou, Ch. Laporte, I. Colwell, T. Soderstrom //KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – 2018. – pp. 387–395

11. Filonov P., Lavrentyev A., Vorontsov A. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model / P. Filonov, A. Lavrentyev, A. Vorontsov // NIPS Time Series Workshop, 2016.

12. Nanduri A., Sherry L. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN) / A. Nanduri, L. Sherry //Integrated Communications Navigation and Surveillance (ICNS), 2016. – IEEE, 2016. – pp. 5C2-1-5C2-8.

13. Grouped Convolutional Neural Networks for Multivariate Time Series /S. Yi, J. Ju, M.-K. Yoon, J. Choi // URL: https://arxiv.org/pdf/1703.09938.pdf (дата обращения: 12.01.2021).

14. TON_IOT DATASETS. – URL: https://ieee-dataport.org/documents/toniot-datasets (дата обращения: 12.01.2021).

15. Stouffer, K. , Falco, J., Scarfone, K. Guide to Industrial Control Systems (ICS) Security – Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and other control system configurations such as Programmable Logic Controllers (PLC), Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, URL: https://doi.org/10.6028/NIST.SP.800-82 (дата обращения: 13.02.2021).

16. Лаврова Д. С., Хушкеев А. А. Обнаружение нарушений информационной безопасности в АСУ ТП на основе прогнозирования многомерных временных рядов, сформированных из значений параметров работы конечных устройств системы / Д. С. Лаврова, А. А. Хушкеев //

Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 1. – С. 18–30

17. Лаврова Д. С., Ярмак А. В. Прогнозирование атак на подсистему управления промышленных объектов с использованием глубокого обучения / Д. С. Лаврова, А. В. Ярмак // Сборник трудов XIII Всероссийского совещания по проблемам управления ВСПУ-2019, Москва, Россия. – 2019. – С. 2581–2586.

18. Лаврова Д. С. Прогнозирование состояния компонентов интеллектуальных сетей энергоснабжения smart grid для раннего обнаружения кибератак / Д. С. Лаврова // Проблемы информационной безопасности. Компьютерные системы. – 2019. – № 4. – С. 101–104.

19. Kalman R. E. A new approach to linear filtering and prediction problems // Journal of basic Engineering. – 1960. – Vol. 82. – № 1. – P. 35–45.

20. Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system // Control engineering practice. – 2004. – Vol. 12. – № 12. – pp. 1531–1539.

21. Лаврова Д. С., Алексеев И. В., Штыркина А. А. Анализ безопасности на основе контроля зависимостей параметров сетевого трафика с использованием дискретного вейвлет-преобразования / Д. С. Лаврова, И. В. Алексеев, А. А. Штыркина // Проблемы информационной безопасности. Компьютерные системы. – 2018. – № 2. – С. 9–15.

22. Sheluhin O. I., Atayero A. A. Detection of DoS and DDoS Attacks in Information Communication Networks with Discrete Wavelet Analysis / O. I. Sheluhin, A. A. Atayero // International Journal of Computer Science and Information Security. – 2012. – Т. 10. – Vol. 1. – pp. 53.

23. Зегжда П. Д., Лаврова Д. С., Штыркина А. А. Мультифрактальный анализ трафика магистральных сетей Интернет для обнаружения атак отказа в обслуживании / П. Д. Зегжда, Д. С. Лаврова, А. А. Штыркина // Проблемы информационной безопасности. Компьютерные системы. – 2018. – № 2. – С. 48–58.

24. Божокин С. В., Паршин Д. А. Фракталы и мультифракталы /

С. В. Божокин, Д. А. Паршин. – Ижевск: НИЦ, 2001. – С. 67–70.

25. Multifractal analysis of soil surface roughness / Moreno G. et al. // Vadose Zone Journal. – 2007. – № 7(2). – pp. 512–520.

26. Sheluhin O., Atayero A., Garmashev A. Detection of Teletraffic Anomalies Using Multifractal Analysis / O. Sheluhin, A. Atayero, A. Garmashev // International Journal of Advancements in Computing Technology. – 2001. – Vol. 3. – № 4. – pp. 174-182.

27. Кириченко Л. О. Сравнительный мультифрактальный анализ временных рядов методами детрендированного флуктуационного анализа и максимумов модулей вейвлет-преобразования / Л. О. Кириченко // Автоматизированные системы управления и приборы автоматики. – 2011. – № 157. – С.66–77.

28. Multifractal detrended fluctuation analysis of nonstationary time series / Kantelhardt J.W. et al. // Physica A. – 2002. – № 316. – pp. 87–114.

29. Олемской А. И., Борисюк В. Н., Шуда И. А. Мультифрактальный анализ временных рядов / А. И. Олемской, В. Н. Борисюк, И. А. Шуда // Вісник СумДУ Серія «Фізика, математика, механіка». – 2008. – №2. – С. 70-81.

30. Лаврова Д. С., Зегжда Д. П., Зайцева Е. А Моделирование сетевой инфраструктуры сложных объектов для решения задачи противодействия кибератакам / Д. С. Лаврова, Д. П. Зегжда, Е. А Зайцева // Вопросы кибербезопасности. – 2019. – № 2. – С. 13–20.

31. Павленко Е. Ю., Ярмак А. В., Москвин Д. А. Иерархический подход к анализу нарушений безопасности в информационных системах / Е. Ю. Павленко, А. В. Ярмак, Д. А. Москвин // Проблемы информационной безопасности. Компьютерные системы. – 2017. – № 1. – С. 92–99.

32. Павленко Е. Ю., Ярмак А. В., Москвин Д. А. Контроль безопасности информационных систем на основе анализа графа событий, полученных в результате мониторинга / Е. Ю. Павленко, А. В. Ярмак, Д. А. Москвин // Проблемы информационной безопасности. Компьютерные системы. – 2017. – № 2. – С. 31–38.

33. Поздняк И. С., Буранова М. А Исследование сетевого трафика на степень самоподобия: методические указания / И. С. Поздняк, М. А Буранова. – Самара. – 2013. – 17 с.

34. Бутаков В., Граковский А. Оценка уровня стохастичности временных рядов произвольного происхождения при помощи показателя Херста / В. Бутаков, А. Граковский // Computer modeling and new technologies. – 2005. – Vol. 9. – № 2. – P. 27-32.

35. Adams R. P., MacKay D. J. C. Bayesian online changepoint detection / R. P. Adams, D. J. C. MacKay //arXiv preprint arXiv:0710.3742. – 2007.

36. Anderson K. C. A novel approach to Bayesian online changepoint detection. / K. C. Anderson // University of Colorado, Boulder. – 2008. – 30 p.

37. Айвазян С. А. Байесовский подход в эконометрическом анализе / С. А. Айвазян // Прикладная эконометрика. – 2008. – №. 1(9). С. 93-130.

38. Vyshemirsky V., Macaulay V. Bayesian changepoint detection in solar activity data. / V. Vyshemirsky, V. Macaulay, Glasgow – 2014. – 52 p.

39. Generalize linear models. URL: http://data.princeton.edu/wws509/notes/c7s1.html (дата обращения 03.12.2018).

40. Kim S. S., Reddy A. L. N., Vannucci M. Detecting traffic anomalies using discrete wavelet transform / S. S. Kim, A. L. N. Reddy, M. Vannucci // Proceedings of the International Conference on Information Networking. – 2004. – pp. 951-961

41. Salagean M., Firoiu I. Anomaly detection of network traffic based on Analytical Discrete Wavelet Transform / M. Salagean, I. Firoiu // 8th International Conference on Communications, IEEE. – 2010.

42. Тишина Н.А., Дворовой И.Г., Соловьев Н.А. Обнаружение вторжений на основе вейвлет-анализа сетевого трафика / Н. А. Тишина, И. Г. Дворовой, Н. А. Соловьев // Вестник Уфимского государственного авиационного технического университета. – 2010. – Т. 14. – № 5 (40). – С. 188-194.

43. Wavelet methods for the detection of anomalies and their application to network traffic analysis / D. W. Kwon et al. // Quality and Reliability Engineering International. – 2006. – Т. 22. – №. 8. – pp. 953-969.

44. Шелухин О. И., Гармашев А. В. Обнаружение аномальных выбросов телекоммуникационного трафика методами дискретного вейвлет-анализа / О. И. Шелухин, А. В. Гармашев // Электромагнитные волны и электронные системы. – 2012. – Т. 17. – №. 2. – С. 15-26.

45. Шелухин О. И., Филинова А. С. Сравнительный анализ алгоритмов обнаружения аномалий трафика методами дискретного вейвлет-анализа / О. И. Шелухин, А. С. Филинова // T-CommТелекоммуникации и Транспорт. – 2014. – Т. 8. – №. 9. – С. 89-97.

46. Ишханян М. В. Основы математического прогнозирования социальноэкономических процессов: учебное пособие/ М. В. Ишханян, Москва: ФГБОУ ВО «Московский государственный университет путей сообщения императора Николая II». – 2016. – 121 с.

47. Лаврова Д. С. Методология предотвращения компьютерных атак на промышленные системы на основе адаптивного прогнозирования и саморегуляции: дис…канд. техн. наук : 05.13.19 / Д. С. Лаврова ; СПбПУ. – СПб., 2019. – 303 с.

48. Омельяненко Я. Эволюционные нейросети на языке Python / Я. Омельяненко, пер. с англ. В. С. Яценкова. – М.: ДМК Пресс, 2020. – 310 с.