Казахский национальный исследовательский технический университет имени К.И. Сатпаева

УДК 004.056.5:004.8

На правах рукописи

БАПИЕВ ИДЕЯТ МЭЛСОВИЧ

Нейросетевые модели и методы противодействия атакам на сетевые ресурсы информационных систем

6D070300 – Информационные системы

Диссертация на соискание ученой степени
доктора философии (PhD)

Научные консультанты:
Айтчанов Б. Х., доктор техн. наук, профессор

Корченко А. Г., доктор техн. наук, профессор

Республика Казахстан
Алматы, 2018

СОДЕРЖАНИЕ

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей диссертации использованы ссылки на следующие стандарты:

– «Инструкция по оформлению диссертации и автореферата», ВАК МОН Республики Казахстан от 28 сентября 2004г. №377-3ж.

– ГОСТ 7.32-2001 – Отчет о научно-исследовательской работе. Структура и правила оформления.

– ГОСТ 7.1-2003. Библиографическая запись. Библиографическое описание. Общие требования и правила составления.

– СТ РК 34.007-2002. Информационная технология. Телекоммуникационные сети. Основные термины и определения.

– СТ РК 34.020-2006. Защита информации. Технические средства защиты информации. Имитаторы излучения. Общие технические требования.

– СТ РК ГОСТ Р 51275-2006 (ГОСТ Р 51275-99, IDT). Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования.

– СТ РК ГОСТ Р ИСО/МЭК 15408-1-2006 (ГОСТ Р ИСО/МЭК 15408-1-2002, IDT). Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий.

– СТ РК ИСО/МЭК 15443-1-2007. Технологии информационные. Методы обеспечения защиты. Структура обеспечения безопасности информационных технологий.

– СТ РК 1699-2007. Система контроля и управления доступом. Общие технические требования.

– СТ РК ИСО/МЭК 10118-1-2006(ИСО/МЭК 10118-1-2000, (Е), IDT). Информационная технология. Методы защиты информации Хэш-функции.

– СТ РК 34.022-2006. Защита информации. Требования к проектированию, установке, наладке, эксплуатации и обеспечению безопасности информационных систем.

– СТ РК ИСО/МЭК 17799-2006 (ИСО/МЭК 17799-2005, IDT. Информационная технология. Методы обеспечения защиты, свод правил по управлению защитой информации.

– СТ РК 1698-2007. Защита информации. Защита информаций от технических разведок и от утечки по техническим каналам на объекте средств вычислительной техники. Методы защиты.

ОПРЕДЕЛЕНИЯ

В настоящей диссертации применяют следующие термины с соответствующими определениями:

Эффективность – множество атрибутов, которые определяют взаимосвязь уровней исполнения информационной программной системы, использование ресурсов (средства, аппаратура, материалы и др.) и услуг, которые выполняются штатным обслуживающим персоналом.

Оперативность – атрибут, который указывает на время отклика, обработки и выполнении функций.

Ресурсоемкость – атрибут, который определяет количество использованных ресурсов и продолжительность такого использования при выполнении функций аппаратно-программной системы.

Кибератака – реализация в кибернетическом пространстве угроз безопасности компонентов информации (а именно конфиденциальности, целостности и доступности) с учетом их уязвимостей.

Кибернетическое пространство – виртуальное пространство, полученное в результате взаимодействия пользователей, программного и аппаратного обеспечения, сетевых технологий для поддержки и управления процессами преобразования информации с целью обеспечения информационных потребностей общества.

Ресурс информационной системы – отдельный программный или аппаратный компонент, обеспечивающий функционирование информационной системы.

Сетевой ресурс информационной системы – ресурс, который используется для обеспечения сетевых функций.

Сигнатура функциональных параметров – зарегистрированное в определенный интервал времени множество функциональных параметров ресурса информационной системы.

Параметр безопасности – параметр, с помощью которого отображается состояние безопасности объекта защиты информационной системы.

Портрет (сигнатура) кибератаки – сигнатура функциональных параметров при реализации определенного вида кибератаки.

Нейронная сеть – сеть, состоящая из искусственных нейронов, объединенных между собою синаптическими (взвешенными) связями.

Нейросетевая модель – модель нейронной сети, которая характеризуется методом обучения, способом распространения сигнала, структурой связей и типом искусственного нейрона.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

АНС – ассоциативная нейронная сеть;
АПО – аппаратно-программное обеспечение;
БД – база данных;
ВС – входной слой;
ГНС – глубокая нейронная сеть;
ДСП – двухслойный персептрон;
ИС – информационная система;
КСС – компьютерные системы и сети;
МСП – многослойный персептрон;
НС – нейронная сеть;
НСД – нейронная сеть Джордана;
НСМ – нейросетевая модель;
НСР – нейросетевое средство;
НСС – нейросетевая система;
РБФ – сеть радиальной базисной функции;
РИС – ресурс информационной системы;
СВ – выходной слой;
СВР – сеть встречного распространения;
СС – слой суммирования;
СЗИ – система защиты информации;
СНС – сверточная нейронная сеть;
СПА – система противодействия атакам;
СО – слой образов;
ССН – скрытый слой нейронов;
СФП – сигнатура функциональных параметров;
ТК – карта Кохонена;
PNN – вероятностная нейронная сеть.

ВВЕДЕНИЕ

Оценка современного состояния решаемой научной или научнотехнологической проблемы (задачи).

В современных условиях системы противодействия кибератакам являются одними из основных средств защиты информации сетевых ресурсов информационных систем. Хотя используются такие системы уже не одно десятилетие, их разработкой занимается много высококвалифицированных специалистов, а созданию соответствующей научно-методической базы посвящено большое количество работ, однако практический опыт указывают на наличие в системах противодействия сетевым кибератакам ряда существенных недостатков. Основным из них является недостаточная точность распознавания всей номенклатуры сетевых кибератак, что подтверждается известными случаями успешного взлома систем защиты информации в ряде мировых стран [92]. Кроме этого, внедрение известных средств противодействия сетевым кибератакам в системы защиты информации отечественных информационных систем вызывает необходимость их сложной адаптации к вариативности условий использования. Также недостатками известных средств противодействия кибератакам на сетевые ресурсы информационных систем является высокая стоимость и отсутствие подробной научно-технической документации.

Исследования ученых [2, 15, 19, 22, 40, 50, 67, 70] и др. указывают на то, что перспективным путем повышения эффективности средств распознавания сетевых кибератак является использование в них аппарата искусственных нейронных сетей. Это объясняется доказанной эффективностью применения НС для решения подобных задач ведущими разработчиками средств защиты информации (компании Cisco, Symantec) и доказанной адаптивностью нейросетевых средств (НСР) к разнообразным условиям применения.

Методологическую и теоретическую основу для эффективного внедрения НСР в СПА составляют теоретические разработки и опыт создания систем защиты информации как отечественных, так и зарубежных ученых [3, 6, 33, 34, 69, 74, 77] и др.

Основание и исходные данные для разработки темы.

Согласно Концепции кибербезопасности («Киберщит Казахстана») [1], проявления компьютерной преступности и компьютерного терроризма относятся к основным угрозам, мешающим устойчивому и безопасному функционированию национальных информационных систем.

Исследования Международного союза электросвязи (ITU) показали, что «Глобальный индекс кибербезопасности (GCI)» Казахстана в 2017 г. составил 0,352, что соответствует 83 месту из 193 исследованных стран (Последние новости в Казахстане: [сайт], URL: https://i-news.kz/news/2017/07/13/8556741-azahstan_v_indekse_kiberbezopasnosti_gc.html).

В соответствии с требованиями государственной программы «Цифровой Казахстан» на 2017-2020 года, новые реалии диктуют необходимость

постоянного увеличения скоростных параметров сетей и мощности объектов информационно-коммуникационной инфраструктуры, предназначенных для обеспечения функционирования технологической среды в целях формирования электронных информационных ресурсов и предоставления доступа к ним.

Вместе с тем, развитие телекоммуникационных сетей и технологий передачи данных требует принятия мер по обеспечению конфиденциальности и целостности информации.

Обоснование необходимости проведения научно-исследовательской работы.

Необходимость выполнения настоящей научно-исследовательской работы продиктована намеченным мероприятиям Концепции кибербезопасности ("Киберщит Казахстана") которая разработана в соответствии с Посланием Президента Республики Казахстан «Третья модернизация Казахстана: Глобальная конкурентоспособность». В частности, в данной Концепции в числе ключевых проблем в сфере защиты электронных информационных ресурсов указано, что «… Казахстан как страна, пока, в значительной мере импортирует (заимствует) не только IT-технологии, но и готовые программные продукты, включая продукты обеспечения информационной безопасности в сфере информатизации и связи, что указывает на недостаточность принимаемых усилий и мер по их рациональному замещению с опорой на собственные силы …». В перечне задач Концепции значится: «Формирование необходимых условий для повышения потенциала отечественной отрасли ИКТ по созданию программных продуктов и систем кибербезопасности, направленных на блокирование и подавление вредоносного программно-технического воздействия и защищенного телекоммуникационного оборудования» [1]. В связи с этим разработка нейросетевых моделей и методов противодействия атакам на сетевые ресурсы информационных систем и получаемая в результате такой разработки система противодействия кибератакам может позволить преодоление проблемы не высокой востребованности отечественных разработок.

Сведения о планируемом научно-техническом уровне разработки.

В ходе исследований планировалась разработка нейросетевой системы противодействия сетевым кибератакам. Уровень разработки соответствует современным требованиям т.к. был использован целый комплекс достаточно точных классических и современных методов исследования (методы теории цифровой обработки сигналов, нейронных сетей, экспертного анализа, математической статистики и оптимизации и др.). Уровень исследований подтвержден апробацией и научными публикациями по теме исследований.

Сведения о метрологическом обеспечении диссертации.

Все результаты, полученные в работе, либо основываются на известных теоретических сведениях, либо доказаны и подкреплены применением современных научных методов анализа и исследований.

Экспериментальная установка представляет собой аппаратно-программный комплекс, предназначенный для проведения экспериментальных

исследований разработанных моделей и методов, а также созданной на их основе нейросетевой системы распознавания сетевых кибератак.

Вычислительные возможности и конфигурация аппаратного обеспечения экспериментальной установки определялись с позиций обеспечения минимально допустимых требований к универсальным СРК типа Snort, приспособленных для разворачивания на операционных системах семейств Windows и Linux [68, 78]. Также учтено, что сетевые возможности аппаратного обеспечения должны обеспечивать потенциальную возможность перехвата сетевого трафика, соответствующего стеку протоколов TCP/IP. Поэтому в базовой конфигурации использован универсальный персональный компьютер на основе процессора Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz с оперативной памятью объемом 3,7 ГБ, жестким диском объема 1 ТБ и сетевой картой Attansic L1 Gigabit Ethernet 10/100/1000Base-T. При этом следует отметить, что для функционирования программного обеспечения экспериментальной установки достаточно 20 ГБ постоянной памяти.

Актуальность исследований.

Информационные технологии стали неотъемлемой частью жизни человека, что требует нового подхода к защите информации. В такой постановке является актуальной задача разработки эффективных моделей, методов и систем противодействия сетевым кибератакам, которые бы были адаптированы к отечественным условиям применения. Перспективным путем повышения эффективности систем противодействия атакам на сетевые ресурсы информационных систем является использование в них аппарата искусственных нейронных сетей.

Таким образом, задача разработки эффективных нейросетевых моделей, методов и средств противодействия кибератакам на сетевые ресурсы информационных систем обуславливает актуальность научных исследований и разработок, которым посвящена диссертационная работа.

Научная новизна полученных результатов состоит в том, что теоретические и практические исследования дозволили разработать и научно обосновать принципы, модели и методы нейросетевого противодействия кибератакам на сетевые ресурсы информационных систем.

Впервые:

– разработан метод создания обучающей выборки для нейросетевого противодействия сетевым кибератакам, который за счет определения параметров допустимых видов выборки и учета в выходном сигнале близости эталонов видов кибератак, позволяет определить круг допустимых видов нейросетевых моделей и обеспечить уменьшение количества учебных итераций;

– разработан метод нейросетевого противодействия кибератакам на сетевые ресурсы информационных систем, который за счет использования разработанных нейросетевых моделей и разработанного метода создания обучающей выборки, позволяет расширить функциональные возможности и обеспечить достаточную точность распознавания.

Получили дальнейшее развитие:

– методологическая база нейросетевого противодействия кибератакам на сетевые ресурсы информационных систем, которая за счет учета условий создания таких средств, обеспечила возможность создания эффективных нейросетевых моделей и методов противодействия;
– нейросетевые модели противодействия, которые за счет возможности обучения с помощью экспертных данных и использования комбинированной обучающей выборки, позволяют оперативно реагировать на новые типы сетевых кибератак.

Связь работы с государственными программами и научно-исследовательскими работами. Диссертационная работа имеет связь с научно-исследовательскими работами, выполняемыми в рамках Концепции кибербезопасности "Киберщит Казахстана". Данная концепция разработана в соответствии с Посланием Президента Республики Казахстан "Третья модернизация Казахстана: Глобальная конкурентоспособность" с учетом подходов Стратегии "Казахстан-2050" по вхождению Казахстана в число 30-ти самых развитых государств мира. А также результаты данной работы соответствуют целями и задачами Государственной программы «Информационный Казахстан - 2020».

Цель работы. Цель работы состоит в разработке эффективных нейросетевых моделей, методов и средств противодействия кибератакам, адаптированных к условиям эксплуатации и способных оперативно реагировать на новые виды сетевых кибератак.

В соответствии с поставленной целью определены следующие **основные задачи исследования**:
– провести анализ возможностей нейросетевых средств противодействия кибератакам на сетевые ресурсы информационных систем;
– развить методологическую базу нейросетевого противодействия кибератакам на сетевые ресурсы информационных систем;
– разработать нейросетевые модели и методы противодействия сетевым кибератакам;
– разработать нейросетевую систему противодействия сетевым кибератакам и провести экспериментальные исследования, направленные на верификацию предложенных решений.

Объект исследования – процессы противодействия кибератакам на сетевые ресурсы информационных систем.

Предмет исследования – нейросетевые модели, методы и средства противодействия кибератакам на сетевые ресурсы информационных систем.

Методы исследования – методы теории цифровой обработки сигналов, нейронных сетей, экспертного анализа, математической статистики и оптимизации.

Методологическая база нейросетевого распознавания кибератак на сетевые ресурсы информационных систем базируется на разработаннных в диссертационном исследовании новых нейросетевых моделях и методах, которые обеспечили возможность создания эффективной нейросетевой

системы противодействия атакам на сетевые ресурсы информационных систем.

Положения, выносимые на защиту.

На защиту диссертационной работы выносятся следующие положения:

– результаты разработки эффективных моделей и методов создания обучающей выборки для нейросетевой системы противодействия атакам на сетевые ресурсы информационных систем, которые за счет определения параметров допустимых видов выборки и учета в выходном сигнале близости эталонов видов кибератак, позволяют определить круг допустимых видов нейросетевых моделей и обеспечить уменьшение количества учебных итераций;

– результат разработки нового метода нейросетевого противодействия кибератакам на сетевые ресурсы информационных систем, который за счет использования разработанных нейросетевых моделей и разработанного метода создания обучающей выборки, позволяет расширить функциональные возможности и обеспечить достаточную точность распознавания;

– результаты экспериментальных исследований, направленных на верификацию предложенных решений.

1 АНАЛИЗ ВОЗМОЖНОСТЕЙ НЕЙРОСЕТЕВЫХ СРЕДСТВ ПРОТИВОДЕЙСТВИЯ КИБЕРАТАКАМ НА СЕТЕВЫЕ РЕСУРСЫ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1 Научно-практическая задача противодействия кибератакам на сетевые ресурсы информационных систем

На первом, наиболее сложном, этапе задача противодействия кибератакам сводится к идентификации событий, связанных с реализацией в кибернетическом пространстве угроз безопасности ресурсам информационных систем с учетом их уязвимостей. Другими словами, данная идентификация это обнаружение и классификация любых видов несанкционированных действий по отношению к этим ресурсам. В данной работе акцент ставится на распознавании кибератак, реализуемых посредством использования сетевого трафика, поскольку результаты [13, 20, 31, 94, 96] указывают на то, что в настоящее время именно такой тип кибератак является одним из наиболее опасных. В дальнейшем этот тип кибератак будем называть сетевыми кибератакими. Распознавание сетевых кибератак можно определить как процесс мониторинга (регистрации и анализа) параметров сетевого трафика на наличие признаков нарушения политики безопасности и попытки поставить под угрозу конфиденциальность, целостность, доступность, или обойти механизмы безопасности хоста или сети.

Традиционно для распознавания разнородных кибератак используются СПА, которые представляют комплекс средств, предназначенных для мониторинга происходящих в ИС событий для дальнейшего анализа с целью определения признаков нарушения безопасности объекта мониторинга [5, 12, 16, 23, 26, 42, 44, 54, 71, 79]. Также отметим, что близкое к СПА назначение имеют системы анализа защищенности (сканеры безопасности, системы поиска уязвимостей), обманные системы, системы контроля целостности, системы анализа журналов безопасности [30, 73, 93]. Однако такие системы имеют существенно другой характер обрабатываемой информации, в них необходимо использовать иные методы распознавания, а, следовательно, методология их построения и эксплуатации значительно отличается от СПА и в данной работе они не рассматриваются.

Особенностью СПА, которая предназначена для противодействия сетевым кибератакам, является исключительно мониторинг сетевого трафика. Подобные системы захватывает поток данных из сети, реализуют определенные методы анализа этих данных, сигнализируют о результатах анализа, а в случае обнаружения кибератаки могут вызывать срабатывание системы реагирования. Основным заданием такой системы является обнаружение сетевой кибератаки в режиме реального (или близкого к нему) времени. Кроме этого, могут быть решены дополнительные задания:

– Спрогнозировать возможные будущие сетевые кибератаки и выявить уязвимости для предотвращения их дальнейшего развития. Атакующий обычно выполняет ряд предварительных действий, таких как, например, сетевое

зондирование (сканирование) или другое тестирование для обнаружения уязвимостей целевой системы.

– Выполнить документирование существующих угроз.

– Обеспечить контроль качества администрирования с точки зрения безопасности, особенно в больших и сложных сетях.

– Получить полезную информацию о проникновениях, которые имели место, для восстановления и корректирования вызвавших проникновение факторов.

– Определить расположение источника сетевой кибератаки по отношению к локальной сети (внешние или внутренние атаки), что важно при принятии решений о расположении ресурсов в сети.

– Современные сетевые СПА обычно состоят из пяти функциональных компонентов, а именно:

– Модуль сбора данных – предназначен для регистрации параметров сетевого трафика, передаваемого в соответствии с различными протоколами.

– Модуль хранения данных, в котором накапливаются первичные статистические данные и результаты анализа.

– Модуль анализа – принимает информацию из модулей сбора и хранения данных и анализирует данные на наличие признаков сетевой кибератаки. Результат срабатывания модуля – распознанное состояние защищенности сетевого ресурса. В наиболее простом случае (системы определения атак) распознаются только два состояния – нормальное или состояние реализации сетевой кибератаки. В современных сетевых СПА дополнительно распознаются известные виды сетевых кибератак. Кроме этого, может рассчитываться вероятность (достоверность) каждого из заранее определенных состояний защищенности. Это увеличивает гибкость реализации защитных мероприятий.

– Модуль реагирования – активируется в том случае, когда анализирующий механизм определил наличие кибератаки. Если СПА действует автономно, то результатом срабатывания данного модуля является сигнализация о параметрах кибератаки. В случае интеграции СПА с системой реагирования на кибератаку реализуется некоторый набор защитных мероприятий.

– Консоль управления, предназначенная для настройки остальных модулей и системы в целом.

Типовая последовательность функционирования современной сетевой СПА показана на рисунке 1.1.

Распространенная классификация сетевых СПА осуществляется с точки зрения локализации параметров защищаемого РИС. К классу host-based относятся СПА, предназначенные для противодействия кибератакам, направленных на конкретный узел сети. В свою очередь СПА этого класса разделяются еще на три группы:

– Системы противодействия кибератакам на уровне прикладного программного обеспечения, обнаруживающие атаки на конкретные приложения (например, на веб-сервер). Примером такой системы является RealSecure OS Sensor или WebStalker Pro.

– Системы противодействия кибератакам на уровне операционной системы. Примером такой системы является RealSecure Server Sensor или Intruder Alert.
– Системы противодействия кибератакам на уровне системы управления базами данных, обнаруживающие атаки на конкретные системы управления БД.

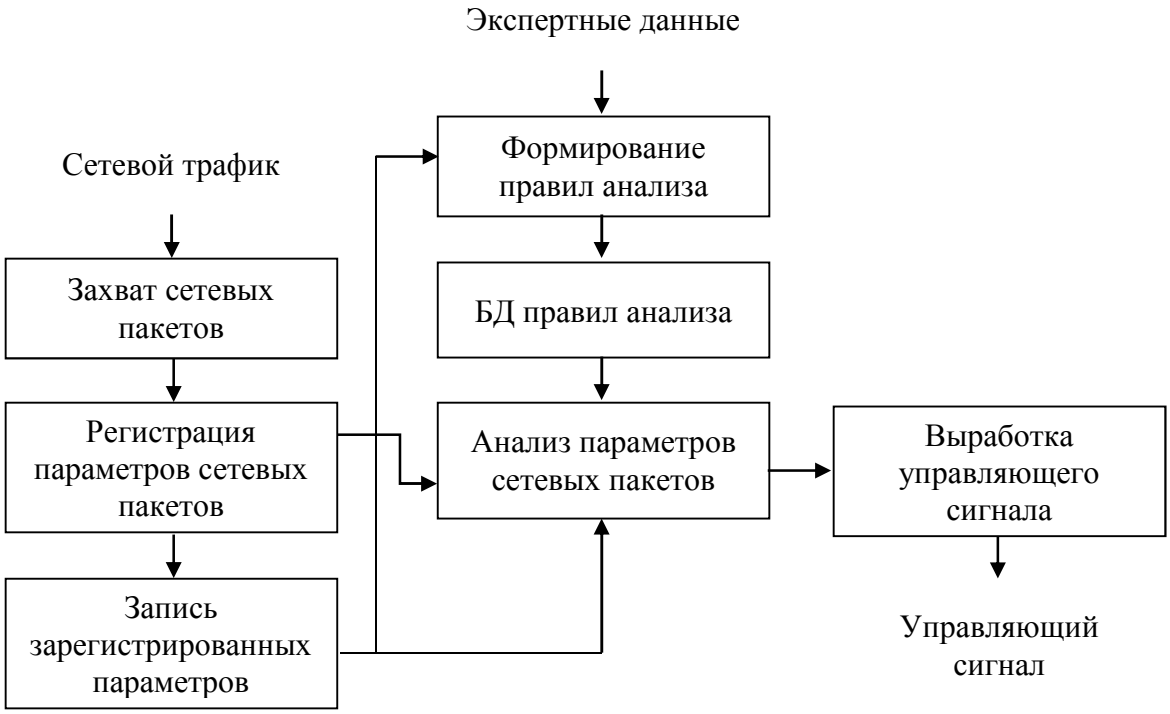

Рисунок 1.1 - Типовая последовательность функционирования сетевой СПА

Преимуществами host-based СПА являются: возможность следить за событиями локально относительно хоста, возможность функционировать в окружении, в котором сетевой трафик зашифрован, не требуют дополнительной функциональности сетевых устройств.

К недостаткам таких систем относят: отсутствие централизованного управления, малое покрытие мониторинга, отбор вычислительных мощностей защищаемого хоста, уязвимость к атакам на отказ в обслуживании.

Также выделяют СПА, предназначенные для противодействия сетевым кибератакам, направленных на всю сеть или сегмент сети. Такие СПА относятся к классу network-based. К этому классу относятся основные коммерческие СПА. В качестве примера можно назвать различные модификации аппаратно-программных комплексов Cisco IPS, IBM Proventia и StoneGate IPS. Также к этому классу сетевых СПА можно отнести один из наиболее известных некоммерческих программных комплексов Snort. Эти системы определяют кибератаки, захватывая и анализируя сетевые пакеты. Слушая сетевой сегмент, СПА класса network-based может перехватывать и анализировать сетевой трафик от нескольких хостов, которые присоединены к сетевому сегменту, и таким образом распознавать сетевые кибератаки на эти хосты.

К преимуществам таких СПА относят: возможность централизованного управления, отсутствие влияния на производительность сети. Недостатками являются: высокая ресурсоемкость, сложность настройки, невозможность анализа зашифрованной информации, невозможность распознавания результата атаки.

Как показывает практика и результаты [35, 36, 56, 69, 80], основным направлением усовершенствования современных СПА является повышение эффективности анализа сетевого трафика.

1.2 Анализ процесса противодействия сетевым кибератакам

В настоящее время процесс противодействия сетевым кибератакам реализуется с использованием двух основных методов: определения аномалий и определения злоупотреблений.

Работа анализатора при определении аномалий базируется на предположении, что признаком атаки есть отклонение текущих величин параметров сетевого трафика от величин, характерных для нормального состояния сетевых РИС (шаблонов нормального поведения). Для определения шаблона нормального поведения применяются статистические модели [38]. В некоторых СПА формируется комплексный показатель аномалий. При формировании данного показателя для определения взаимосвязей между показателями используются ковариационные матрицы. Также используется подход к определению аномалий с использованием метода прогноза событий, который позволяет выявить кибератаку на ранних этапах ее осуществления. Суть метода заключается в прогнозировании кибератаки на основе анализа предыдущих событий, связанных с объектом защиты [43].

Преимущества метода аномалий заключаются в следующем:

– Возможность определения сетевой кибератаки без знания конкретных деталей (сигнатуры).

– Детекторы аномалий могут создавать информацию, которая в дальнейшем будет использоваться для определения сигнатур сетевых кибератак.

К принципиальным недостаткам метода аномалий относят:

– Длительный срок формирования шаблона нормального поведения.

– Высокий уровень ложных срабатываний, обусловленный не достаточной адаптацией моделей шаблонов нормального поведения к сложной динамике параметров сетевого трафика [57].

Особенно сложно адаптировать модель шаблона нормального поведения к возможной реконфигурации защищаемой компьютерной сети. Даже незначительное изменение структуры или состава сети может повлечь за собой значительное изменение шаблона нормального поведения. Ситуация усугубляется тем, что составление и актуализация шаблона нормального поведения для конкретной ИС, требует значительных усилий высококвалифицированных специалистов. Поэтому использование метода аномалий для распознавания кибератак на сетевые ресурсы универсальных ИС весьма ограничено.

СПА, использующие метод определения злоупотреблений, анализируют последовательность событий, связанных с деятельностью объекта защиты и сравнивают их с образцами известных атак. Такие образцы называют шаблоном атак, а сам метод называют методом определения атак на основе сигнатур. По причине неполноты информации и наличия шумов при регистрации параметров безопасности трудности вызывает расчет соответствия шаблона атаки реальным событиям, касающихся объекта защиты. В базовом случае сетевая кибератака определенного вида распознается только в случае полного совпадения параметров сетевого запроса с соответствующей сигнатурой.

Кроме этого, для расчета соответствия применяются следующие подходы: экспертный, анализа переходов, моделирования атак. При применении экспертного подхода известные кибератаки описываются в виде некоторого набора правил, выполнение которых сигнализирует об их реализации. Подход на основании анализа переходов предусматривает представление сетевой кибератаки в виде последовательности переходов объектов защиты с одного состояния в другое [39, 54]. При моделировании кибератак предварительно сформированные последовательности событий, характерные для реализации кибератаки, сравниваются с текущими показателями. В результате сравнения формируется вывод о вероятности осуществления кибератаки. Часто используются статистические модели изменения параметров безопасности ИС при кибератаке.

В целом метод определения злоупотреблений позволяет достаточно эффективно выявлять кибератаки известных типов при низком показателе ложных срабатываний, но не позволяет выявить кибератаку, образец которой не известен. Вместе с тем сетевые кибератаки постоянно изменяются из-за индивидуальности подходов злоумышленников и регулярных изменений в программном обеспечении и аппаратных средствах целевых систем. Из-за большого количества видов указанных кибератак, динамичного и нечетко определенного характера их параметров очень сложно оперативно поддерживать в актуальном состоянии базу правил экспертной системы [2]. Поэтому в современных сетевых СПА для анализа сигнатур используются решения, базирующиеся на теории искусственного интеллекта [41, 45, 61, 62]. Из них наиболее апробированными являются нейросетевые модели и методы. Это объясняется доказанной способностью НС анализировать неполные/искаженные данные со сложным характером, обобщать накопленную статистическую информацию, а также оперативно реализовать расчет выходного сигнала в виде вероятности (достоверности) распознанного состояния. Еще одним достоинством применения анализатора на основе НС является возможность использования как в СПА класса network-based, так и в СПА класса host-based.

Общепринято, что современные СПА должны распознавать следующие классы сетевых кибератак: IP-спуфинг, отказ в обслуживании (Denial of Service), подбор парольных данных, атака на уровне приложений, сетевая разведка, переадресация портов.

Атака типа IP-спуфинг заключается в том, что злоумышленник, находящийся внутри локальной сети или вне ее, выдает себя за санкционированного пользователя. Как правило, для этого злоумышленник использует или IP-адрес, находящийся в пределах диапазона санкционированных IP-адресов, или авторизуется внешним адресом, которому разрешается доступ к определенным сетевым ресурсам. Атаки IP-спуфинга часто являются отправной точкой для прочих атак. Классический пример – атака DoS, которая начинается с чужого адреса, скрывающего истинную личность хакера.

Атаки с целью вызвать отказ в обслуживании или DoS-атаки отличаются от атак других типов тем, что нацелены на то, чтобы сделать РИС недоступным для обычного использования за счет превышения допустимых пределов функционирования сети, операционной системы или приложения. В случае использования некоторых серверных приложений (таких как Web-сервер или FTP-сервер) атаки DoS могут заключаться в том, чтобы занять все соединения, доступные для этих приложений, и держать их в занятом состоянии, не допуская обслуживания рядовых пользователей. В ходе атак DoS могут использоваться обычные Интернет-протоколы, такие как TCP и ICMP. Большинство атак DoS рассчитано не на программные ошибки или бреши в системе безопасности, а на общие слабости системной архитектуры. Некоторые атаки сводят к нулю производительность сети, переполняя ее нежелательными и ненужными пакетами или сообщая ложную информацию о текущем состоянии сетевых ресурсов. Когда атака данного типа проводится одновременно через множество устройств, то ее называют распределенной DoS-атакой (distributed DoS, DDoS).

Подбор парольных данных может быть реализован с помощью целого ряда методов, таких как простой перебор (brute force attack), троянский конь, IP-спуфинг и сниффинг пакетов. Хотя логин и пароль зачастую можно получить при помощи IP-спуфинга и сниффинга пакетов, хакеры нередко пытаются подобрать пароль и логин, используя для этого многочисленные попытки доступа. Такой подход носит название простого перебора.

Атаки на уровне приложений могут проводиться несколькими способами. Самый распространенный из них – использование хорошо известных уязвимостей серверного программного обеспечения. Используя эти уязвимости, можно получить доступ к компьютеру от имени пользователя, работающего с приложением (обычно это бывает не простой пользователь, а привилегированный администратор с правами системного доступа). Сведения об атаках на уровне приложений широко публикуются, чтобы дать администраторам возможность исправить проблему с помощью коррекционных модулей (патчей). Однако эта же информация позволяет злоумышленникам использовать выявленные уязвимости.

Сетевой разведкой называется сбор информации о сети с помощью общедоступных данных и приложений. Как правило, сетевая разведка производится при подготовке более опасной атаки. Сетевая разведка проводится в форме запросов DNS, эхо-тестирования и сканирования портов.

Запросы DNS помогают понять, кто владеет тем или иным доменом и какие адреса этому домену присвоены. Эхо-тестирование адресов, раскрытых с помощью DNS, позволяет увидеть, какие хосты реально работают в данной среде. Получив список хостов, злоумышленник использует средства сканирования портов, чтобы составить полный список услуг, поддерживаемых этими хостами. Это предоставляет злоумышленнику возможность провести анализ характеристик приложений, работающих на хостах.

Переадресация портов представляет собой разновидность злоупотребления доверием, когда взломанный хост используется для передачи через межсетевой экран трафика, который в противном случае был бы обязательно отбракован.

1.3 Анализ нейросетевых моделей и методов противодействия кибератакам

В работе [4] предложена **система обнаружения вторжений** с использованием **НС** (**СОВНС**), предназначенная для распознавания различных видов DDoS атак. Предложено использовать ДСП, структура которого адаптирована к виду DDoS атаки. В качестве примера на рисунке 1.2 представлена архитектура ДСП, предназначенного для распознавания сетевых кибератак, реализуемых на уровне протокола TCP. Предложен механизм работы НС, интегрированной в систему обнаружения.

Описание **нейросетевой системы обнаружения вторжений (НСОВ)**, взято из работы [90]. Предполагается использовать указанную систему для обнаружения сетевых кибератак, сигнатуры которых представлены в БД KDD-99. Для обнаружения использована НСМ типа МСП. Показано, что использование МСП целесообразно с точки зрения высокой вычислительной мощности. Кроме этого, в работе [90] представлено описание численных экспериментов, результаты которых доказывают эффективность применения МСП.

В работе [46] описан **бинарный нейросетевой метод** (**БНМ**) обнаружения сетевых атак. Особенностью метода является использование так называемой бинарной НС, положительными свойствами которой является возможность обработки входной информации, которая имеет фрактальную структуру, а также прямая вычислительная процедура обучения.

В работе [100] предложен метод **выделения сетевых атак** с типичного сетевого трафика **(ВСА)**. Метод предусматривает использование трехслойного персептрона. Выбор вида НСМ и оптимизация ее параметров обоснованы с позиций многокритериальной оптимизации. В качестве критериев оптимизации использованы критерии, отображающие гибкость и функциональность НСМ. Также в работе описана процедура предварительной обработки параметров сетевого трафика, которые применяются для составления учебной и тестовой выборки.

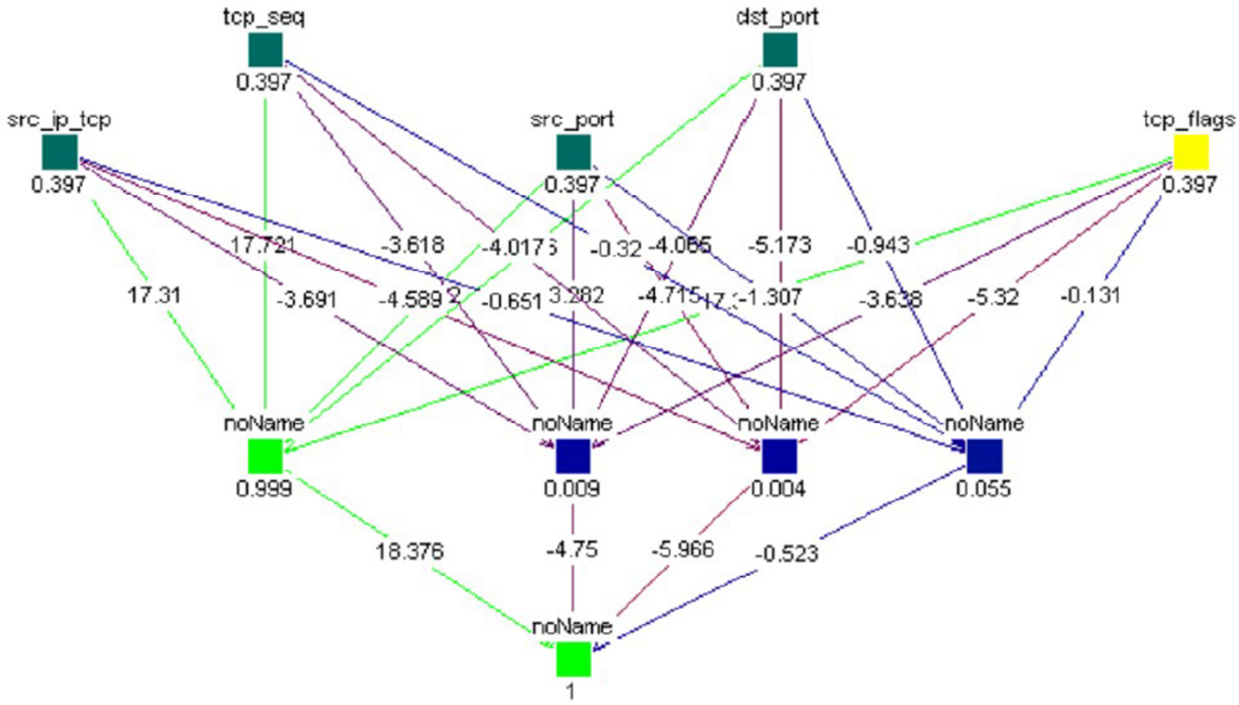

Рисунок 1.2 - Архитектура ДСП для распознавания сетевых кибератак на уровне протокола TCP

В работе [40] предложен метод использования **нейронной сети гибридной структуры (НСГС),** предназначенный для обнаружения сетевых атак на веб-сервер. Предложенная сеть – это комбинация НСМ вида ТК и МСП. Предусмотрено, что источником входных данных НСМ являются параметры сетевого трафика по каждому из протоколов IP, TCP, HTTP, HTTPS, CGI, SQLNet. Для преобразования параметров сетевого трафика к виду, пригодному для обработки в НСМ, использована специальная процедура, аналогичная процедуре обработки образов когнитивной графики для минимизации размерности входных данных. Кроме этого, на одном из этапов метода реализуется оптимизация процесса обучения НСМ. Критерием оптимизации является минимизация ошибки распознавания.

В работах [15, 67] описан **способ обнаружения DDoS-атак (СОД)**. Показаны результаты исследований, которые доказывают целесообразность применения нечетких НС, обучаемых с помощью нечетких экспертных правил. Способ в основном ориентирован на обнаружение DDoS-атаки типа SYN Flood. Обосновано применение пяти лингвистических переменных, характеризующих различные параметры сетевого трафика: X_1 – время получения пакетов, X_2 – процент пакетов из различных внешних ip-адресов, X_3 – процент пакетов с разных портов, X_4 – процент пакетов с поврежденными заголовками, Y – степень уверенности. Именно эти переменные являются входными параметрами НС. Разработаны предикатные правила вида: Если X_1= «большой» $\rightarrow$ Y $\rightarrow$ «высокая». Структура классификатора показана на рисунке 1.3.

На рисунке 1.3 символом обозначен нечеткий нейрон «ИЛИ», символом – нечеткий нейрон "И", а обозначение tLittle, tMiddle, tHigh, extraLittle, extraLots,

pLittle, pLots, dhLots соответствуют функциям активации нечетких переменных. Предложено представить нечеткий классификатор в виде НС с прямым распространением сигнала, которая учится с помощью модифицированного алгоритма обратного распространения ошибки. Модификация заключается в приспособлении классического алгоритма к нечетким нейронам «И» и «ИЛИ».

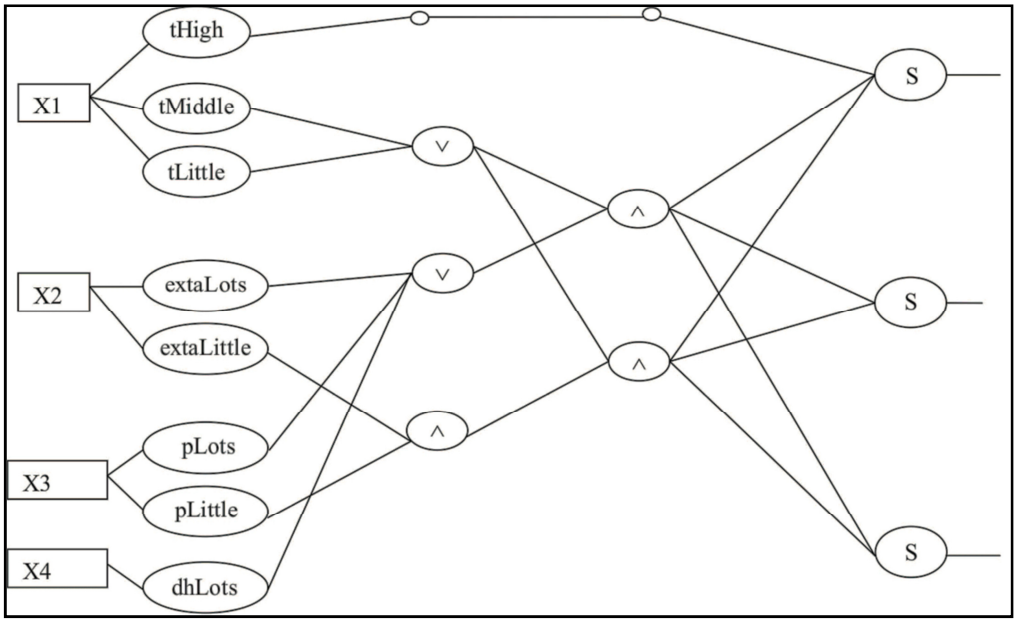

Рисунок 1.3 - Схема нечеткого классификатора для выявления SYN Flood-атак

Нейросетевой **подход** к **выявлению сетевых атак (ПВСА)** на компьютерные системы приведен в работе [32]. Акцент ставится на распознавание атак, сигнатуры которых представлены в БД KDD-99. Согласно данных этой БД, количество входных параметров – 41. В качестве критерия выбора оптимального типа нейросетевой модели предложено использовать минимум объема обучающей выборки. Путем анализа литературных источников определено, что к допустимым типам НС относятся ТК, МСП с одним скрытым слоем нейронов и сеть радиальной базисной функции (РБФ). Отмечено, что для ТК минимальный объем обучающей выборки (L) должен в 2 раза превышать количество входных нейронов (n), то есть:

$$L \approx 2n, \tag{1.1}$$

Для МСП и РБФ предложено объем обучающей выборки рассчитывать так:

$$L \approx W/\varepsilon, \tag{1.2}$$

где W – количество синаптических связей, ε – допустимая ошибка обучения.

В дальнейшем в [32] сделана попытка определить оптимальную структуру МСП. Заявлено, что определенное экспериментальным путем количество скрытых нейронов равно $m = 10$. При этом количество выходных нейронов равно 2. В соответствии с (1.1, 1.2), необходимый объем обучающей выборки ТК составляет $L = 82$ примеров, а для МСП и РБФ при $\varepsilon = 0.1$, $L = (m(n+3)+2)/\varepsilon = 4420$. Поэтому оптимальным типом НСМ выбрана ТК. Отметим, что правильность рассчитанных величин вызывает сомнения, ведь согласно теории НС [81], при заданной точности обучения количество скрытых нейронов МСП напрямую зависит от величины обучающей выборки. В дальнейшем в [32] проводится оптимизация структуры ТК. Неявно использован критерий максимизации точности обучения. Также использована процедура предварительной обработки входных параметров.

Адаптивная система обнаружения атак (АСОА) описана в работе [102]. Система предназначена для распознавания сетевых атак и базируется на совместной работе ТК и МСП, выполняющих задачи кластеризации и классификации данных. Обнаружение атак, которое проводится в несколько этапов, стало возможным благодаря тому, что в базу данных экспертной системы вносилась информация об изменениях в поведении конкретного объекта в течение некоторого отрезка времени. Доказывается, что оптимизация архитектуры позволит повысить точность и оперативность распознавания. В качестве входных данных использованы параметры сетевого трафика по протоколу TCP. Для обработки входных данных использован метод скользящего временного окна. ТК использована для предварительной обработки данных, поступающих на вход МСП, с целью их сжатия и повышения информативности. Приведено математическое выражение для расчета частоты определения нейрона в позиции (i, j) в качестве нейрона-победителя:

$$\Delta w_{jk}(i) = -\eta(y_n(i) - f(x_i))\varphi'(v_n(i))y_n, \qquad (1.3)$$

где η – количество раз, когда нейрон в позиции (i, j) был победителем, r – расстояние между центрами кластеров, x – длина входного вектора.

В дальнейшем эта частота используется для определения центров и границ кластеров. Структура МСП оптимизирована с точки зрения объема контролируемых ресурсов.

Нейросетевая технология **обнаружения** и **классификации сетевых атак (ОКСА)** описана в работе [77]. В технологии предложено использование трехслойной НС, которая обучается на основании метода обратного распространения ошибки. При этом для распознавания каждого вида сетевой атаки применяется отдельная НС. В качестве входных параметров предлагается использование параметров сетевого трафика по стеку протоколов TCP/IP. В качестве обучающей выборки предлагается использовать данные из базы данных KDD-99. Приведены словесное описание и фрагменты программного

кода для подготовки входных данных из этой базы данных к виду входных параметров НС. При этом одной из целей подготовки является уменьшение объема обучающей выборки НС. Описания подходов к оптимизации архитектуры и параметров нейросетевой модели отсутствуют.

Метод **распознавания аномалий сетевого трафика (РАСТ)**, разработан в работе [70]. Методом предусмотрено использование НС типа МСП. В качестве входных данных НС использованы параметры заголовков IP-дейтаграмм. Выбор архитектуры НС базируется на утверждении о высоких аппроксимационных возможностях МСП. МСП состоит из трех слоев нейронов. Количество нейронов первого (входного) слоя – 18, что равняется числу параметров заголовка IP-дейтаграммы. Количество нейронов в выходном слое 2. Выход нейрона №1 отвечает за наличие аномалии, а выход нейрона №2 – за безопасное состояние сетевого трафика. Приведены выражения для расчета количества нейронов в скрытом слое. Таким образом, метод предусматривает оптимизацию параметров архитектуры НС. Для упрощения создания репрезентативной выборки разработан метод уточняющих сигнатур, суть которого заключается во введении дополнительных искусственно созданных сигнатур, описывающих априорно аномальный трафик. Таким образом, в методе в неявном виде возможно использовать экспертные данные о сетевых атаках.

В работе [14] предложена **схема обнаружения сетевых атак** на основе комплексирования нейронных, иммунных и нейронечетких классификаторов (**СОСА**). Основными особенностями предлагаемой схемы является многоуровневый анализ сетевого трафика, а также использование различных адаптивных, в том числе и нейросетевых, модулей в процессе обнаружения атак. Для уменьшения числа используемых для анализа признаков предложено применять метод главных компонент. Проведены вычислительные эксперименты на двух открытых наборах данных с использованием различных способов комбинирования классификаторов.

В работе [31] предложен метод **построения совокупного классификатора трафика (ПСКТ).** Метод предназначен для иерархической классификации 22 типов сетевых атак, представленных в БД KDD-99. Для предобработки данных обучающей выборки использован метод главных компонент. Метод предусматривает использование 22 НСМ, каждая из которых должна быть обучена для распознавания конкретного типа сетевой атаки. НСМ представляет собой двухслойную НС с 12 входными нейронами и 2 выходными нейронами, один из которых отвечает за наличие, а второй за отсутствие атаки. В качестве скрытого слоя использован слой Кохонена. Для предотвращения ситуации, когда несколько НСМ одновременно сигнализируют о собственном типе атаки, на второй выход каждой из них передается минимальное евклидово расстояние между входным образом и эталоном:

$$E_j = \min_i \sqrt{\left(x_1 - w_{1,j}\right)^2 + \ldots + \left(x_{12} - w_{12,j}\right)^2}\,, \qquad (1.4)$$

где x_i – i-ый входной параметр, $w_{i,j}$ – весовой коэффициент связи между i-ым входным и j-ым скрытым нейроном.

В дальнейшем классифицируется атака, НСМ которой имеет минимальное евклидово расстояние. В работе также описаны механизмы оптимизации обучения и функционирования НСМ.

Алгоритм преобразования параметров трафика (АППТ) описан в работе [13]. Алгоритм предназначен для получения из сетевого трафика входных данных для нейросетевой системы обнаружения сетевых атак. В качестве входной информации указанного алгоритма используются параметры TCP-сессии. Преобразование параметров трафика применяется с целью уменьшения количества входных параметров НС и увеличения их информативности и реализуется с помощью математического аппарата, основанного на методе главных компонент. В АППТ оптимизация архитектуры и параметров нейросетевой модели не предусмотрена. Также отметим, что работы [11, 20] имеют аналогичный характер.

Нейросетевая система обнаружения компьютерных атак на основе анализа сетевого трафика **(НСОК)** описана в работе [50]. Задекларирована разработка метода анализа входящего трафика на основе трехслойной НС. Показано, что расчет топологи НСМ должен бать реализован с учетом меры Вапника-Червоненкиса вида:

$$K \times N \le VC_{\dim} \le N_w \times \left(1 + \lg N_n\right), \tag{1.5}$$

где N – размерность данных на входе, K – количество нейронов в скрытом слое, N_w – общее количество весов сети, N_n – общее количество нейронов сети.

Приведены результаты обучения и тестирования спроектированной НС, которые показывают возможность её успешного применения для решения задачи обнаружения сетевых компьютерных атак. Выдвинуто предположение, что наилучшие результаты могут быть получены в вычислительных системах, использующих ограниченный набор сетевого программного обеспечения, что позволяет более эффективно формировать признаки нормального поведения для обнаружения атак.

Нейросетевая **технология обнаружения сетевых атак (ТОСА)** на информационные ресурсы описана в [22]. В технологии предусмотрен модуль сжатия входных данных, который базируется на применении нейросетевого аналога метода главных компонент – рециркуляционной нейронной сети (РНС) с двумя слоями нейронов. Структура РНС показана на рисунке 1.4.

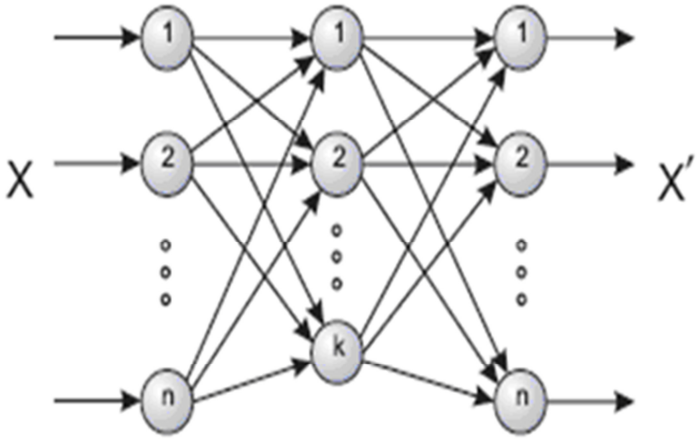

Рисунок 1.4 - Структура рециркуляционной нейронной сети

Первый слой, состоящий из *k* нейронов, позволяет управлять количеством информационных признаков (*x*), а второй слой, состоящий из n нейронов, позволяет проводить фильтрацию данных (*x*′). Настройки первого слоя позволяют получить сжатую до k признаков форму представления входного n-мерного объекта, то есть определить *k* главных компонент. В методе путем численных экспериментов доказана возможность использования ТК и МСП для обнаружения сетевых атак, сигнатуры которых представлены в базе данных KDD-99.

В работе [82] предложен **метод обнаружения вторжений** в информационную систему на основе нейронных сетей (**МОВ**). Указанный метод базируется на комбинированном применении методов поиска сигнатуры атаки и обнаружения аномалий в работе пользователя. В процессе разработки метода предложен подход к решению задачи классификации образов, заключающийся в представлении входных данных в виде сигнатур и отнесении их к классам атаки либо безопасным действиям пользователя. На основе модели безопасной работы пользователя в ИС и предложенного подхода к упрощению задачи обработки информации, синтезирована структура нейросетевой системы обнаружения атак. Также в работе проведены исследования по определению оптимальных параметров алгоритмов обучения НС, включающих выбор методов формирования репрезентативных обучающих множеств, оценку качества функционирования НС, а также поиск оптимальных значений параметров.

В работе [69] предложена **система обнаружения аномалий**, использующая аппарат искусственных иммунных систем и нейронных сетей (**СОАСТ**). Описываются структура, алгоритмы функционирования и программная реализация системы обнаружения аномалий. В системе предусмотрен модуль предварительной обработки входных параметров, источником данных для которого является статистика параметров сетевых запросов. Обосновано применение НСМ типа карты Кохонена. Описан механизм оптимизации параметров НСМ.

Работа [85] посвящена разработке **НСМ обнаружения DDOS-атак** (**НСМРЧ**), обучаемой с применением **метода роя частиц**. Использован МСП с двумя скрытыми слоями, построенный на сигмоидальной логистической функции активации. Количество входных сигналов – 28. В скрытых слоях количество нейронов составляет 28 и 14, в выходном слое содержится 2 нейрона. Представленная модель выдает ответы (1, 0) и (0, 1), характеризующие наличие или отсутствие атаки. Для оптимизации параметров обучения МСП применен метод роя частиц, представляющий собой эвристический метод оптимизации, который, за счет имитации социального поведения роя биологических объектов, не требует знания точного градиента оптимизируемой функции. В работе разработан соответствующий математический аппарат, показаны результаты численных экспериментов.

В работе [88] предложена **НСС** обнаружения сетевых кибератак на основе распознавания **аномального сетевого трафика** (**НССАСТ**). Использована НСМ с прямым прохождением сигнала, обучаемая с помощью алгоритма обратного распространения ошибки. Предложена процедура оптимизации метода обучения. В качестве источника данных использована БД NSL-KDD.

Система обнаружения вторжений на базе **глубокой нейронной сети** (**СОВГНС**) разработана в работе [99]. Система предназначена для использования в системе защиты информации автомобилей. В системе предусмотрены модули предварительной обработки входных данных НСМ и оптимизации процесса обучения. С помощью экспериментов показано, что использование глубокой НСМ позволяет значительно повысить точность обнаружения вторжений.

Нейросетевая методология оценки параметров безопасности Интернет-ориентированных информационных систем (**НМОПБ**) представлена в работе [33]. Среди проанализированных данная работа является наиболее фундаментальной. В ней получили дальнейшее развитие теоретические положения построения НСР оценки ПБ, которые заключаются в разработанных подходах к распознаванию постепенных и неожиданных кибератак, определении оптимального вида НСМ, целесообразности применения НСР, классификации статистически подобных кибератак, применении продукционных правил для представления экспертных знаний, параметрах оценки эффективности НСР. Также разработаны модели создания и использования НСР оценки ПБ, которые за счет применения разработанных теоретических положений позволяют: определить перечень оцениваемых ПБ, а также уменьшить ресурсоемкость создания НСМ. На основе указанных моделей разработан ряд методов, позволяющих повысить эффективность использования НСР. Так, метод представления экспертных знаний для НСР оценки ПБ, позволяет обеспечить оперативность распознавания и расширить множество типов кибератак, для которых отсутствуют статистические данные. Метод определения временных характеристик использования НСР оценки ПБ благодаря использованию разработанных аналитических зависимостей между ожидаемыми и допустимыми сроками разработки обеспечивает возможность

определения целесообразности применения указанных средств. Метол определения эффективности разработки нейросетевых средств оценки параметров безопасности за счет применения предложенных параметров оценки эффективности и сформированного интегрального показателя эффективности позволяет выбрать наиболее эффективное средство. Применение метода позволило определить, что в известных НСР распознавания сетевых кибератак недостаточно полно разработан механизм формирования обучающей выборки при кодировании ожидаемого выходного сигнала, не учитывается близость эталонов распознаваемых классов. Кроме этого, такие средства недостаточно адаптированы к применению современных типов НСМ. На основе взаимосвязанного использования разработанных подходов, моделей и методов разработана комплексная методология нейросетевой оценки ПБ, которая позволяет значительно расширить функциональные возможности НСР и выбрать из них наиболее эффективное.

С позиций сформулированной цели исследования наибольший интерес в этой работе представляет предложенный перечень параметров, характеризующих эффективность НСР. Отметим, что недостаток этого перечня вытекает из достаточно общего характера работы [33], которая направлена на оценку ПБ для распознавания широкого круга кибератак и уязвимостей Интернет-ориентированных ИС. Поэтому, с учетом указанных ранее ограничений, при оценке НСР распознавания кибератак на сетевые РИС предложенный перечень является во многом избыточным. В то же время в нем недостаточно полно учтены особенности оценки эффективности НСР при распознавании сетевых кибератак.

1.4 Пути совершенствования нейросетевых средств противодействия кибератакам

В результате проведенного анализа установлено, что повышение эффективности современных НСМ распознавания сетевых кибератак идет путем обеспечения в них определенных возможностей, которые характеризуются с помощью параметров, представленных в таблице 1.1. Также сделан вывод о том, что эффективность НСР распознавания в значительной степени зависит от полноты и представительности обучающей выборки, которая применяется для обучения современных НСМ, заложенных в их основе, а также от того, учитывается ли при кодировании ожидаемого выходного сигнала близость эталонов распознаваемых классов сетевых кибератак. Данный вывод сформулирован на основании анализа результатов работы [47], в которой обоснован метод применения НС для распознавания голосовых сигналов. За счет этого, предложено использование параметров $E_{ов}$ и $E_{квс}$, описание которых также представлено в таблице 1.1. В дальнейшем представленный в таблице 1.1 перечень параметров может быть расширен.

Таблица 1.1 - Параметры оценки эффективности нейросетевых средств

Название параметра	Описание параметра
$E_{по}$	Предварительная обработка входящих параметров
$E_{ота}$	Оптимизация типа архитектуры
$E_{опа}$	Оптимизация параметров архитектуры
$E_{омо}$	Оптимизация метода обучения
$E_{вэп}$	Возможность обучения с помощью экспертных правил
$E_{пна}$	Возможность применения в методе перспективных типов нейросетевых архитектур
$E_{оцп}$	Возможность принципиальной оценки целесообразности применения НС для решения поставленной задачи
$E_{ов}$	Наличие процедуры формирования обучающей выборки из разнородных статистических данных
$E_{квс}$	Наличие процедуры кодирования ожидаемого выходного сигнала НСМ, учитывающей близость эталонов распознаваемых классов сетевых кибератак

Величины предложенных параметров в первом приближении можно оценить по бинарной шкале: 0 или 1. Параметр равен 0, когда соответствующая возможность в НСР не обеспечивается и 1 в противоположном случае. Для проанализированных случаев величины указанных параметров приведены в таблице 1.2. При этом для всех проанализированных методов $E_{квс} = 0$. То есть в большинстве из проанализированных методов не реализована процедура формирования обучающей выборки. Кроме того, использование предложенных критериев позволяет определить интегральный показатель эффективности НСР (E_Σ) с помощью следующего выражения:

$$E_\Sigma = \sum_{i=1}^{9} \alpha_i E_i \,, \tag{1.6}$$

где α_i – весовой коэффициент i-го критерия.

Определить наиболее эффективное НСР можно, воспользовавшись выражением:

$$\max_{E_i} = \{ E_1, E_2, .. E_I \}, \tag{1.7}$$

где I – количество видов НСМ, E_i – интегральный показатель эффективности i-го НСР.

В общем случае определение весовых коэффициентов требует отдельного исследования, а в базовом варианте можно предположить, что $\alpha_i = 1$.

Таблица 1.2 - Величины параметров, характеризующих эффективность нейросетевых моделей и методов

Метод	Параметр								
	$E_{по}$	$E_{ота}$	$E_{опа}$	$E_{омо}$	$E_{вэп}$	$E_{пна}$	$E_{оцп}$	$E_{ов}$	$E_{квс}$
АППТ	1	0	0	0	0	0	0	0	0
НСОВ	0	1	0	0	0	0	0	0	0
ТОСА	1	1	0	0	0	0	0	0	0
РАСТ	0	1	1	0	0	0	0	0	0
ВСА	0	1	1	0	0	0	0	0	0
ПСКТ	1	0	0	0	0	0	0	0	0
ПВСА	1	1	0	1	0	0	0	0	0
АСОА	1	1	1	0	0	0	0	0	0
СОД	0	1	0	1	0	0	0	0	0
БНМ	0	1	0	1	0	0	0	1	0
ОКСА	1	0	0	0	0	0	0	1	0
МОВ	1	0	0	0	0	0	0	0	0
НСОК	1	0	0	0	0	0	0	0	0
НСГС	1	0	0	0	0	0	0	1	0
СОСА	1	0	0	0	0	0	0	1	0
НМОПБ	1	1	1	1	1	1	1	0	0
СОВНС	1	0	1	0	0	0	0	0	0
СОАСТ	1	1	1	1	0	0	0	0	0
НСМРЧ	1	0	0	1	0	0	0	0	0
НССАСТ	1	0	1	1	0	0	0	0	0
СОВГНС	1	0	1	1	0	1	0	0	0

Отметим, что практическая ценность данных таблицы 1.2 состоит в обрисовке недостатков и перспектив совершенствования современных нейросетевых методов и моделей. Например, величина $E_{ота} = 0$ свидетельствует о том, что к недостаткам метода АППТ можно отнести недостаточную оптимизацию вида архитектуры НСМ. Это свидетельствует о возможности соответствующего совершенствования указанных методов. При этом величина параметра E_{Σ} позволяет оценить интегральную эффективность нейросетевого метода. Также в результате проведенного анализа доказано, что в современных СРК в основном используются классические типы НСМ, которые в той или иной степени адаптированы к условиям поставленной задачи. Это позволяет сузить круг допустимых современных видов НСМ, что в свою очередь позволяет повысить оперативность определения модели, оптимальной с точки зрения поставленной задачи. Таким образом, появляется возможность повышения оперативности создания соответствующих СРК.

В результате анализа научно-практических работ посвященных разработке и эксплуатации систем распознавания кибератак на сетевые ресурсы информационных систем общего назначения показано, что одним из основных путей развития указанных систем является внедрение в них методов анализа сетевого трафика, базирующихся на современных решениях теории искусственных нейронных сетей. Также определено, что актуальной задачей является внедрение нейросетевых средств распознавания в сетевые системы распознавания кибератак использующие метод определения злоупотреблений для распознавания кибератак типа IP-спуфинг, отказ в обслуживании, подбор парольных данных, атака на уровне приложений, сетевой разведки и переадресации портов.

Также анализ нейросетевых методов распознавания сетевых кибератак позволил определить, что повышение их эффективности во многом связано с адаптацией к ожидаемым условиям эксплуатации, которые во многом зависят от условий формирования примеров обучающей выборки. Еще одним направлением повышения эффективности является использование современных видов нейросетевых моделей, позволяющих в различных условиях эксплуатации обеспечить высокую точность распознавания кибератак. В результате возникает необходимость усовершенствования методологической базы нейросетевого распознавания сетевых кибератак и разработке на этой базе метода создания обучающей выборки и метода создания, соответствующих нейросетевых средств. Для апробации предложенных решений целесообразно разработать нейросетевую систему и провести исследование ее эффективности.

2 РАЗВИТИЕ МЕТОДОЛОГИЧЕСКОЙ БАЗЫ НЕЙРОСЕТЕВОГО ПРОТИВОДЕЙСТВИЯ КИБЕРАТАКАМ НА СЕТЕВЫЕ РЕСУРСЫ ИНФОРМАЦИОННЫХ СИСТЕМ

2.1 Концептуальная модель обеспечения эффективности нейросетевого противодействия кибератакам

Результаты исследований, проведенных в первом разделе данной диссертационной работы, указывают на то, что важным направлением развития СПА на сетевые РИС является внедрение в них НСР распознавания кибератак. Для этого необходимо решить научное задание нейросетевого распознавания кибератак на основании анализа подконтрольных на эксплуатации параметров функционирования ИС.

Особенностью сформулированного задания является необходимость теоретического обоснования характеристик нейросетевых моделей и методов, адаптированных к условиям внедрения в современные СПА. К указанным условиям относятся допустимый срок разработки, возможность привлечения трудовых ресурсов, наличие доступа к базам данных шаблонов атак и шаблонов нормального поведения, необходимых для обучения НСМ, особенности системы контроля функциональных параметров ИС и допустимый объем вычислительных ресурсов, которые потребляет СПА.

Решение данного научного задания позволит решить практическую задачу распознавания сетевых кибератак на основании схожести параметров сетевого трафика с сигнатурами (шаблонами) известных сетевых кибератак. При этом задачи фиксации параметров функционирования ИС, предварительной фильтрации таких параметров, а также сигнализации о выявленных кибертаках считаются решенными и в данной диссертационной работе не рассматриваются [27, 48, 95].

В соответствии с рекомендациями [18, 29, 60], отправным пунктом решения сформулированного задания стала разработка концептуальной модели обеспечения эффективности нейросетевого распознавания кибератак на сетевые РИС. В связи с тем, что ожидаемый практический результат диссертационной работы предусматривает создание программно-аппаратного комплекса, то для определения эффективности процесса нейросетевого распознавания кибератак на сетевые РИС предусмотрено использовать терминологию в области защиты информации, компьютерной и программной инженерии.

Также определено, что в контексте задачи данного диссертационного исследования концептуальная модель, прежде всего, предназначена для формализации причинно-следственных связей, которые свойственны процессу распознавания кибератак на сетевые РИС, определенных необходимостью повышения уровня защищенности современных ИС. Кроме этого, в концептуальной модели учтено:

– Условия функционирования НСР распознавания кибератак на сетевые РИС, определяемые характером взаимодействия отдельных частей СПА и компонентами ИС.

– Необходимость реализации эффективного использования НСМ для распознавания кибератак и основные направления улучшения его функционирования.
– Возможность управления НСР и определение его настраиваемых переменных.

На следующем этапе построения концептуальной модели с учетом общепринятой технологии использования НСМ определено, что процесс нейросетевого распознавания кибератак должен предусматривать формирование параметров учебных примеров, формирование обучающей выборки, определение вида и параметров НСМ и использование НСМ для распознавания.

Использование данного утверждения позволило построить показанную на рисунке 2.1 диаграмму декомпозиции нейросетевого распознавания кибератак на сетевые РИС.

Рисунок 2.1 - Диаграмма декомпозиции нейросетевого распознавания кибератак

Назначение составляющих данной диаграммы состоит в следующем:
– Формирование параметров учебных примеров – определение для каждого вида кибератак множества входных и выходных параметров и способа их кодирования.

– Формирование обучающей выборки – определение такого множества учебных примеров, которое отвечает эталонам кибератак. Количество, качество и номенклатура примеров должны быть достаточными для обучения НСМ.
– Определение вида и параметров НСМ – определение для использования такого вида НСМ, с такими параметрами, которые наиболее полно отвечают условиям задачи распознавания кибератак на сетевые ресурсы конкретной ИС.
– Использование НСМ – распознавание кибератак на сетевые РИС. Следует учесть, что использование НСМ влечет за собой дополнительную нагрузку на ИС, что может вызвать исчерпание вычислительных ресурсов.

Следующим этапом создания концептуальной модели стала разработка показанной на рисунке 2.2 схемы компонентов НСС распознавания кибератак. В схеме учтены особенности реализации НСС, предназначенные для распознавания кибератак на сетевые РИС, и результаты раздела 1, которые касаются недостатков известных НСР для распознавания кибератак на сетевые РИС.

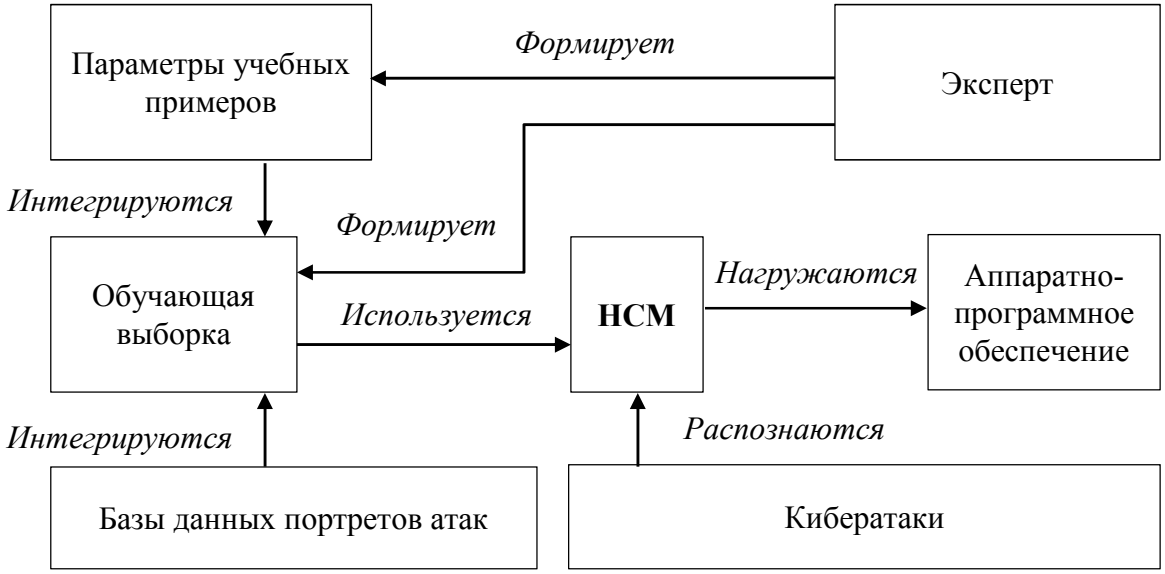

Рисунок 2.2 - Схема взаимодействия компонентов НСС распознавания кибератак

Таким образом, в процессе разработки учтено:
– Несовершенство методов формирования параметров обучающих примеров для НСМ, предназначенных для распознавания кибератак на сетевые РИС.
– Длительный период формирования обучающей выборки для НСМ в случае ограниченного доступа к базам данных портретов кибератак.
– Сложность доступа к существующим базам данных портеров кибератак.
– Дополнительная нагрузка на аппаратно-программное обеспечение ИС за счет функционирования НСР.

Поэтому в схеме предусмотрена возможность формирования параметров обучающих примеров и учебной выборки с помощью экспертных данных. Анализ данных, показанных на рисунке 2.1 и рисунке 2.2, позволяет утверждать, что на эффективность нейросетевого распознавания кибератак на

сетевые РИС влияют ряд факторов, показанных на рисунке 2.3.

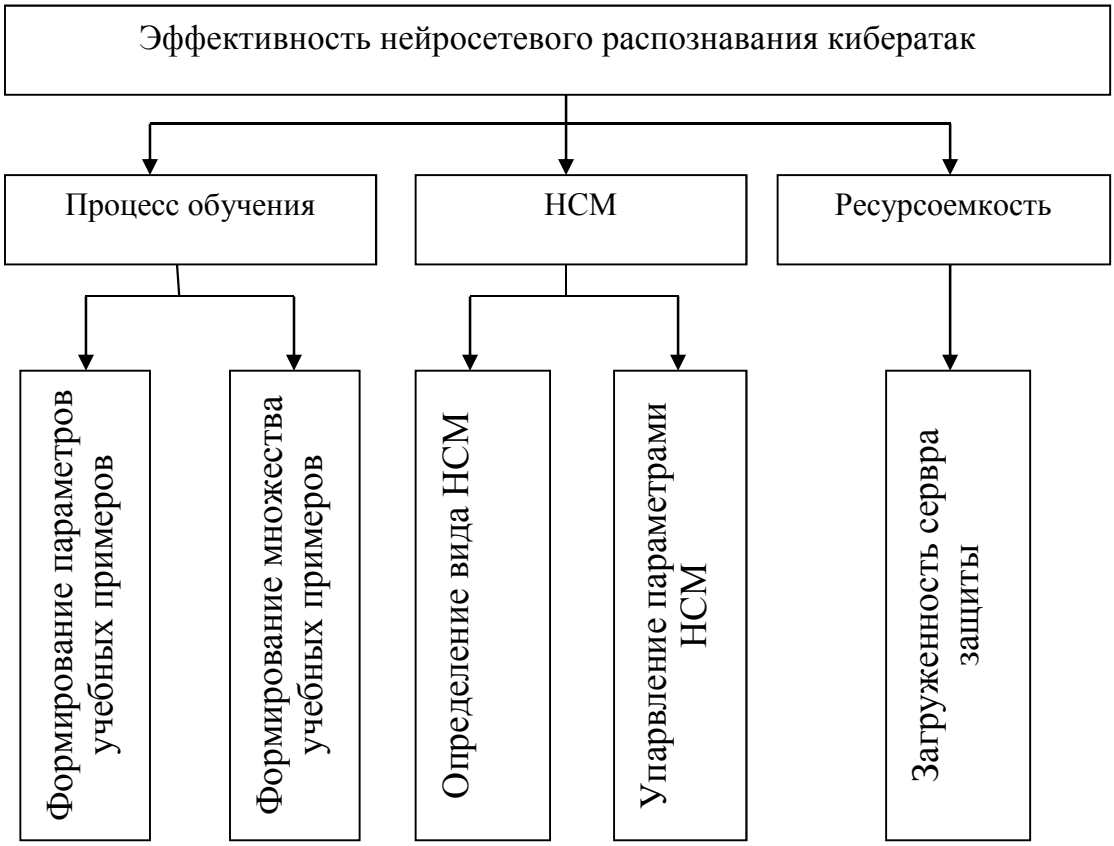

Рисунок 2.3 - Факторы, влияющие на эффективность распознавания

Кроме этого, можно утверждать, что эффективность нейросетевого распознавания целесообразно оценивать как с точки зрения эффективности процесса использования НСР, так и с точки зрения эффективности процесса обучения НСМ. При этом показатели эффективности должны отображать длительность, ресурсоемкость и точность названных процессов. Таким образом, обоснованы показанные на рисунке 2.4 показатели оценки эффективности нейросетевого распознавания кибератак на сетевые РИС.

В результате определено, что в аналитическом виде концептуальную модель обеспечения эффективности процесса нейросетевого распознавания кибератак на сетевые РИС можно отобразить с помощью выражений:

$$E_{\Sigma} = f\left(E_{HCP}, E_{OB}\right), \tag{2.1}$$

$$E_{HCP} = f\left(e_1, e_2, e_3\right), \tag{2.2}$$

$$E_{OB} = f\left(e_4, e_5\right), \tag{2.3}$$

где E_{Σ} – интегральная эффективность процесса, $E_{\text{НСР}}$ – эффективность создания и использования НСР, $E_{\text{ОВ}}$ – эффективность создания обучающей выборки, e_1 –

определение эффективных видов НСМ, e_2 – определение параметров НСМ, e_3 – ресурсоемкость использования НСР, e_4 – определение параметров обучающих примеров, e_5 – формирование обучающей выборки.

Рисунок 2.4 - Показатели оценки эффективности нейросетевого распознавания

Анализ разработанной концептуальной модели позволяет утверждать, что для эффективного нейросетевого распознавания кибератак необходимо дополнить методологическую базу рядом принципов и базирующихся на них моделей процессов использования НСР. В дальнейшем необходимо применить полученные элементы методологической базы для разработки нейросетевых моделей и методов распознавания сетевых кибератак.

2.2 Принципы использования нейронных сетей

Принцип допустимости использования вида НСМ для распознавания кибератак на сетевые РИС. Как показывают результаты первого раздела, основным фактором, который влияет на формирование множества допустимых видов НСМ, является возможность их эффективного обучения. Для этого за допустимое время необходимо: определить множество входных и выходных параметров НСМ, провести кодирование указанных параметров, создать обучающую выборку, реализовать процесс обучения. Выполнение первой процедуры реализуется на подготовительном этапе разработки НСР. Поэтому внимание акцентировано на вторую и третью процедуры. При этом приемлемый срок создания обучающей выборки и обучения НСМ определяется на основании требований к созданию СПА:

$$t_{\Sigma} \le t_d , \tag{2.4}$$

где t_{Σ} – общий срок обучения НСМ распознавания кибератак, t_d – приемлемый срок создания НСР распознавания кибератак.

Таким образом, принцип допустимости использования *i*-го вида НСМ можно задать с помощью правила:

$$If \;\; t_{\Sigma}(net_i) \le t_d \rightarrow net_i \in \boldsymbol{Net_d}, \tag{2.5}$$

где net_i – *i*-ый вид НСМ; $\boldsymbol{Net_d}$ – множество допустимых видов НСМ.

Принцип определения множества эффективных видов НСМ для распознавания кибератак на сетевые РИС. В соответствии с результатами п. 1.3 и выводами [8 с. 80 - 82, 37, 63] для определения множества видов НСМ, которые обеспечат эффективное распознавание кибератак на сетевые РИС, предлагается использовать процедуру вида:

$$\mathbf{Net_a} \rightarrow \mathbf{Net_d} \rightarrow \mathbf{Net_e}, \tag{2.6}$$

где $\boldsymbol{Net_a}$ – множество доступных видов НСМ, $\boldsymbol{Net_d}$ – множество допустимых видов НСМ, $\boldsymbol{Net_e}$ – множество эффективных видов НСМ.

Принцип оценивания эффективности вида НСМ, предназначенной для распознавания кибератак на сетевые РИС. По аналогии с [21, 55, 64] будем считать, что среди множества допустимых *i*-ый вид НСМ является наиболее эффективным, если для него функция эффективности примет максимальное значение:

$$\max_{V_i} = \{ V_1, V_2, .. V_I \}, \tag{2.7}$$

где I – количество видов НСМ, V_i – функция эффективности *i*-го вида НСМ.

Расчет функции эффективности выполняется так:

$$V_i = \sum_{k=1}^{K} \alpha_k R_k(net_i), \quad net_i \in \boldsymbol{Net_d}, \; i = 1,...I. \tag{2.8}$$

где $\alpha_k = [0..1]$– весовой коэффициент *k*–го критерия эффективности; net_i – *i*-ый вид НСМ; $\boldsymbol{Net_d}$ – множество допустимых видов НСМ; K – количество критериев эффективности; $R_k(net_i)$ – значение *k*-го критерия для НСМ *i*-го вида.

В соответствии с результатами [9 с. 27 - 30, 24, 59, 75], под *k*-ым критерием определения эффективности вида НСМ будем понимать меру обеспечения в НСМ *k*-ого требования задачи распознавания сетевых кибератак.

Принцип определения ожидаемого выходного сигнала НСМ для портретов кибератак. Значение выходного сигнала должно отображать схожесть обучающих примеров. В противоположном случае обучение НСМ может существенно ухудшиться. Поэтому выходной сигнал для портретов кибератак предлагается описать выражением:

$$\boldsymbol{Y_{\Phi}} = f\left(\boldsymbol{d}_{\Phi}\right), \tag{2.9}$$

где $\boldsymbol{Y_{\Phi}}$ – ожидаемые выходные сигналы НСМ для кибератак вида $\boldsymbol{\Phi}$; $\boldsymbol{d}_{\Phi}$ – множество мер схожести между компонентами $\boldsymbol{\Phi}$.

Принцип использования экспертных знаний для формирования обучающей выборки. По аналогии с [28, 65], данный принцип предусматривает, что обучающие примеры можно формировать на основании экспертных знаний касательно кибератак в виде продукционных правил вида:

$$If\ x_1 \in \left[X_1^{\min}, X_1^{\max}\right]_l \wedge \ldots x_k \notin \left[X_k^{\min}, X_k^{\max}\right]_l \ldots \wedge x_K \in \left[X_K^{\min}, X_K^{\max}\right]_l \rightarrow Y, \tag{2.10}$$

где $x_1,\ldots x_K$ – параметры, идентифицирующие кибератаку, $\left[X_1^{min}, X_1^{max}\right],\ldots,\left[X_K^{min}, X_K^{max}\right]$ – заданные диапазоны $x_1,\ldots x_K$, K – количество идентифицирующих параметров, Y – результат продукционного правила (ожидаемая кибератака).

Разработанные принципы послужили основой для создания моделей процессов использования НСМ для распознавания кибератак.

2.3 Модель правил определения эффективных видов нейросетевых моделей

Детализировав выражение (2.5), которое соответствует принципу допустимости использования *i*-го вида НСМ для распознавания кибератак на сетевые РИС, получим:

$$t_{\Sigma}\left(net_i\right) = t_v + t_l\left(net_i\right), \tag{2.11}$$

где t_v – время создания обучающей выборки, $t_l\left(net_i\right)$ – время определения параметров модели для *i*-го вида НСМ.

Отметим, что в первом приближении значение $t_l\left(net_i\right)$ примерно равно времени определения весовых коэффициентов синаптических связей НСМ. При этом, с точки зрения решения задачи определения только принципиальной возможности использования определенного вида НСМ, создание обучающей выборки сводится к формированию такого количества учебных примеров, которое считается достаточным для качественного обучения НСМ.

В соответствии с [17], это количество зависит от количества входных параметров НСМ и в базовом случае рассчитывается так:

$$P_{min} \approx 10 N_x, \tag{2.12}$$

где P_{min} – минимально допустимое количество учебных примеров, N_x – количество входных параметров НСМ.

Также можно принять, что:

$$t_v = \bar{t}_v P_{min}, \tag{2.13}$$

где $\bar{t}_v$ – среднее время создания одного учебного примера.

Очевидно, что величина $\bar{t}_v$ является индивидуальной для конкретного сетевого РИС и зависит от многих факторов: организации процесса создания обучающей выборки, аппаратно-программного обеспечения и т.д. Определить величину $\bar{t}_v$ возможно путем экспертного оценивания.

После подстановки (2.12) в (2.13) получим:

$$t_v = 10 \bar{t}_v N_x. \tag{2.14}$$

В свою очередь, в соответствии с данными [66, 72], для приблизительной оценки времени определения значений весовых коэффициентов синаптических связей НСМ необходимо учесть вид этой модели, скорость ее аппаратно-программной реализации, количество учебных примеров, количество входных и выходных параметров, а также допустимую величину ошибки обучения.

При определенной структуре для НСМ *i*-го вида продолжительность процесса определения весовых коэффициентов можно оценить так:

$$t_l(net_i) = \tau \times L_i \times W_i \times K_{o,i}, \tag{2.15}$$

где τ – продолжительность одной учебной итерации для одной синаптической связи; W_i – количество синаптических связей для *i*-го вида НСМ; L_i - количество нейронов; $K_{o,i}$ – количество итераций в процессе обучения. При этом

$$K_{o,i} = f_{o,i}(\varepsilon, P), \tag{2.16}$$

где $f_{o,i}(\varepsilon, P)$ – зависимость количества итераций от ошибки обучения и количества учебных примеров для НСМ *i*-го вида; ε – допустимая ошибка обучения НС; P – количество учебных примеров.

В соответствии с [10 с. 8 - 9], при приблизительных расчетах для множества видов НСМ $\boldsymbol{net_1}$, которое складывается из НСМ на основе PNN, сети адаптивной резонансной теории, ТК, РБФ, АНС, которые обучаются путем непосредственного запоминания учебных примеров или с использованием правил Гебба, Ойя, коррелятивного или дельта правил, для оценки продолжительности обучения можно использовать выражение (2.17). Для оценки продолжительности обучения множества видов НСМ $\boldsymbol{net_2}$, которое состоит из НСМ, которые базируются на МСП и обучаются с помощью градиентных методов или генетических алгоритмов, можно использовать выражение (2.18).

$$t_l(\boldsymbol{net_1}) \approx k_1 \tau e^{-\varepsilon} P\left(N_x + N_y\right), \tag{2.17}$$

$$t_l(\boldsymbol{net_2}) \approx k_2 \tau e^{-\chi\varepsilon} P^2\left(N_x + N_y\right)^2, \tag{2.18}$$

где $t_l(\boldsymbol{net}_1)$– продолжительность определения весовых коэффициентов для видов НСМ, которые относятся к $\boldsymbol{net}_1$; $t_l(\boldsymbol{net}_2)$– продолжительность определения весовых коэффициентов для видов НСМ, которые относятся к $\boldsymbol{net}_2$; k_1 – коэффициент пропорциональности для видов НСМ, которые входят в $\boldsymbol{net}_1$; τ – продолжительность одной вычислительной операции процесса обучения; P – количество учебных примеров; N_y – количество выходных параметров; k_2 – коэффициент пропорциональности для видов НСМ, которые входят в $\boldsymbol{net}_2$; χ – эмпирический коэффициент.

Отметим, что выражения (2.17, 2.18) получены при условии последовательного расчета сигналов искусственных нейронов, которые входят в состав НСМ, что характерно при ее общепринятой реализации на распространенной компьютерной технике. Кроме этого, принята предпосылка, что структура НСМ и вычислительные возможности вида НСМ достаточны для получения допустимой ошибки обучения.

Как свидетельствуют результаты первого раздела, с точки зрения распознавания кибератак на сетевые РИС наиболее перспективными видами НСМ являются РБФ, ТК, МСП, PNN, ГНС. Для РБФ, ТК и PNN приблизительную продолжительность обучения возможно оценить с помощью выражения (2.17), а для МСП и ГНС целесообразно использовать выражение (2.18). Соответственно [7 с. 21-24], для заданной программной реализации НСМ продолжительность одной вычислительной операции процесса обучения (τ) в основном зависит от вычислительной мощности аппаратного обеспечения контура распознавания кибератак в системе защиты сетевых РИС.

Допустимую ошибку обучения НСМ (ε) можно рассчитать с помощью [72, 101] на основании требований к точности распознавания кибератак на сетевые РИС. В первом приближении величины τ и ε возможно определить путем экспертного оценивания.

При определении принципиальной возможности использования НСМ целесообразно ориентироваться на минимально допустимое количество учебных примеров. Учитывая в (2.17) и (2.18) зависимость (2.12), получим:

$$t_l(\boldsymbol{net}_1) \approx 10 k_1 \tau e^{-\varepsilon} N_x (N_x + N_y), \tag{2.19}$$

$$t_l(\boldsymbol{net}_2) \approx 100 k_2 \tau e^{-\chi\varepsilon} N_x^2 (N_x + N_y)^2. \tag{2.20}$$

Подстановка (2.14, 2.19) и (2.14, 2.20) в (2.11) позволяет детализировать выражение для расчета общего времени обучения НСМ:

$$t_\Sigma(\boldsymbol{net}_1) \approx 10 \bar{t}_v N_x + 10 k_1 \tau e^{-\varepsilon} N_x (N_x + N_y), \tag{2.21}$$

$$t_\Sigma(\boldsymbol{net}_2) \approx 10 \bar{t}_v N_x + 100 k_2 \tau e^{-\chi\varepsilon} N_x^2 (N_x + N_y), \tag{2.22}$$

где $t_\Sigma(\boldsymbol{net}_1)$– общее время обучения для НСМ из состава $\boldsymbol{net}_1$; $t_\Sigma(\boldsymbol{net}_2)$– общее время обучения для НСМ из состава $\boldsymbol{net}_2$.

После тривиальных упрощений (2.21, 2.22) получим:

$$t_\Sigma(\boldsymbol{net}_1) = 10 N_x \left(\bar{t}_v + k_1 \tau e^{-\varepsilon} (N_x + N_y)\right), \tag{2.23}$$

$$t_\Sigma(\boldsymbol{net}_2) = 10 N_x \left(\bar{t}_v + 10 k_2 \tau e^{-\chi\varepsilon} N_x (N_x + N_y)\right). \tag{2.24}$$

Результаты теоретических работ, посвященных НС [17], позволяют утверждать, что $k_1 \approx 0{,}1$, $k_2 \approx 0{,}001$, $\chi \approx 1$. Также на основании [33] путем экспертного оценивания определено, что максимально допустимая ошибка обучения НСМ, которые используются в системе распознавания кибератак на сетевые РИС, $\varepsilon \approx 0{,}05$. Подставив указанные величины в (2.23, 2.24), получим:

$$t_\Sigma(\boldsymbol{net}_1) \approx 10 N_x \left(\bar{t}_v + 0{,}1 \tau e^{-0{,}05} (N_x + N_y)\right), \tag{2.25}$$

$$t_\Sigma(\boldsymbol{net}_2) \approx 10 N_x \left(\bar{t}_v + 0{,}01 \tau e^{-0{,}05} N_x (N_x + N_y)\right). \tag{2.26}$$

Учитывая, что $e^{-0{,}05} = 0{,}951229 \approx 1$, получим:

$$t_\Sigma(\boldsymbol{net}_1) \approx 10 N_x \left(\bar{t}_v + 0{,}1 \tau (N_x + N_y)\right), \tag{2.27}$$

$$t_\Sigma(\boldsymbol{net}_2) \approx 10 N_x \left(\bar{t}_v + 0{,}01 \tau N_x (N_x + N_y)\right). \tag{2.28}$$

Результаты первого раздела и данные [58] указывают на то, что в первом приближении при распознавании сетевых кибератак множество входных параметров НСМ может соответствовать множеству параметров сетевого трафика.

Также данные [83, 84] свидетельствуют о том, что количество элементов каждого из указанных множеств не превышает 50. Поскольку на вход НСМ, кроме непосредственно зарегистрированных параметров могут подаваться и другие параметры, в первом приближении примем, что количество входных параметров НСМ $N_x = 50 \ldots 100$. Так как выражения (2.25, 2.26) имеют приблизительный характер и ориентируясь на определение максимального срока обучения принимаем: $N_x + N_y \approx N_x \approx 100$. Эта предпосылка позволяет модифицировать (2.29, 2.30) следующим образом:

$$t_{\Sigma}(\boldsymbol{net}_1) \approx 1000(\bar{t}_v + 10\tau), \tag{2.29}$$

$$\boldsymbol{t}_{\Sigma}(\boldsymbol{net}_2) \approx 1000(\bar{\boldsymbol{t}}_v + 100\boldsymbol{\tau}). \tag{2.30}$$

Так как $t_{\Sigma}(\boldsymbol{net}_2) > t_{\Sigma}(\boldsymbol{net}_1)$, то с учетом выражений (2.29, 2.30) правило определения допустимости использования вида НСМ для распознавания кибератак на сетевые РИС (2.6) можно детализировать следующим образом:

$$If\ 1000(\bar{t}_v + 10\tau) \le t_d \rightarrow \boldsymbol{net_1} \in \boldsymbol{Net}, \tag{2.31}$$

$$If\ 1000(\bar{t}_v + \tau) \le t_d \rightarrow \boldsymbol{Net} = \{\boldsymbol{net_1}, \boldsymbol{net_2}\}. \tag{2.32}$$

Условие (2.31) определяет допустимость использования для распознавания кибератак на сетевые РИС НСМ на основе АНС, ТК, СНС, РБФ, PNN, сетей адаптивной резонансной теории. Условие (2.32) дополняет допустимое множество НСМ моделями на основе МСП и ГНС.

Рассмотрим приблизительную оценку допустимого срока создания НСМ распознавания кибератак на сетевые РИС. Учтем, что разработка указанной НСМ является лишь одной из составляющих общего процесса создания СЗИ. Поэтому

$$t_d = k_{nsm} \times t_{\max}, \tag{2.33}$$

где t_{max} – максимально допустимый срок разработки СЗИ; k_{nsm}– коэффициент пропорциональности между t_d и t_{max}.

В соответствии с [97], при оценочных расчетах можно принять, что длительность разработки НСМ занимает примерно четверть всего срока

создания СЗИ. Поэтому:

$$k_{nsm} \approx 0{,}25\,. \tag{2.34}$$

Приняв во внимание годичный срок разработки СЗИ [91], получено

$$t_d \approx 0{,}25 \times 1 год = 7{,}5 \times 10^6\, c\,. \tag{2.35}$$

Использовав (2.37), выражения (2.33, 2.34) модифицированы так:

$$If\ 1000\left(\bar{t}_v + 10\tau\right) \le 7{,}5 \times 10^6 \rightarrow \boldsymbol{net_1} \in \boldsymbol{Net}, \tag{2.36}$$

$$If\ 1000\left(\bar{t}_v + \tau\right) \le 7{,}5 \times 10^6 \rightarrow \boldsymbol{Net} = \left\{\boldsymbol{net_1}, \boldsymbol{net_2}\right\}. \tag{2.37}$$

Выражения (2,31, 2.32, 2.36, 2.37) являются правилами для определения допустимых видов НСМ, предназначенных для распознавания кибератак на сетевые РИС. Применение этих правил ко множеству доступных НСМ позволяет перейти к определению множества эффективных видов НСМ. Для этого сформировано соответствующее множество критериев эффективности. Разработка базировалась на предложенном принципе оценивания эффективности НСМ, предназначенных для распознавания кибератак на сетевые РИС. Также использованы результаты [33], в которых определен перечень критериев эффективности оценки параметров безопасности ИС. В процессе разработки определено, что с позиций диссертационного исследования, требования к НСМ характеризуют их обучаемость, вычислительные возможности и техническую реализацию. В свою очередь требования к обучаемости определяются возможностями:

– Использования примеров с разным количеством входных параметров. Это требование существенно упрощает организацию процесса сбора и предварительной обработки реальных статистических данных.

– Использования обучающей выборки, объем которой меньше количества входных параметров, т. е. на выборке в ≤100 примеров. Выполнение этого требования позволяет распознавать новые виды кибератак, статистика которых непредставительна.

– Непропорционального представления в обучающей выборке распознаваемых классов.

– Применения учебных примеров, в которых отсутствует ожидаемый выходной сигнал.

– Обучения на зашумленных учебных примерах.

– Усваивать обученной НСМ новые учебные примеры без полного переобучения.

– Параллельного обучения. В этом случае НСМ может обучаться по частям.
– Стабильного обучения. В этом случае НСМ за приемлемый период обучения гарантированно обеспечивает достаточную ошибку обучения, которая в теории НС традиционно рассчитывается с помощью выражения:

$$\varepsilon = N_{true} / N_{\Sigma}, \tag{2.38}$$

где N_{true} – количество правильно распознанных учебных примеров; N_{Σ} – общее количество учебных примеров.
– Минимизации срока обучения, который в основном определяется количеством учебных итераций. Отметим, что длительность обучения НСМ из состава множества net1, возможно ценить с помощью выражения (2.29), а длительность обучения НСМ из состава net2 – с помощью выражения (2.30).
– Обеспечить высокий уровень автоматизации обучения, что при использовании качественной обучающей выборки зависит от количества эмпирически настраиваемых параметров НСМ.
– Возможность подачи в НСМ явных экспертных знаний.
– Требования к вычислительным («интеллектуальным») возможностям НСМ определяются.
– Отношением количества учебных примеров, которые может запомнить НСМ к количеству синаптических связей в этой модели.
– Ошибкой интерполяции данных, которая характеризует возможность правильного распознавания примеров, которые хотя и не вошли в обучающую выборку, но параметры которых находятся между пределами параметров учебных примеров.
– Ошибку экстраполяции данных, которая характеризует возможность правильного распознавания примеров, параметры которых лежат вне пределов параметров учебных примеров.
– Возможностью вербализации обученной НСМ, что подразумевает под собой обеспечение получения явных правил с помощью которых НС принимает решения.

Перечень критериев эффективности, отвечающих указанным требованиям, показан в приложении А. По аналогии с [98], принято, что значения предложенных критериев могут изменяться в пределах от 0 до 1. При этом для *i*-го вида НСМ значение *k*-го критерия равно 1, если соответствующее *k*-ое требование полностью обеспечивается в данном виде НСМ и равно 0, если не обеспечивается. Если *k*-ое требование обеспечивается частично, то величина $R_k \in]0,1[$ и может быть определена с помощью экспертного оценивания. Значения элементов множества критериев ($\boldsymbol{R}_a$) для апробированных видов НСМ показаны в приложении Б.

В соответствии с принципом оценивания эффективности вида НСМ для распознавания кибератак на сетевые РИС, количество критериев

эффективности K в выражении (2.8) равно 19. Также в первом приближении будем считать:

$$\mathbf{Net}_a = \mathbf{Net}_d = \{\text{МСП}, \text{ГНМ}, \text{ТК}, \text{PNN}, \text{РБФ}\}. \quad (2.39)$$

Таким образом, количество допустимых видов НСМ $I = 5$, что позволяет трансформировать выражение (2.8) для расчета функции эффективности i-го вида НСМ следующим образом:

$$V_i = \sum_{k=1}^{19} \alpha_k R_k(net_i), \quad net_i \in \{\text{МСП, ГНМ, ТК, PNN, РБФ}\}, \ i = 1,...5. \quad (2.40)$$

Отметим, что с помощью весового коэффициента α_k учитывается значимость k-го критерия эффективности для конкретной прикладной задачи распознавания кибератак на сетевые РИС. На практике множество значений весовых коэффициентов $(\boldsymbol{\alpha})$ можно получить с помощью экспертного оценивания. В результате можно утверждать, что правило формирования множества эффективных видов НСМ определяется выражением (2.41), а правило нахождения наиболее эффективного вида НСМ определяется выражением (2.42).

$$If\ V(net) \geq \Delta_V \wedge net \in \boldsymbol{Net}_d \rightarrow net \in \boldsymbol{Net}_e, \quad (2.41)$$

$$If\ \max_i = \{ V(net_1), \ldots V_I \}, net_i \in \boldsymbol{Net}_e \rightarrow net_i = net_e^{\max}, \quad (2.42)$$

где $V(net)$ – эффективность НСМ, которая рассчитывается с помощью выражения (2.40), Δ_V – минимально допустимая эффективность НСМ.

2.4 Модель формирования параметров учебных примеров

В общем случае процесс нейросетевого анализа k-го примера контролируемых на эксплуатации параметров КСС с целью распознавания сетевой кибератаки i-го вида можно отобразить выражением вида:

$$NNet(\boldsymbol{R}(k)) \rightarrow \boldsymbol{S}(Ka_i) \quad (2.43)$$

где $NNet$ – оператор нейросетевого анализа, $\boldsymbol{R}$ – множество контролируемых на эксплуатации параметров КСС, $\boldsymbol{R}(k)$ – значения слагаемых $\boldsymbol{R}$ для k-го примера, $\boldsymbol{S}(Ka_i)$ – сигнал НСМ о реализации кибератаки i-го вида.

При формировании примеров следует принимать во внимание необходимость предварительной обработки слагаемых $\boldsymbol{R}$ для их приведения к

виду, пригодному к использованию в НСМ. Методология такой обработки достаточно подробно сформирована в [81] и сводится к реализации различных процедур центрирования и нормализации. Результатом обработки является множество входных параметров НСМ $\boldsymbol{X}$. При этом для общего случая достаточно сложной задачей является определение перечня слагаемых $\boldsymbol{R}$. Однако, в соответствии с [49], для задачи распознавания сетевых кибератак номенклатура контролируемых на эксплуатации параметров КСС ограничена номенклатурой параметров сетевого трафика.

Также отметим, что $\boldsymbol{S}(Ka_i)$ – это множество чисел, характеризующее результат распознавания. В базовом случае $\boldsymbol{S}(Ka_i)$ может принимать только бинарные значения: 0 – кибератаки нет, 1 – кибератака есть. В более сложных случаях $\boldsymbol{S}(Ka_i)$ – это множество вещественных чисел, которые интерпретируются как вероятность или уверенность в реализации кибератаки. При использовании НСМ, обучаемых без учителя, в учебных примерах $\boldsymbol{S}(Ka_i)$ отсутствует, в противоположном случае $\boldsymbol{S}(Ka_i)$ нужно заранее определить в виде ожидаемого выходного сигнала $\boldsymbol{Y}(Ka_i)$.

Таким образом, модель формирования параметров *k*-го учебного примера, описывающего реализацию кибератаки *i*-го вида, можно записать в виде:

$$\begin{cases} \boldsymbol{R}(k)_{Ka_i} \to \boldsymbol{X}(k)_{Ka_i} \\ \boldsymbol{S}(Ka_i) \to \boldsymbol{Y}(Ka_i) \end{cases} \mapsto \boldsymbol{X}(k)_{Ka_i} \leftrightarrow \boldsymbol{Y}(Ka_i), \qquad (2.44)$$

где $\boldsymbol{R}(k)_{Ka_i} \to \boldsymbol{X}(k)_{Ka_i}$ – процедура адаптации контролируемых параметров КСС к НСМ, $\boldsymbol{S}(Ka_i) \to \boldsymbol{Y}(Ka_i)$ – процедура определения ожидаемого выходного сигнала НСМ, $\boldsymbol{X}(k)_{Ka_i} \leftrightarrow \boldsymbol{Y}(Ka_i)$ – процедура сопоставления входных и выходных параметров.

Рассмотрим процедуру $\boldsymbol{X}(k)_{Ka_i} \leftrightarrow \boldsymbol{Y}(Ka_i)$. Воспользуемся сформулированным принципом о необходимости учета в ожидаемом выходном сигнале близости указанных эталонов. В соответствии с возможными структурными решениями НСМ необходимо рассмотреть два случая формирования выходного сигнала [87]. В первом случае выходной сигнал НСМ реализуется с помощью одного нейрона в выходном слое:

$$N_y = 1, \qquad (2.45)$$

где N_y – количество нейронов в выходном слое.

Во втором случае количество нейронов в выходном слое равно количеству распознаваемых состояний защищенности сетевых РИС:

$$N_y = K_s, \qquad (2.46)$$

где K_s – количество распознаваемых состояний защищенности.

Детализируем первый случай. При анализе защищенности каждой из возможных кибератак, а также каждому из возможных безопасных состояний сетевого РИС ставится в соответствие некоторый диапазон величин выходного сигнала. В базовом случае можно предположить, что величины диапазонов для разных состояний защищенности разные [53]. Кроме того, для учебных примеров, которые отвечают эталонам состояний защищенности, выходной сигнал будет равен середине указанного диапазона. При использовании сигмоидальной функции активации нейронов выходного слоя выходной сигнал находится в пределах от 0 до 1:

$$y \in]0..1[, \tag{2.47}$$

где y – выходной сигнал НСМ.

При условии равномерного квантования диапазона возможных значений y ожидаемый выходной сигнал для эталона произвольного *i*-го состояния защищенности рассчитывается так:

$$y_{s_i} = \frac{1}{K_s} \times i - \frac{0{,}5}{K_s} = \frac{i - 0{,}5}{K_s}, \tag{2.48}$$

где i – номер состояний защищенности.

Схожесть состояний защищенности в выражении (2.47) возможно учесть только за счет того, что схожие состояния защищенности должны иметь близкие номера.

Детализируем второй случай. В учебном примере для эталона *i*-го состояния выходной сигнал соответствующего *i*-го нейрона равен 1. При этом порядок нумерации выходных нейронов может быть произвольным. Вместе с тем возникает необходимость определения ожидаемого выходного сигнала для всех других выходных нейронов, которые не отвечают данному эталону. Отметим, что нейроны, которые отвечают состояниям защищенности, близким к эталону, должны иметь схожие величины выходного сигнала. Так, базируясь на результатах [89], можно утверждать, что в примере для эталона кибератаки типа loadmodule величина выходного сигнала соответствующего нейрона должна меньше отличаться от величины выходного сигнала нейрона кибератаки типа rootkit, чем от величины выходного сигнала нейрона кибератаки типа «шторм запросов». То есть

$$\left| y_{[а]} - y_{[о]} \right| < \left| y_{[а]} - y_{[б]} \right|, \tag{2.49}$$

где $y_{[a]}, y_{[o]}, y_{[б]}$ – выходные сигналы нейронов, которые отвечают кибератакам loadmodule, rootkit, «шторм запросов».

По сути, определение ожидаемого выходного сигнала НСМ для второго случая является несколько усложненным вариантом первого случая, который сводится к расчету числовой оценки близости состояний защищенности. При этом известные аналитические методы такого расчета [63] отличаются большой сложностью. В то же время, анализ и распознавание кибератак на сетевые РИС – это задачи, которые достаточно эффективно решаются экспертами в области защиты информации [104]. Поэтому целесообразно определять оценку степени схожести параметров кибератак и параметров безопасных состояний на основе экспертных данных. Базируясь на [51, 52], предлагается использовать статистические методы обработки экспертных данных. В этом случае полученные от экспертов количественные данные обрабатываются с целью оценки коллективного мнения экспертной группы, оценки согласованности мнений экспертов и оценки их компетентности. Для определения оценок используются статистические методы точечного и интервального оценивания. Для этого рекомендуется, чтобы количество экспертов было не менее 10.

Процедура экспертного оценивания степени близости состояний защищенности. Пусть в результате опроса экспертной группы, которая состоит из *m* участников, получены следующие данные:

$$\begin{matrix} x_{1,1}, & \dots & x_{n,1} & \dots & x_{N,1} \\ \dots & \dots & \dots & \dots & \dots \\ x_{1,m} & \dots & x_{n,m} & \dots & x_{N,m}, \\ \dots & \dots & \dots & \dots & \dots \\ x_{1,M} & \dots & x_{n,M} & \dots & x_{N,M} \end{matrix} \qquad (2.50)$$

где $x_{n,m}$ – оценка степени схожести *n*-го объекта (состояния защищенности) *m*-ым экспертом; *N* – количество объектов (состояний защищенности); *M* – количество экспертов.

Средняя коллективная оценка *n*-го состояния защищенности рассчитывается с помощью формулы:

$$x_n = \frac{1}{M} \times \sum_{m=1}^{M} x_{n,m}, \qquad (2.51)$$

где $x_{n,m}$ – оценка степени схожести *n*-го состояния защищенности *m*-ым экспертом, $n = 1 \dots N$.

Дисперсия средней коллективной оценки определяется так:

$$\sigma^2 = (M-1)^{-1} \times \sum_{m=1}^{M} (x_{n,m} - x_n)^2 . \quad (2.52)$$

Для определения статистической значимости полученных результатов необходимо указать доверительный интервал, в который оцениваемая величина попадает с заданной доверительной вероятностью *P*.

Задавшись вероятностью ошибки P_n (уровнем значимости) можно определить интервал, в который оцениваемая величина попадает с вероятностью $(1 - P_n)$:

$$I_{x_n} = (x_n - \varepsilon_{pn}, x_n + \varepsilon_{pn}). \quad (2.53)$$

Величина ε_{pn} определяет границы доверительного интервала и рассчитывается так:

$$\varepsilon_{pn} = t_p \times \sigma_n / \sqrt{M} , \quad (2.54)$$

где t_p – коэффициент.

Считается, что оцениваемая величина имеет нормальное распределение с центром x_i и дисперсией σ. Коэффициент t_p имеет распределение Стьюдента с $(N-1)$ степенями свободы и определяется с помощью данных, которые для отдельных значений доверительной вероятности *P* показаны в таблице 2.1.

Степень согласованности экспертных мнений определяется с помощью коэффициента вариации γ_n, который рассчитывается по формуле:

$$\gamma_n = \sigma / x_n . \quad (2.55)$$

Таблица 2.1 - Значения коэффициента t_p

P	0,8	0,85	0,9	0,95
$\boldsymbol{t_p}$	1,282	1,439	1,643	1,960

Рассчитанный с помощью выражения (2.92) коэффициент вариации γ_n определяет относительную величину диапазона изменения оценок экспертов относительно среднего значения коллективной оценки x_n. Считается, что согласованность экспертных мнений удовлетворительна, если все $\gamma_n < 0{,}3$, и минимально достаточная, если все $\gamma_n < 0{,}2$. В противоположном случае процедуру экспертного оценивания следует повторить, с учетом результатов

[38]. Оценка компетентности экспертов может определятся по двум коэффициентам: объективному коэффициенту компетентности и коэффициенту относительной самооценки эксперта [76].

В данном разделе решалась научная задача развития методологической базы нейросетевого распознавания кибератак на сетевые ресурсы информационных систем. Основные результаты раздела следующие:

– Получила дальнейшее развитие концептуальная модель, которая за счет конкретизации параметров оценивания и факторов, влияющих на эффективность процесса нейросетевого распознавания кибератак, позволила детализировать направления дальнейших исследований.

– Получили дальнейшее развитие принципы использования нейронных сетей для распознавания сетевых кибератак, которые за счет учета степени обеспечения нейросетевой моделью характеристик поставленной задачи, соотнесения ожидаемых выходных сигналов нейросетевой модели со схожестью кибератак между собой и использования продукционных правил при формировании учебных примеров, обеспечивают возможность повышения эффективности нейросетевых моделей.

– Получила дальнейшее развитие модель правил определения эффективных видов нейросетевых моделей, в которой за счет реализации предложенных принципов обеспечивается возможность формализации определения эффективных видов нейросетевых моделей, что позволит повысить их точность и оперативность создания.

– Впервые разработана модель формирования учебных примеров, которая за счет использования предложенного принципа определения ожидаемого выходного сигнала обеспечивает возможность повышения точности и уменьшения срока обучения нейросетевых моделей.

3 РАЗРАБОТКА НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ И МЕТОДОВ РАСПОЗНАВАНИЯ КИБЕРАТАК

3.1 Нейросетевая модель противодействия сетевым кибератакам с помощью экспертных знаний

Базируясь на предложенном принципе использования экспертных знаний для формирования обучающей выборки определено, что в соответствии с [6], в первом приближении в качестве идентифицирующих параметров кибератак, которые применяются в продукционных правилах вида (2.12), возможно использовать параметры сетевого трафика [13]. За счет данного предположения выражение (2.11, 2.12) модифицируется следующим образом:

$$If\ x_1 = X_1 \wedge ... \wedge x_K = X_K \rightarrow Y_l, l = 1..L, \tag{3.1}$$

$$If\ x_1 \in \left[X_1^{\min}, X_1^{\max}\right]_l \wedge ... \wedge x_K \in \left[X_K^{\min}, X_K^{\max}\right]_l \rightarrow Y_l, l = 1..L, \tag{3.2}$$

где $x_1,...x_K$ – значения входных параметров НСМ, K – количество параметров сетевого трафика, $\left[X_1^{\min}, X_1^{\max}\right],..., \left[X_K^{\min}, X_K^{\max}\right]$ – заданные диапазоны $x_1,...x_K$, l – номер продукционного правила, L – количество продукционных правил.

В случае использования в качестве базы для формирования экспертных правил БД KDD-99 [105] количество входных параметров НСМ будет равно K=41, что равняется количеству полей указанной базы данных. Отметим, что значения этих полей соответствуют значениям сетевого трафика по протоколу TCP.

Также следует отметить, что в случае использования базы данных KDD-99 результатом продукционного правила может быть или вывод об одном из 22 видов кибератак или о нормальном сетевом запросе. В случае определения продукционных правил выражением (3.1) задача экспертов состоит в определении значений X_1, X_2,…X_K, а в случае определения продукционных правил выражением (3.2) задача экспертов состоит в определении границ диапазонов $\left[X_1^{\min}, X_1^{\max}\right],..., \left[X_K^{\min}, X_K^{\max}\right]$.

Результаты анализа [58], указывают на то, что наиболее полно приспособлены для обучения с помощью продукционных правил НСМ вида PNN и MPNN. Вместе с этим проведенный анализ указывает на то, что негативными особенностями PNN и MPNN является низкая информативность выходного сигнала. В этих НСМ значения выходного сигнала указывают только на то, вероятность какого события больше – реализации кибератаки или нормального. Это не позволяет использовать в СЗИ установленных пороговых значений вероятности кибератаки/нормального состояния или порогового значения разницы между этими вероятностями. Таким образом, в СЗИ, блок распознавания которой функционирует на основе сети MPNN невозможно внедрить защитные правила вида:

$$If\ \Theta_{Ka} > \Delta_{Ka} \rightarrow Z \quad (3.3)$$
$$If\ \Theta_{Nm} < \Delta_{Nm} \rightarrow Z \quad (3.4)$$
$$If\ \left(\Theta_{Ka} - \Theta_{Nm}\right) > \Delta_{\Delta} \rightarrow Z \quad (3.5)$$
$$If\ \Omega_i > \Delta_i \rightarrow Z \quad (3.6)$$

где Θ_{Ka}, Θ_{Nm} – вероятность реализации кибератак, Θ_{Nm} – вероятность нормального состояния, Z – защитные мероприятия, Ω_i – вероятность кибератаки i-го вида, $\Delta_{Ka}, \Delta_{Nm}, \Delta_{\Delta}, \Delta_i$ – пороговые значения.

Кроме этого недостаточная информативность выходного сигнала НСМ не позволяет СЗИ гибко реагировать на различные комбинации параметров сетевого трафика, а также не позволяет регистрировать параметры состояний защищенности, вероятность которых отлична от максимально вероятного состояния. Возможность регистрации таких параметров могло бы позволить создать базу данных, предназначенную для уточнения правил распознавания известных видов кибератак и формирования правил распознавания неизвестных видов кибератак.

Для исправления указанных недостатков, базируясь на [33] определено, что выходная информация сети MPNN, предназначенной для распознавания кибератак на сетевые РИС, должна быть дополнена:

Вероятностями реализации известных видов кибератак.

Вероятностью реализации наиболее вероятного вида кибератаки.

Вероятностью реализации кибератаки.

Вероятностью нормального состояния.

Для этого модели PNN и MPNN должны быть дополнены соответствующими выходными связями, а также дополнительным слоем нейронов, предназначенным для расчета вероятностей известных видов кибератак.

Модифицированная структура PNN предназначенная для распознавания состояния сетевых РИС на основании продукционных правил вида (3.1) показана на рисунке 3.1, а структура MPNN адаптированная к использованию продукционных правил вида (3.2) показана на рисунке 3.2.

Структурно модифицированная PNN состоит из 5 нейронных слоев. Это входной слой (Ln_{in}), слой образов (Ln_o), первый слой суммирования (Ln_{s1}), второй слой суммирования (Ln_{s2}) и слой сравнения (Ln_{coll}).

Как и в классической сети PNN нейроны входного слоя только передают данные из внешней среды к нейронам слоя образов. Слой образов предназначен для запоминания параметров обучающих примеров для всех классов, которые должна распознавать НС. Соответственно количество нейронов слоя образов равно количеству обучающих примеров, которые запомнила НС. На рисунке 3.1 нейроны слоя образов условно разделены на группы которые соответствуют известным видам кибератак и нормальному состоянию сетевого РИС. Так например, нейроны обозначенные $1_{KaN}…L_{KaN}$ соответствуют учебным примерам *N*-го вида кибератак, а нейроны обозначенные $1_{Nm}…M_{Nm}$ соответствуют

учебным примерам нормального состояния. В общем случае, количество учебных примеров для каждого вида кибератак и для нормального состояния может быть разным.

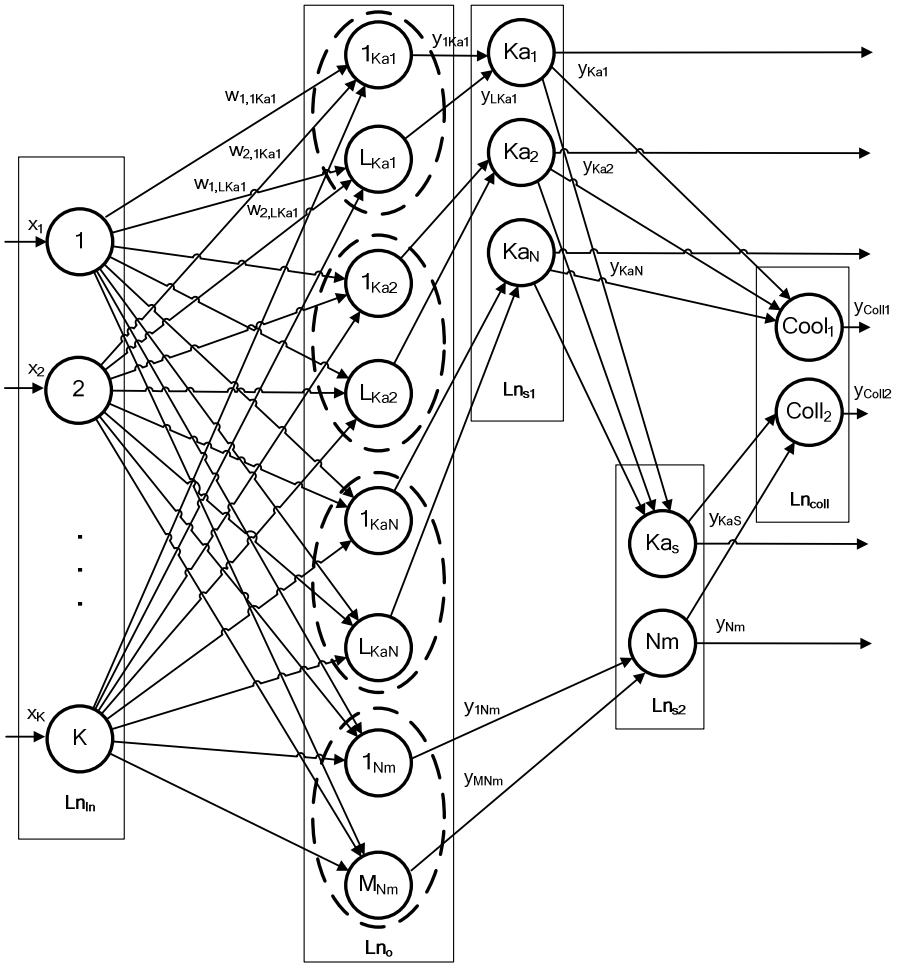

Рисунок 3.1 - Структура модифицированной PNN

Задачей первого слоя суммирования является расчет вероятности реализации каждого вида кибератак, известного для нейросети. Поэтому нейроны этого слоя ассоциируются с известными видами кибератак Ka_1, Ka_2, … Ka_N, а их количество равно количеству указанных видов кибератак (N). Второй слой суммирования предназначен для определения общей вероятности всех видов кибератак и вероятности нормального состояния сетевого РИС. Этот слой содержит два нейрона. Нейрон Ka_s используется для расчета общей вероятности кибератак, а нейрон Nm – для расчета вероятности нормального состояния. Задачей слоя сравнения является определение вида наиболее вероятной кибератаки, а также вида наиболее вероятного состояния защищенности. Вид наиболее вероятной кибератаки определяется нейроном $Coll_1$, а наиболее вероятное состояние защищенности (кибератака реализуется или нормальное состояние) определяется нейроном $Coll_2$.

Структура связей между слоями Ln_{in} и Ln_o полносвязная. При этом каждый нейрон входного слоя связан прямой связью с каждым нейроном слоя образов. Весовые коэффициенты этих связей равны компонентам обучающих примеров:

$$w_{1,l} = X_1, w_{2,l} = X_2, ... w_{K,l} = X_K \,, \tag{3.7}$$

где l – номер нейрона слоя образов, который соответствует номеру продукционного правила вида (3.1), K – количество входных параметров, $X_1, X_2, ... X_K$ – компоненты l -го продукционного правила вида.

Для примера на рисунке 3.1 показаны весовые коэффициенты связей между первым входным нейроном и $1_{Ka1}…L_{Ka1}$, $1_{Ka2}…L_{Ka2}$ нейронами слоя образов. Отметим, что весовые коэффициенты всех остальных связей в сети PNN равны 1. В нейронах слоя образов в качестве функции активации используется функция Гаусса. Выходной сигнал произвольного l-го из нейронов слоя образов, рассчитывается так:

$$y_l = \sum_{k=1}^{K} \exp\left(-(w_{k_l,l} - x_k)^2 / 2\sigma^2\right), \tag{3.8}$$

где $w_{k_l,l}$ – весовой коэффициент связи между k-ым нейроном входного слоя и l-ым нейроном слоя образов, x_k – величина k-го входного сигнала, K – количество входных параметров, σ – радиус функции Гаусса.

Отметим, что в соответствии с [17] радиус функции Гаусса $\sigma \in [0{,}3...{,}0{,}7]$. Структура связей между слоем образов и первым слоем суммирования имеет особенность. С нейроном первого слоя суммирования связываются только те нейроны слоя образов, которые соответствуют параметрам вида кибератаки с которой ассоциирован данный нейрон. Например, как показано на рисунке 3.1, нейроны $1_{Ka1}…L_{Ka1}$ связываются только с нейроном Ka_1. В нейронах слоев суммирования применяется линейная функция активации. Выходной сигнал для произвольного n-го нейрона первого слоя суммирования рассчитывается так:

$$y_{Ka_n} = \frac{\sum_{l=1}^{L_{Ka_1}} y_{l_{Ka_1}}}{L_{Ka_1}}, \tag{3.9}$$

где L_{Ka_1} – количество нейронов слоя образов, связанных с n-ым нейроном первого слоя суммирования, $y_{l_{Ka_1}}$ – выходной сигнал l-го нейрона слоя образов, связанного с n-ым нейроном первого слоя суммирования.

Величины выходных сигналов $y_{Ka_1}, ... y_{Ka_N}$ свидетельствуют о вероятности реализации каждого известного вида кибератак $Ka_1, ... Ka_N$. Эти сигналы составляют часть выходной информации сети PNN. Кроме этого они

передаются и нейрон $Cool_1$ слоя сравнения и в нейрон Ka_s второго слоя суммирования. Выходной сигнал этого нейрона рассчитывается так:

$$y_{Ka_s} = \frac{\sum_{n=1}^{N} y_{Ka_n}}{N}, \tag{3.10}$$

где N – количество известных видов кибератак.

Величина y_{Ka_s}, которая свидетельствует о суммарной вероятности реализации всех видов кибератак также является частью выходной информации сети PNN. Кроме этого y_{Ka_s} передается в нейрон $Cool_2$ слоя сравнения для дальнейшей обработки.

Выходной сигнал нейрона Nm второго слоя суммирования является оценкой вероятности нормального состояния сетевого РИС. Этот нейрон связан с только с теми нейронами слоя образов, которые ассоциированы с обучающими примерами нормального состояния. Величина выходного сигнала нейрона Nm рассчитывается так:

$$y_{Nm} = \frac{\sum_{m=1}^{M_{Nm}} y_{Nm_m}}{M_{Nm}}, \tag{3.11}$$

где M_{Nm} – количество нейронов слоя образов, которые соответствуют учебным примерам нормального состояния, y_{Nm_m} – выходной сигнал m-го нейрона слоя образов, соответствующего с m-му примеру нормального состояния.

Как и y_{Ka_s} сигнал y_{Nm} являющийся частью выходной информации сети передается в нейрон $Cool_2$. В нейронах $Cool_1$ и $Cool_2$ производится определение наиболее вероятного вида кибератаки и определение более вероятного состояния защищенности – кибератака реализуется/нормальное состояние. На рисунке 3.1 соответствующие выходные сигналы обозначены как y_{Coll1} и y_{Coll2}.

Основным отличием модифицированной сети MPNN от модифицированной сети PNN является наличие слоя фильтрации (на рисунке 3.2 обозначен – Ln_f), нейроны которого предназначены для фильтрации значений компонент входного сигнала в соответствии с продукционным правилом вида (3.2). При этом задачей некоторого $f_l^{x_k}$-го нейрона является фильтрация входного параметра x_k в соответствии с l–м продукционным правилом распознавания. Фильтрация сигнала осуществляется за счет использования функции активации вида:

$$\exists x_k \in \left[X_k^{\max}, X_k^{\min}\right]_l \rightarrow y_l^{f_{x_k}} = x_k, \exists x_k \notin \left[X_k^{\max}, X_k^{\min}\right]_l \rightarrow y_l^{f_{x_k}} = 0, \tag{3.12}$$

где x_k – значение k-го входного параметра, $y_l^{f_{x_k}}$ – выходной сигнал $f_l^{x_k}$-го нейрона СФ.

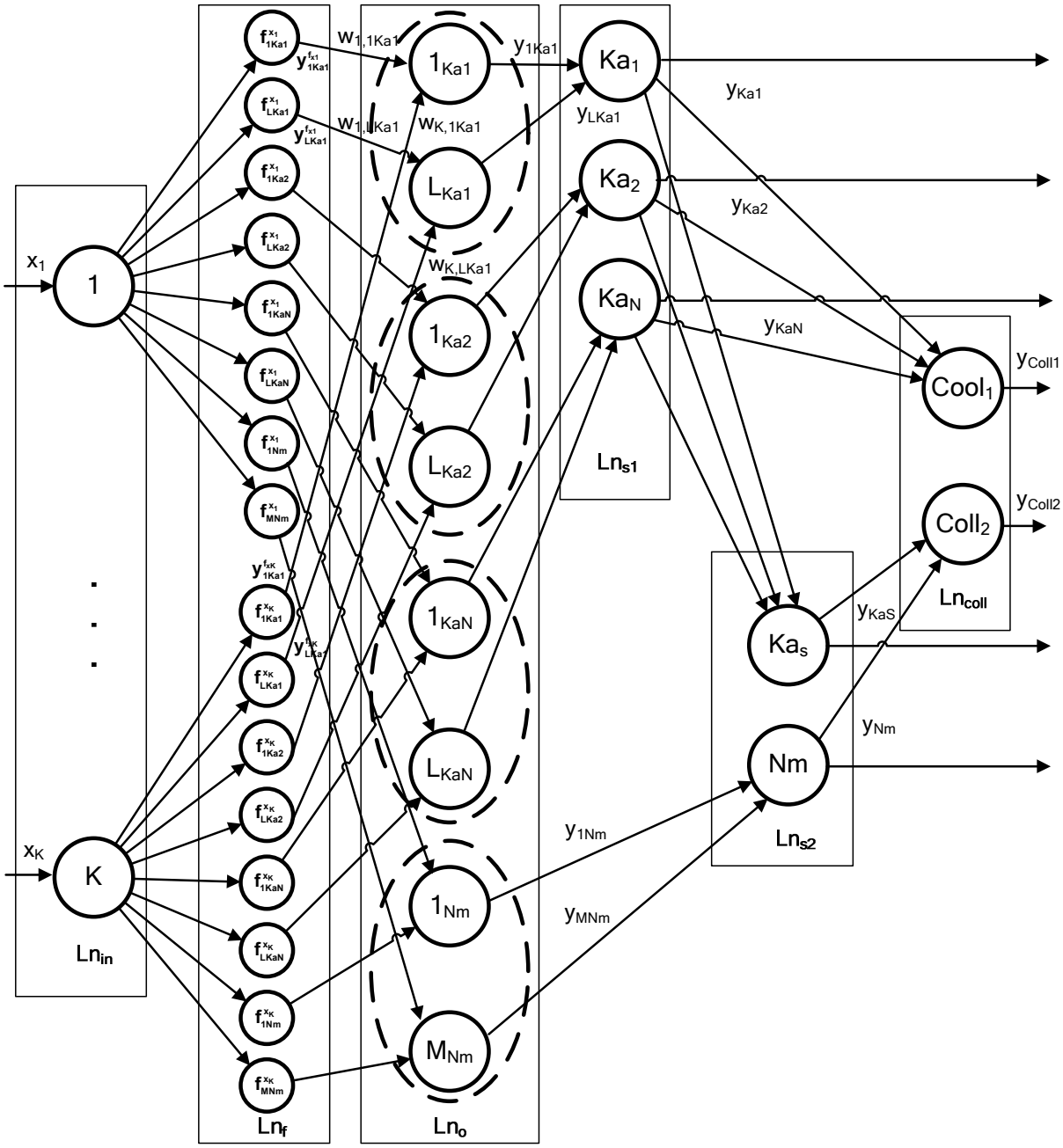

Рисунок 3.2 - Структура модифицированной MPNN

Весовые коэффициенты между нейронами слоя фильтрации и нейронами слоя образов рассчитываются так:

$$w_{f_l^{x_k},l} = \frac{X_k^{\max} - X_k^{\min}}{2}, \tag{3.13}$$

где l – номер нейрона слоя образов, который соответствует номеру продукционного правила вида (3.2), $f_l^{x_k}$ – нейрон, который фильтрует сигнал от x_k–го входного нейрона к l–му нейрону слоя образов, k – количество входных параметров, $X_k^{\max}, X_k^{\min}$ – компоненты l -го продукционного правила вида

Также отметим, что все весовые коэффициенты связей между входными нейронами и нейронами слоя фильтрации равны 1. Поэтому на рисунке 3.2 они не показаны.

Для внесения в сеть PNN l-го продукционного правила вида (3.1), определяющего наличие кибератаки вида $\mathrm{Ka_J}$ необходимо:

– Внести в слой образов новый нейрон l_{Kan}.

– Связать нейрон l_{Kan} с входными нейронами и в соответствии с выражением (3.7) установить для этого нейрона весовые коэффициенты входных связей.

– Внести в первый слой суммирования новый нейрон $\mathrm{Ka_J}$.

– Связать нейрон $\mathrm{Ka_J}$ с нейроном l_{Kan}.

– Связать нейрон $\mathrm{Ka_J}$ с нейроном $\mathrm{Ka_S}$, который рассчитывает суммарную вероятность реализации кибератак всех известных видов.

Для внесения в сеть PNN m-го продукционного правила вида (3.1), определяющего нормальное состояние сетевого РИС необходимо:

1) Внести в слой образов новый нейрон m_{Nm}.

2) Связать нейрон m_{Nm} с входными нейронами и использовав выражение (3.7), установить для него весовые коэффициенты входных связей равными соответствующим компонентам продукционного правила.

3) Связать нейрон m_{Nm} с нейроном Nm, который рассчитывает суммарную вероятность нормального состояния сетевого РИС.

В обучении сети MPNN есть некоторые отличия, связанные с более сложным характером продукционных правил, на которых она обучается. Для запоминания ею продукционного правила вида (3.2) этап два разделяется на три шага:

1) В слой фильтрации вносятся дополнительные нейроны, предназначенные для фильтрации значений входных параметров в соответствии с компонентами продукционного правила. Количество новых фильтрующих нейронов равняется количеству входных параметров НСМ.

2) Новые фильтрующие нейроны связываются с входными нейронами и с нейроном шара образов, который был внесен в сеть на предыдущем этапе.

3) Используя выражение (3.13) устанавливаются весовые коэффициенты связей между нейронами слоя образов и нейронами слоя фильтрации.

В режиме распознавания состояния защищенности сетевого РИС, PNN и MPNN функционируют так:

1) На вход НС подается вектор параметров $\{x\}_K$, который характеризует неизвестное состояние защищенности сетевого РИС.

2) Для MPNN рассчитываются выходные сигналы нейронов шара фильтрации. Для этого используется выражение (3.12). Для PNN этот этап не выполняется.

3) С использованием выражения (3.8), рассчитывается выходной сигнал каждого из нейронов шара образов Ka_1, Ka_2,…Ka_N.
4) Определяются вероятности каждого из известных видов кибератак. Для этого, с использованием выражения (3.9), рассчитывается выходной сигнал каждого из нейронов первого шара суммирования.
5) Определяется интегральная вероятность всех видов кибератак. Для этого с использованием выражения (3.10) рассчитывается выходной сигнал нейрона Ka_s принадлежащего второму шару суммирования.
6) Определяется интегральная вероятность нормального состояния сетевого РИС. Для этого с использованием выражения (3.11) рассчитывается выходной сигнал нейрона Nm, принадлежащего второму шару суммирования.
7) Определяется наиболее вероятная кибератака. Для этого с помощью нейрона $Cool_1$, рассчитывается у какого нейрона первого слоя суммирования выходной сигнал имеет наибольшее значение. Выходной сигнал $y_{Coll1} = k$, где k – номер нейрона первого слоя суммирования с наибольшим выходным сигналом. Кибератака, ассоциированная с этим нейроном считается наиболее вероятной.
8) Определяется наиболее вероятное состояние защищенности. Для этого с помощью нейрона $Cool_2$, сравниваются величины выходных сигналов нейронов Ka_s и Nm. Если $y_{Ka_s} \geq y_{Nm}$, то $y_{Coll1} = Ka_s$ и считается, что сетевой РИС подвергается кибератаке. В этом случае $y_{Coll1} = k$. В противоположном случае $y_{Coll1} = Nm$ и считается, что состояние сетевого РИС нормальное.

3.2 Модель глубокой нейронной сети

В соответствии с [106] при разработке НСМ типа ГНС в качестве базовой архитектуры использован трехслойный персептрон, для предобучения которого применен разреженный автокодировщик.

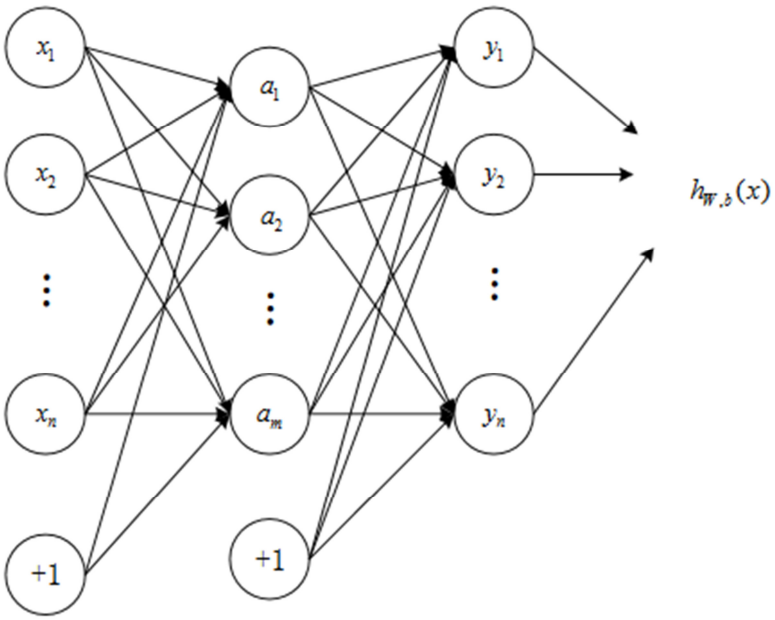

Рисунок 3.3 - Архитектура автокодировщика

Структура автокодировщика показана на рисунке 3.3. Отметим, что на рисунке 1 выходные сигналы входных нейронов помечены меткой «*x*», скрытых нейронов – меткой «*a*», выходных – «*y*», а блоков смещения – меткой «+1».

Входными данными автокодировщика является неразмеченная обучающая выборка $x = (x_1, x_{2,\ldots,} x_i)$. В скрытых и выходных нейронах используется сигмоидальная функция активации:

$$f(z_k) = \frac{1}{1 + e^{-z_k}}. \tag{3.14}$$

где z_k – входной сигнал *k*-го нейрона в скрытом или выходном слое.

В свою очередь

$$z_k = \sum_{i=1}^{n} \left(W_{i,k} x_{i,k} + x_0 b_k \right), \tag{3.15}$$

где $W_{i,k}$ – вес связи от *i*-го нейрона предыдущего слоя к *k*-му нейрону в скрытом или выходном слое, $x_{i,k}$ –сигнал от *i*-го нейрона предыдущего слоя к *k*-му нейрону, $x_0 = 1$ – вес связи нейрона с самим собой, b_k – смещение *k*-го нейрона.

Выход автокодировщика с количеством нейронных слоев *l* равен

$$h_{W,b}(x) = a^{(l)}, \tag{3.16}$$

где *W* – массив весовых коэффициентов, *b* – массив смещений, $a^{(l)}$ – массив выходных значений нейронов в слое *l*

Применительно к рисунку 3.3.

$$a^{(l)} = y, \tag{3.17}$$

где y – массив выходных значений нейронов последнего (*l*-го) слоя.

Особенностью автокодировщика является применение обучения без учителя при использовании алгоритма обратного распространения ошибки. Для этого целевая функция обучения автокодировщика определяется так:

$$h_{W,b}(x) \approx x. \tag{3.18}$$

Использование целевой функции вида (3.18) предполагает равенство выходного сигнала автокодировщика входному сигналу. Таким образом, обучение классического автокодировщика сводится к тому, чтобы с помощью алгоритма обратного распространения ошибки найти такие значения весовых коэффициентов, при которых выходной сигнал будет равен входному [106, 107]. При этом учебные примеры могут быть немаркированными, то есть не содержать ожидаемый входной сигнал.

Поиск оптимального значения весовых коэффициентов производится с помощью градиентного спуска путем минимизации функции потерь:

$$J(W,b)=\left[\frac{1}{m}\sum_{i=1}^{m}\left(0{,}5\left\|h_{W,b}(x^{(i)})-y^{(i)}\right\|^2\right)\right]+0{,}5\lambda\sum_{l=1}^{m}\sum_{i=1}^{s_{l-1}}\sum_{j=1}^{s_l}\left(W_{j,i}^{(l-1)}\right)^2, \qquad (3.19)$$

где m – количество скрытых слоев, s^l – количество нейронов в слое l, $W_{j,i}^{(l-1)}$ – вес связи между нейроном i в слое l и нейроном j в слое *(l-1)*.

Первая часть функционала – усредненная квадратичная ошибка по всем обучающим примерам, вторая часть – регуляризация (или контроль угасания весов), которая контролирует порядок весов и не дает переобучиться. Параметр λ, контролирующий угасание весов, регулирует относительную важность двух частей функционала.

Обучение производится до тех пор пока:

$$J(W,b)<\theta, \qquad (3.20)$$

где θ – заранее определенный коэффициент (порог).

Относительно классического варианта особенностью разреженного автокодировщика является ограничение количества одновременно активных нейронов в промежуточных слоях. Считается, что за счет этого разреженный автокодировщик автоматически обучается выделять из входных данных общие признаки, которые отображаются в значениях весовых коэффициентов. Для этого в функцию потерь вводится дополнительная компонента:

$$P=\sum_{j=1}^{h}\left(p\log\frac{p}{\hat{p}_j}+(1-p)\log\frac{(1-p)}{(1-\hat{p}_j)}\right), \qquad (3.21)$$

где $\hat{p}_j$ – среднее значение функции активации нейрона j по всем учебным примерам, $p\approx 0{,}05$ – параметр разреженности.

Отметим, что нейрон считается активным, если его выходной сигнал близок 1, а неактивным – близок к 0.

С учетом (3.20) оптимизируемая функция потерь разреженного автокодировщика имеет вид:

$$J_s(W,b) = J(W,b) + \beta P, \qquad (3.22)$$

где β – заданный коэффициент (в первом приближении $\beta \approx 3$).

Предобучение глубокой нейронной сети у которой количество нейронных слоев равно *m* реализуется так:

1) Случайным образом инициализируются весовые коэффициенты всех синаптических связей.
2) Исходя из необходимой точности обучения устанавливается значение коэффициента θ.
3) Устанавливается номер обучаемого слоя $l = 2$ (входной слой имеет номер 1).
4) К *l*-му слою нейронов подключается новый дополнительный слой.
5) На вход *l*-го слоя подается множество обучающих примеров.
6) С помощью (3.13-3.22) рассчитывается значение матрицы весовых коэффициентов связей *l*-го слоя нейронов.
7) Подключенный на 4 этапе слой нейронов удаляется.
8) Если $l < m$, то $l = l + 1$ и осуществляется переход на 5 этап. В противном случае предобучение заканчивается.

После этапа предобучения два последних слоя глубокой нейронной сети обучаются на маркированных данных.

3.3 Метод создания обучающей выборки для нейросетевой модели противодействия кибератакам

Отправной точкой разработки метода создания обучающей выборки послужили предложенные во втором разделе принцип допустимости использования вида НСМ для распознавания кибератак на сетевые РИС, принцип определения ожидаемого выходного сигнала НСМ для портретов кибератак, принцип использования экспертных знаний для формирования обучающей выборки, и разработанная на их основе модель формирования параметров обучающих примеров.

Принято во внимание, что в некоторых случаях БД сетевых кибератак могут быть недоступными или малоинформативными. В этих случаях предусмотрено формировать множество обучающих примеров на основе экспертной оценки параметров сетевого трафика. В этом случае задание экспертов будет состоять в определении, какие параметры сетевого трафика будут отвечать тому или иному состоянию безопасности сетевого РИС. В первом приближении предусмотрено, что выходной сигнал НСМ будет реализован с помощью одного выходного нейрона. Также предусмотрено случай, когда для формирования обучающей выборки будут доступны только немаркированные данные, соответствующие учебным примерам.

В общем случае преобразование информации, реализуемое предложенным методом, возможно представить с помощью следующих выражений:

$$\left\langle \boldsymbol{\Phi}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, t_d, N_x, N_y, \boldsymbol{Z}_{ов}, k_{нсс}, W_{\text{п}} \right\rangle \rightarrow \left\langle \boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{Z}_3, \boldsymbol{Z}_4 \right\rangle, \tag{3.23}$$

$$\boldsymbol{Z}_1 = \left\{ z_1^{(\phi_k)} \right\}_{K_\phi} = \left\{ \left(\boldsymbol{x}^{(\phi_k)}, y^{(\phi_k)} \right) \right\}_{K\phi}, \tag{3.24}$$

$$\boldsymbol{Z}_2 = \left\{ z_2^{(\phi_k)} \right\}_{K\phi} = \left\{ \boldsymbol{r}^{(\phi_k)} \right\}_{K_\phi}, \tag{3.25}$$

$$\boldsymbol{Z}_3 = \left\{ z_3^{(\varphi_l)} \right\}_{L_\phi} = \left\{ \boldsymbol{x}^{(\varphi_l)} \right\}_{L_\phi}, \tag{3.26}$$

$$\boldsymbol{Z}_4 = \left\{ z_4^{(\varphi_l)} \right\}_{L_\phi} = \left\{ \left\{ \left(\boldsymbol{x}^{(\phi_k)}, y^{(\phi_k)} \right) \right\}_{0,2K_\phi}, \left\{ \boldsymbol{x}^{(\varphi_l)} \right\}_{0,8L_\phi} \right\}, \tag{3.27}$$

где $\boldsymbol{\Phi}$ – множество (перечень) распознаваемых сетевых кибератак, $\boldsymbol{\Omega}_1$ – множество портретов сетевых кибератак, полученных из БД, $\boldsymbol{\Omega}_2$ – множество примеров параметров сетевого трафика, каждый из которых соответствует отдельному распознаваемому состоянию безопасности (сетевой кибератаки), $\boldsymbol{D}_1$ – множество экспертных данных, которые касаются соотношения ожидаемого выходного сигнала НСМ с эталоном кибератаки, $\boldsymbol{D}_2$ – множество экспертных данных, касающихся продукционных правил распознавания кибератак, $\boldsymbol{D}_3$ – множество экспертных данных, касающихся распознавания кибератак на основе анализа параметров сетевых запросов, t_d – допустимый срок формирования обучающей выборки, $\mathbf{Z_1}$ – обучающая выборка, примеры которой содержат ожидаемый выходной сигнал, $\mathbf{Z_2}$ – обучающая выборка с примерами в виде продукционных правил, $\mathbf{Z_3}$ – обучающая выборка, примеры которой не содержат ожидаемого выходного сигнала, $\mathbf{Z_4}$ – обучающая выборка, в которой большая часть примеров (≈80%) не содержат ожидаемого выходного сигнала, $\boldsymbol{Z}_{ов}$ – множество ограничений на формирование обучающей выборки, W_n – вычислительная мощность используемого АПО, $k_{нсс}$ – коэффициент использования НСР мощностей АПО, $z_1^{(\phi_k)}$, $z_2^{(\phi_k)}$ – учебные примеры вида $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$ для кибератаки *k*-го вида, $z_3^{(\varphi_l)}$ – учебные примеры вида $\boldsymbol{Z}_3$ для примера сетевого трафика *l*-го вида, $\boldsymbol{x}^{(\phi_k)}$ – множество входных параметров для кибератаки *k*-го вида, $y^{(\phi_k)}$ – ожидаемый выходной сигнал НСМ для кибератаки *k*-го вида, $\boldsymbol{r}^{(\phi_k)}$ – множество продукционных правил вида (3.1, 3.2) для

кибератаки k-го вида, K_ϕ – количество кибератак, которые должны быть распознаны, $\boldsymbol{x}^{(\varphi_l)}$ – входные параметры для примера сетевого трафика l-го вида, L_φ – количество примеров сетевых запросов.

Отметим, что основным источником данных для формирования обучающей выборки НСМ являются $\boldsymbol{\Omega}_1$ и $\boldsymbol{\Omega}_2$, представляющие собой примеры сетевого трафика, каждый из которых соответствует отдельному сетевому запросу, соответственно стеку протоколов TCP/IP. Разница между $\boldsymbol{\Omega}_1$ и $\boldsymbol{\Omega}_2$ состоит в том, что каждому сетевому запросу (примеру), который входит в множество $\boldsymbol{\Omega}_1$, поставлен в соответствие определенный вид состояния защищенности (вид кибератаки), а в $\boldsymbol{\Omega}_2$ такое соответствие отсутствует. $\boldsymbol{\Phi}$ представляет собой упорядоченное множество кибератак, которые должны быть распознаны. Упорядочивание множества может быть реализовано например по принципах используемых в БД KDD-99. При разработке модели формирования параметров обучающих примеров определено, что множество $\boldsymbol{x}^{(\varphi_l)}$ состоит из 40 компонентов. Также, при разработке НСМ распознавания сетевых кибератак с помощью экспертных знаний показано, что количество параметров в продукционных праивлах, которые входят в состав $\boldsymbol{r}^{(\phi_k)}$, равно 7.

Анализируя (3.1-3.4) определено, что для их реализации необходимо шест этапов: расчет допустимого объема обучающей выборки, определение ожидаемого выходного сигнала НСМ для каждого из эталонов сетевых кибератак, формирование $\boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{Z}_3, \boldsymbol{Z}_4$. Также целесообразно использовать этап определения возможности формирования обучающей выборки в виде множеств $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2, \boldsymbol{Z}_3$ или $\boldsymbol{Z}_4$.

Отметим, что наличие обучающей выборки в виде $\boldsymbol{Z}_1$ предоставляет возможность использования более мощных видов НСМ, однако ее формирование является более сложным и трудоемким. Поэтому принято считать:

$$If\ \boldsymbol{Z}_1 \in \boldsymbol{Z} \rightarrow \boldsymbol{Z}_2, \boldsymbol{Z}_3, \boldsymbol{Z}_4 \notin \boldsymbol{Z}. \tag{3.28}$$

При этом формирование обучающей выборки вида $\boldsymbol{Z}_2$ может быть усложненным за счет необходимой процедуры экспертного оценивания, а использование НСМ, адаптированных к $\boldsymbol{Z}_2$ или $\boldsymbol{Z}_3$ может быть ограничено в связи с максимально допустимой ошибкой распознавания.

Кроме этого, исходя из общепринятой методологии разработки НСМ [17], сформулировано вывод о необходимости этапа масштабирования входных и выходных параметров и этапа проверки качества предварительной обработки обучающей выборки.

Структурная схема метода создания обучающей выборки показана на рисунке 3.4. Указанные на рисунке 3.4 этапы метода детализируются так.

Этап 1 – расчет объема обучающей выборки. На данном этапе рассчитывается минимально и максимально допустимое количество учебных примеров. Входными данными этапа являются – W_n, N_x та K_ϕ. Для расчета минимально допустимого количества учебных примеров используется выражение:

$$L_\Sigma^{min} = \sum_{k=1}^{K_\phi} L_{\phi_k}^{min}, \tag{3.29}$$

где L_Σ^{min} – общее минимально допустимое количество учебных примеров, $L_{\phi_k}^{min}$ – минимально допустимое количество учебных примеров для каждой *k*-го вида кибератаки.

Учитывая разработанную модель формирования параметров учебных примеров, ориентацию на использование БД KDD-99, NSL-KDD и [37], для обучающей выборки вида $\boldsymbol{Z}_1, \boldsymbol{Z}_3, \boldsymbol{Z}_4$ минимально допустимое количество обучающих примеров для каждого вида кибератаки равно:

$$L_{\phi_k}^{\min}(\boldsymbol{Z}_1) = 10 \times N_{\text{x}} = 10 \times 40 = 400, \tag{3.30}$$

$$L_{\phi_k}^{\min}(\boldsymbol{Z}_3) = 30 \times N_{\text{x}} = 30 \times 40 = 1200, \tag{3.31}$$

$$L_{\phi_k}^{\min}(\boldsymbol{Z}_4) = 0{,}2 \times L_{\phi_k}^{\min}(\boldsymbol{Z}_1) + 0{,}8 \times L_{\phi_k}^{\min}(\boldsymbol{Z}_3) = 0{,}2 \times 400 + 0{,}8 \times 1200 = 1040, \tag{3.32}$$

где N_x – количество входных параметров НСМ.

Соответственно разработанной НСМ вида PNN и результатов [6], для обучающей выборки вида $\boldsymbol{Z}_2$ минимально допустимое количество учебных примеров для каждого вида кибератаки возможно рассчитать так:

$$L_{\phi_k}^{min}(\boldsymbol{Z}_2) = 20 \times N_x = 20 \times 7 = 140. \tag{3.33}$$

Общее минимально допустимое количество учебных примеров для каждого вида выборки получим, определив количество распознаваемых атак и подставив (3.30-3.33) в (3.29).

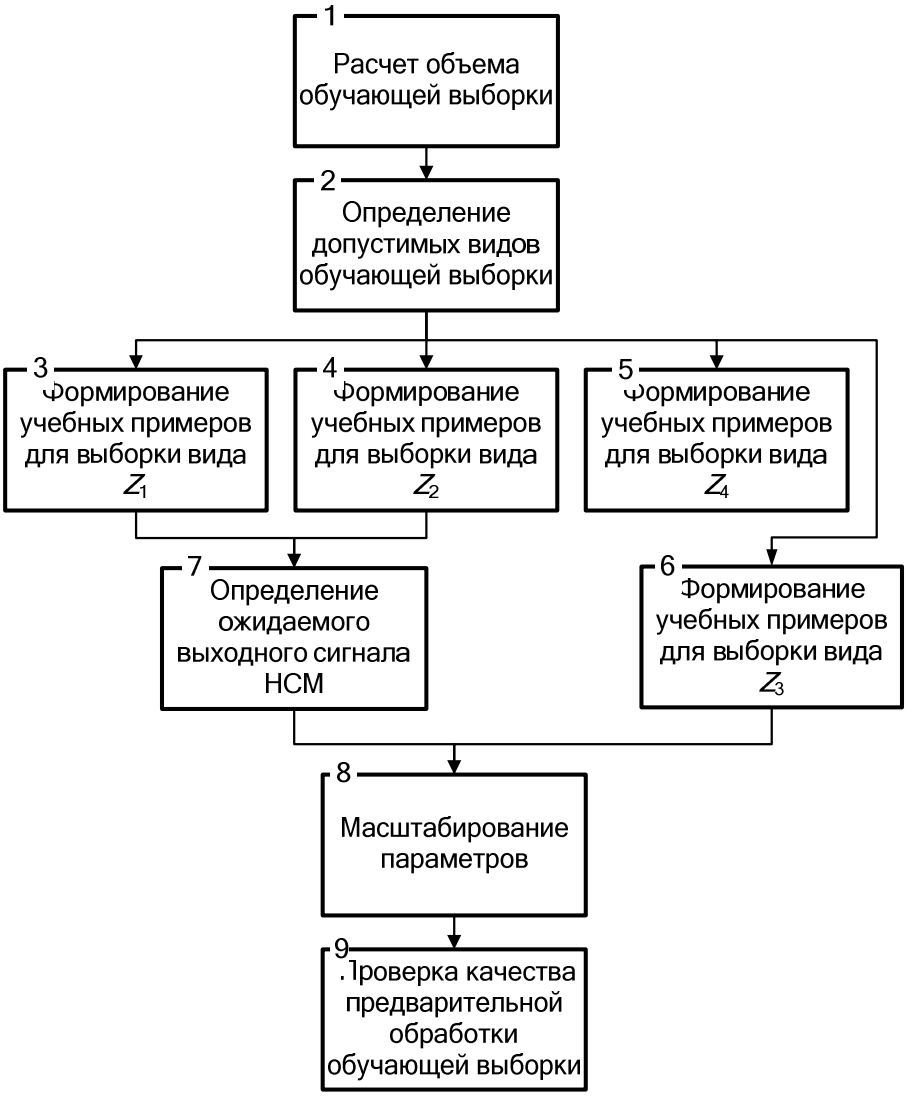

Рисунок 3.4 - Структурная схема метода создания обучающей выборки

Максимально допустимый объем обучающей выборки определяется исходя из возможности обеспечения бесперебойного обучения НСМ при ее реализации на общедоступном аппаратно-программном обеспечении. В первом приближении можно считать, что такая реализация дает возможность бесперебойного обучения на протяжении суток (8640 с) [9, с. 27-30].Использование указанной величины в выражениях (2.17, 2.18), которые получены при разработке моделей правил определения эффективных видов НСМ, позволяет записать следующие уравнения:

$$8640 \approx k_1 \tau e^{-\varepsilon} L_{\phi_k}^{max}\left(\boldsymbol{Z}_2, \boldsymbol{Z}_3\right)\left(N_x + N_y\right), \tag{3.34}$$

$$8640 \approx k_2 \tau e^{-\chi\varepsilon} L_{\phi_k}^{max}\left(\boldsymbol{Z}_1\right)^2\left(N_x + N_y\right)^2, \tag{3.35}$$

$$8640 \approx 0{,}2k_2\tau e^{-\chi\varepsilon}L_{\phi_k}^{\max}(\boldsymbol{Z}_4)^2(N_x+N_y)^2+0{,}8k_1\tau e^{-\chi\varepsilon}L_{\phi_k}^{\max}(\boldsymbol{Z}_4)(N_x+N_y), \qquad (3.36)$$

где $L_{\phi_k}^{max}(\boldsymbol{Z}_1)$ – максимально допустимое количество примеров вида $\boldsymbol{Z}_1$, $L_{\phi_k}^{\max}(\boldsymbol{Z}_2,\boldsymbol{Z}_3)$ – максимально допустимое количество примеров вида $\boldsymbol{Z}_2,\boldsymbol{Z}_3$, $L_{\phi_k}^{max}(\boldsymbol{Z}_2,\boldsymbol{Z}_3)$ – максимально допустимое количество примеров вида $\boldsymbol{Z}_4$, τ – продолжительность одной вычислительной операции процесса обучения.

После подстановки в (3.33-3.35) определенных в п. 2.3 величин $k_1 \approx 0{,}1$, $k_2 \approx 0{,}01$, $\chi \approx 1$, $\varepsilon \approx 0{,}01$ и тривиальных упрощений получим:

$$L_{\phi_k}^{\max}(\boldsymbol{Z}_2,\boldsymbol{Z}_3) \approx 86400/\tau(N_x+N_y), \qquad (3.37)$$

$$L_{\phi_k}^{\max}(\boldsymbol{Z}_1) \approx 864000/\tau(N_x+N_y)^2, \qquad (3.38)$$

$$L_{\phi_k}^{\max}(\boldsymbol{Z}_4) \approx 172800/\tau(N_x+N_y)^2 + 69120/\tau(N_x+N_y). \qquad (3.39)$$

Учитывая в (3.37-3.39), что для НСМ, которые отвечают выборкам $\boldsymbol{Z}_1$, $\boldsymbol{Z}_3$, $\boldsymbol{Z}_4$ $N_x+N_y=41$, для НСМ, отвечающим $\boldsymbol{Z}_2$, $N_x+N_y\approx10$ и ориентируясь на максимизацию срока обучения получим:

$$L_{\phi_k}^{\max}(\boldsymbol{Z}_1) \approx 504\tau^{-1}, \qquad (3.40)$$

$$L_{\phi_k}^{\max}(\boldsymbol{Z}_2) \approx 8640\tau^{-1}, \qquad (3.41)$$

$$L_{\phi_k}^{\max}(\boldsymbol{Z}_3) \approx 2107\tau^{-1}. \qquad (3.42)$$

$$L_{\phi_k}^{\max}(\boldsymbol{Z}_4) \approx 1789\tau^{-1}. \qquad (3.43)$$

Отметим, что $\tau = f(W_n)$, а ее величину следует определить экспериментальным путем, учитывая возможность распараллеливания процесса обучения. Таким образом, выходом первого этапа являются L_{Σ}^{min}, $L_{\phi}^{min}(\boldsymbol{Z}_1,\boldsymbol{Z}_3)$, $L_{\phi}^{min}(\boldsymbol{Z}_2)$, $\mathrm{L}_{\phi}^{\min}(\boldsymbol{Z}_4)$ $L_{\phi}^{max}(\boldsymbol{Z}_1)$, $L_{\phi}^{max}(\boldsymbol{Z}_2)$, $L_{\phi}^{max}(\boldsymbol{Z}_3)$, $\mathrm{L}_{\phi}^{\max}(\boldsymbol{Z}_4)$.

Этап 2 – определение допустимых видов обучающей выборки. Входными данными этапа являются K_ϕ, t_d, L_Σ^{min}, $L_\phi^{min}(\boldsymbol{Z}_1,\boldsymbol{Z}_3)$, $L_\phi^{min}(\boldsymbol{Z}_2)$, $\boldsymbol{\Phi}$, $\boldsymbol{\Omega}_1$, $\boldsymbol{\Omega}_2$, $\boldsymbol{D}_2$, $\boldsymbol{D}_3$. Выходом этапа является множество допустимых видов обучающей выборки $\boldsymbol{Z}$. Этап разделен на четыре шага, каждый из которых соотносится с проверкой вхождения в $\boldsymbol{Z}$ каждого из видов обучающей выборки $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, $\boldsymbol{Z}_3$ или $\boldsymbol{Z}_4$.

Шаг 1 – проверка допустимости $\boldsymbol{Z}_1$. Обучающая выборка вида $\boldsymbol{Z}_1$ считается допустимой при выполнении любого из двух условий.

Условие 1. Доступна БД, которая позволяет сформировать $\boldsymbol{z}_1^{(\phi_k)}$ с достаточным количество элементов. Для этого в указанных БД для каждого из видов кибератак должно быть достаточное количество соответствующих записей. Аналитическое выражение условия 1 имеет следующий вид:

$$If\ \boldsymbol{\Omega}_1 \notin \varnothing \wedge L_{\phi_k}^{\boldsymbol{\Omega}_1} \geq 400, k = 1,..\mathrm{K}_\phi \rightarrow \boldsymbol{Z}_1 \in \boldsymbol{Z} \qquad (3.44)$$

где $L_{\phi_k}^{\Omega_1}$ – количество элементов $\boldsymbol{\Omega}_1$, соответствующих k-му виду кибератак.

Условие 2. С помощью экспертного оценивания можно в приемлемый срок сформировать из доступных параметров сетевого запросов множество $\boldsymbol{z}_1^{(\phi_k)}$. В первом приближении это условие, возможно записать с помощью следующего выражения:

$$If\ \boldsymbol{\Omega}_2 \notin \varnothing \wedge \mathrm{E}_1\left(\boldsymbol{\Phi},\boldsymbol{\Omega}_2,\boldsymbol{D}_3\right) = \boldsymbol{z}_1^{(\phi_k)}, L_{\phi_k}^{z_1} \geq 400 \wedge t_{Z_1} \leq t_d \rightarrow \boldsymbol{Z}_1 \in \boldsymbol{Z}, \qquad (3.45)$$

где E_1 – процедура формирования $\boldsymbol{z}_1^{(\phi_k)}$ из $\boldsymbol{\Omega}_2$, $L_{\phi_k}^{z_1}$ – количество элементов $\boldsymbol{z}_1^{(\phi_k)}$, t_d – допустимый срок формирования обучающей выборки, длительность которого оценивается с помощью (2.33), $t_{\boldsymbol{Z}_1}$ – срок формирования обучающей выборки вида $\boldsymbol{Z}_1$, длительность которого оценивается с помощью (2.27).

Шаг 2 – проверка допустимости $\boldsymbol{Z}_2$. Учебная выборка вида $\boldsymbol{Z}_2$ считается допустимой при выполнении условия – с помощью экспертного оценивания в приемлемый срок для кожного k-го вида кибератаки можно разработать множество $\boldsymbol{r}^{(\phi_k)}$ с количеством элементов не меньшим чем 140:

$$If\ \mathrm{E}_2\left(\boldsymbol{\Phi},\boldsymbol{D}_2\right) = \boldsymbol{r}^{(\phi_k)}, L_{\phi_k}^{D_2} \geq 140 \wedge t_{Z_2} \leq \bar{\mathrm{t}}_\partial \rightarrow \boldsymbol{Z}_2 \in \boldsymbol{Z}, \qquad (3.46)$$

где E_2 – процедура формирования продукционных правил,

$t_{\boldsymbol{Z}_2}$ – срок формирования обучающей выборки вида виду $\boldsymbol{Z}_2$, который можно оценить с помощью (2.28).

В (3.46) учтено, что минимально допустимое количество элементов $\boldsymbol{r}^{(\phi_k)}$ определено с помощью выражения (3.24).

Шаг 3 – <u>проверка допустимости</u> $\boldsymbol{Z}_3$. Учебная выборка вида $\boldsymbol{Z}_3$ считается допустимой при выполнении условия:

$$If\ L_{\varphi} \geq k_{\varphi} L_{\Sigma}^{\min} \rightarrow \boldsymbol{Z}_3 \in \boldsymbol{Z}\ , \tag{3.47}$$

где k_{φ} – коэффициент ожидаемого распределения примеров видов кибератак в доступной $\boldsymbol{\Omega}_2$.

Базируясь на результатах [33] определено, что при использовании в качестве источника данных $\boldsymbol{\Omega}_2$ сетевых снифферов $k_{\varphi} = 10^3$. Это позволяет переписать (3.47) в виде:

$$If\ L_{\varphi} \geq 10^3 L_{\Sigma}^{\min} \rightarrow \boldsymbol{Z}_3 \in \boldsymbol{Z} \tag{3.48}$$

Шаг 4 – <u>проверка допустимости</u> $\boldsymbol{Z}_4$. Учебная выборка вида $\boldsymbol{Z}_4$ считается допустимой при выполнении условия – можно в приемлемый срок сформировать выборку в состав которой входит как минимум 20% примеров вида $\boldsymbol{z}_1^{(\phi_k)}$ и не белее80%, примеров вида $\boldsymbol{z}_3^{(\varphi_l)}$. Аналитической описание этого условия является симбиозом условий (3.44, 3.45, 3.47, 3.48):

$$\begin{array}{l}\left(If\ \left(\boldsymbol{\Omega}_2 \notin \varnothing \wedge \mathrm{E}_1\left(\boldsymbol{\Phi}, \boldsymbol{\Omega}_2, \boldsymbol{D}_3\right) = \boldsymbol{z}_1^{(\phi_k)}, L_{\phi_k}^{z_1} \geq 80 \wedge t_{Z_1} \leq t_d\right) \wedge If\left(L_{\varphi} \geq k_{\varphi} L_{\Sigma}^{\min}\right)\right) \\ \vee \left(If\ \left(\boldsymbol{\Omega}_1 \notin \varnothing \wedge L_{\phi_k}^{\boldsymbol{\Omega}_1} \geq 80\right) \wedge If\left(L_{\varphi} \geq 0{,}8 k_{\varphi} L_{\Sigma}^{\min}\right)\right) \rightarrow \boldsymbol{Z}_4 \in \boldsymbol{Z}\end{array}, \tag{3.49}$$

Этап 3 – формирование учебных примеров для выборки вида $\boldsymbol{Z}_1$. Базой этапа является разработанная модель формирования параметров учебных примеров. Учтено, что источником формирования $\boldsymbol{Z}_1$ могут быть или БД кибератак, или множества зафиксированных параметров сетевых запросов. Для этого случая предусмотрено использовать процедуру экспертного оценивания соответствия параметров сетевого запроса некоторому виду кибератак.

Входными данными этапа является K_{ϕ}, $\boldsymbol{\Phi}$, $L_{\phi}^{min}\left(\boldsymbol{Z}_1\right)$, $L_{\phi}^{max}\left(\boldsymbol{Z}_1\right)$, $\boldsymbol{\Omega}_1$, $\boldsymbol{\Omega}_2$, $\boldsymbol{D}_3$, t_d а выходом множество учебных примеров $\boldsymbol{Z}_{1,a} = \left\{\boldsymbol{z}_{1,a}^{(\phi_k)}\right\}_{K_{\phi}} = \left\{\left(\boldsymbol{x}_a^{(\phi_k)}, y_a^{(\phi_k)}\right)\right\}_{K\phi}$ параметры которых нуждаются в масштабировании. При этом ожидаемый выходной сигнал представлен в символьном виде. Этап разделен на три шага.

Шаг 1 – экспертное оценивание параметров сетевого запроса. Этот шаг выполняется только в случае использования в качестве источника данных множества зафиксированных параметров сетевых запросов. Результатом выполнения является определение вида состояния защищенности (кибератака определенного вида/безопасное состояние), соответствующее каждому из зафиксированных запросов. Для этого предлагается использовать процедуру экспертного оценивания, подобную процедуре экспертной оценки близости видов кибератак, которая используется в разработанной модели формирования параметров учебных примеров, а математический аппарат которой определен выражениями (2.50-2.56). Отличие указанных процедур между собой состоит только в том, что при оценивании отдельных сетевых запросов эксперт для каждого их них устанавливает вероятность принадлежности к каждому из распознаваемых видов кибератак, которые входят в множество $\boldsymbol{\Phi}$. Таким образом в выражении (2.50) величина $x_{n,m}$ представляет собой установленную экспертом вероятность того, что данный запрос является кибератакой *n*-го вида. Соответственно в выражении (2.51) величина x_n интерпретируется как средняя коллективная оценка того, что данный запрос является кибератакой *n*-го вида. Также в выражении (2.51) считается

$$If\ x_n \le \Delta_\phi \to x_n = 0\,, \tag{3.50}$$

где Δ_ϕ – эмпирический коэффициент (в первом приближении $\Delta_\phi = 0{,}1$).

Кроме того, предложенная процедура экспертного оценивания предусматривает сопоставления сетевого запроса с несколькими видами кибератак. Входные данные для экспертного оценивания как правило представляются в виде параметров сетевого трафика по стеку протоколов TCP/IP.

Шаг 2 – определение выходного сигнала в символьном виде. Шаг адаптирован к источнику учебных примеров. Если источником данных является $\boldsymbol{\Omega}_1$, то значением выходного параметра учебного примера является комбинация символов:

$$y^{(\phi_k)} = 1 * \phi_k\,. \tag{3.51}$$

где k – номер вида кибератаки в алфавите, ϕ_k – название k-го вида кибератаки.

Если источником данных является $\boldsymbol{\Omega}_2$ и в результате выполнения шага 1 определено, что есть ненулевая вероятность принадлежности сетевого запроса к нескольким видам кибератак, то данный элемент используется для формирования нескольких учебных примеров. Выходной сигнал для каждого из указанных примеров определяется следующим образом:

$$y^{(\phi_k)} = y_k * \phi_k . \qquad (3.52)$$

где k – номер вида кибератаки в алфавите, y_k – средняя коллективная оценка того, что данный сетевой запрос является кибератакой k-го вида.

Шаг 3 – определение входных параметров. Шаг ориентирован на определение значений входных параметров учебных примеров. Принято, что номер параметра учебного примера равен номеру параметра в БД KDD-99. При этом параметры БД имеющие перечисляемый тип данных подлежат целочисленному кодированию, а параметры имеющие числовой тип данных масштабируются в интервале от 0 до 1.

Этап 3 выполняется до тех пор, пока объем обучающей выборки не превысит $L_{\phi_k}^{max}(\boldsymbol{Z}_1)$, или пока срок ее формирования не превысит $\bar{t}_d$.

Этап 4 – формирование учебных примеров для выборки вида $\boldsymbol{Z}_2$. Этап базируется на разработанной НСМ распознавания кибератак с помощью экспертных знаний. Вход этапа – $\boldsymbol{\Phi}$, $\boldsymbol{D}_2$ и K_ϕ, выход – учебные примеры вида $\boldsymbol{Z}_{2,a} = \left\{\boldsymbol{z}_{2,a}^{(\phi_k)}\right\}_{K\phi} = \left\{\boldsymbol{r}_a^{(\phi_k)}\right\}_{K_\phi}$. При этом выходной сигнал представлен в символьном виде. Суть этапа состоит в том, что для каждого вида кибератак из перечня $\boldsymbol{\Phi}$ реализуется процедура экспертного оценивания данных, которая касается формирования продукционных правил распознавания кибератак на основании анализа семи параметров сетевого запроса. Как и в случае экспертного оценивания сетевых запросов, в качестве базы выбрана процедура экспертной оценки близости видов кибератак, которая использована в модели формирования параметров учебных примеров (2.50-2.55). Однако считается, что в отличии от (2.50) экспертные данные, которые касаются некоторой кибератаки k-го вида представляют собой матрицу типа:

$$D_{2,k} = \begin{vmatrix} \left(X_{k,1,1}^{min}, X_{k,1,1}^{max}\right), & \ldots & \left(X_{k,n,1}^{min}, X_{k,n,1}^{max}\right), & \ldots & \left(X_{k,N,1}^{min}, X_{k,N,1}^{max}\right), y_{k,1} * \phi_k \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \left(X_{k,1,m}^{min}, X_{k,1,m}^{max}\right), & \ldots & \left(X_{k,n,m}^{min}, X_{k,n,m}^{max}\right), & \ldots & \left(X_{k,N,m}^{min}, X_{k,N,m}^{max}\right), y_{k,m} * \phi_k \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \left(X_{k,1,M}^{min}, X_{k,1,M}^{max}\right), & \ldots & \left(X_{k,n,M}^{min}, X_{k,n,M}^{max}\right), & \ldots & \left(X_{k,N,M}^{min}, X_{k,N,M}^{max}\right), y_{k,M} * \phi_k \end{vmatrix}, \qquad (3.53)$$

где N – количество компонент в продукционном правиле для k-го вида кибератаки (N=7), $X_{k,n,m}^{min}, X_{k,n,m}^{max}$ – границs n-го диапазона продукционного правила вида (3.1, 3.2) для k-го вида кибератаки, определенная m-ым экспертом, $y_{k,m}$– определенная m-ым экспертом вероятность реализации кибератаки k-го вида предопределено продукционным правилом, M – количество экспертов.

Учитывая (3.31), множество $\boldsymbol{D}_2$ можно представить так:

$$\boldsymbol{D}_2 = \left\{D_{2,1},...D_{2,K_\phi}\right\}. \quad (3.54)$$

Процедура экспертного оценивания состоит в применении выражений (2.50-2.55) к компонентам (3.53, 3.54). Выходом этапа являются учебные примеры, которые входят в состав $\boldsymbol{Z}_2$ и определяются выражениями вида:

$$\boldsymbol{z}_2^{(\phi_k)} = \left\{\left(\left(X_{k,1}^{min}, X_{k,1}^{max}\right),...\left(X_{k,7}^{min}, X_{k,7}^{max}\right), y_k * \phi_k\right)_i\right\}_I, \quad (3.55)$$

где I – количество примеров для кибератаки k-го вида.

Этап 4 выполняется пока объем обучающей выборки не превысит $L_{\phi_k}^{max}(\boldsymbol{Z}_2)$ или пока срок ее формирования не превысит t_d.

Этап 5 – формирования учебных примеров для выборки вида $\boldsymbol{Z}_3$. Реализация данного этапа схожа с реализацией этапа 3 за исключением того, что выходной сигнал в учебных примерах не рассчитывается. Входными данными являются $\boldsymbol{\Omega}_2$, $L_{\phi_k}^{max}(\boldsymbol{Z}_3)$, t_d, K_ϕ та L_φ, а выходом $\boldsymbol{Z}_{3a}$. Этап выполняется пока объем обучающей выборки не превысит $L_{\phi_k}^{max}(\boldsymbol{Z}_3)$, или срок ее формирования не превысит t_d.

Этап 6 – формирование обучающих примеров для выборки вида $\boldsymbol{Z}_4$. Реализация этапа является симбиозом этапов 3 и 5. Отличием является необходимое количество маркированных и немаркированных учебных примеров. Вход этапа – K_ϕ, $\boldsymbol{\Phi}$, $L_\phi^{min}(\boldsymbol{Z}_1)$, $L_\phi^{max}(\boldsymbol{Z}_1)$, $L_{\phi_k}^{\max}(\boldsymbol{Z}_3)$, $\boldsymbol{\Omega}_1$, $\boldsymbol{\Omega}_2$, $\boldsymbol{D}_3$, L_φ, а выход – $\boldsymbol{Z}_4$. Выполнение этапа разделено на 3 шага.:

Шаг 1 – <u>определение входных параметров</u>. На данном шаге производится определение значений входных параметров учебных примеров вида $\boldsymbol{Z}_1$ и $\boldsymbol{Z}_3$. Процедура определения аналогична той, которая применяется на третьем шаге третьего этапа данного метода.

Шаг 2 – <u>экспертное оценивание параметров сетевого запроса.</u> Этот шаг идентичен первому шагу третьего этапа данного метода.

Шаг 3 – <u>определение выходного сигнала в символьном виде.</u> Этот шаг идентичен второму шагу третьего этапа данного метода.

Этап 6 выполняется до тех пор, пока объем обучающей выборки не превысит $L_{\phi_k}^{\max}(\boldsymbol{Z}_4)$, или пока срок ее формирования не превысит t_d.

Этап 7 – определение выходного сигнала. Этап ориентирован на определение для каждого учебного примера вида $\boldsymbol{Z}_1$ и $\boldsymbol{Z}_2$ величины ожидаемого выходного сигнала НСМ, которая учитывает схожесть параметров видов распознаваемых кибератак. Входными данными этапа являются $\boldsymbol{\Phi}$, $\boldsymbol{D}_1$, K_ϕ, $\boldsymbol{Z}_{1,a}$ и $\boldsymbol{Z}_{2,a}$. Выход – модифицированные множества $\boldsymbol{Z}_{1,b}$ и $\boldsymbol{Z}_{2,b}$. Этап базируется на разработанной модели формирования параметров учебных

примеров и разделен на два шага.

Шаг 1 – расчет меры схожести кибератак. Расчет состоит в экспертном оценивании $\boldsymbol{\Phi}$ и обработке экспертных данных с помощью (2.50-2.55). Выход шага – множество мер схожести видов кибератак между собою:

$$\boldsymbol{O} = \{o_k\}_{K_\phi}, \tag{3.56}$$

где o_k – оценка меры схожести кибератаки k-го вида.

Шаг 2 – учет схожести видов кибератак. Для учета оценки меры схожести в ожидаемом выходном сигнале учебного примера следует в выражениях (3.52, 3.56) компонент $y_k * \phi_k$ заменить на $y_k \times o_k$. В результате выполнения данного шага i-ый учебный пример относящийся к $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$ или $\boldsymbol{Z}_4$ и касающийся k-го вида кибератаки можно записать в следующем виде:

$$z_{1,i}^{(\phi_k)} = (x_1, ...x_{217})_i, y_{k,i} \times o_{k,i}, \tag{3.57}$$

$$z_{2,i}^{(\phi_k)} = \left(\left(X_{k,1}^{min}, X_{k,1}^{max}\right)...\left(X_{k,7}^{min}, X_{k,7}^{max}\right)\right)_i, y_{k,i} \times o_{k,i}. \tag{3.58}$$

Этап 8 – масштабирование параметров. Этап предусматривает приведение величин параметров учебных примеров к интервалу значений допустимых для использования в НСМ [17]. Входом этапа являются учебные примеры из множеств $\boldsymbol{Z}_{1,b}$, $\boldsymbol{Z}_{2,b}$, $\boldsymbol{Z}_{3,a}$. В [17, 18] определено, что масштабирование числовых параметров учебных примеров выполняется так:

$$\bar{a} = \left(a - a^{\min}\right) / \left(a^{\max} - a^{\min}\right), \tag{3.59}$$

где a, $\bar{a}$ – начальная и масштабированная величина параметра, a^{max}, a^{min} – максимальная и минимальная величина a в выборке. Поэтому для каждого из параметров этап разделен на два шага.

Шаг 1 – определение экстремумов. На основании анализа всех примеров обучающей выборки определяется значения a^{max} и a^{min}.

Шаг 2 – реализация масштабирования. Величина параметра масштабируется с использованием выражения (3.59).

Выход этапа – $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, $\boldsymbol{Z}_3$. Если выходом этапа являются $\boldsymbol{Z}_2$, $\boldsymbol{Z}_3$, то на этом выполнение метода заканчивается.

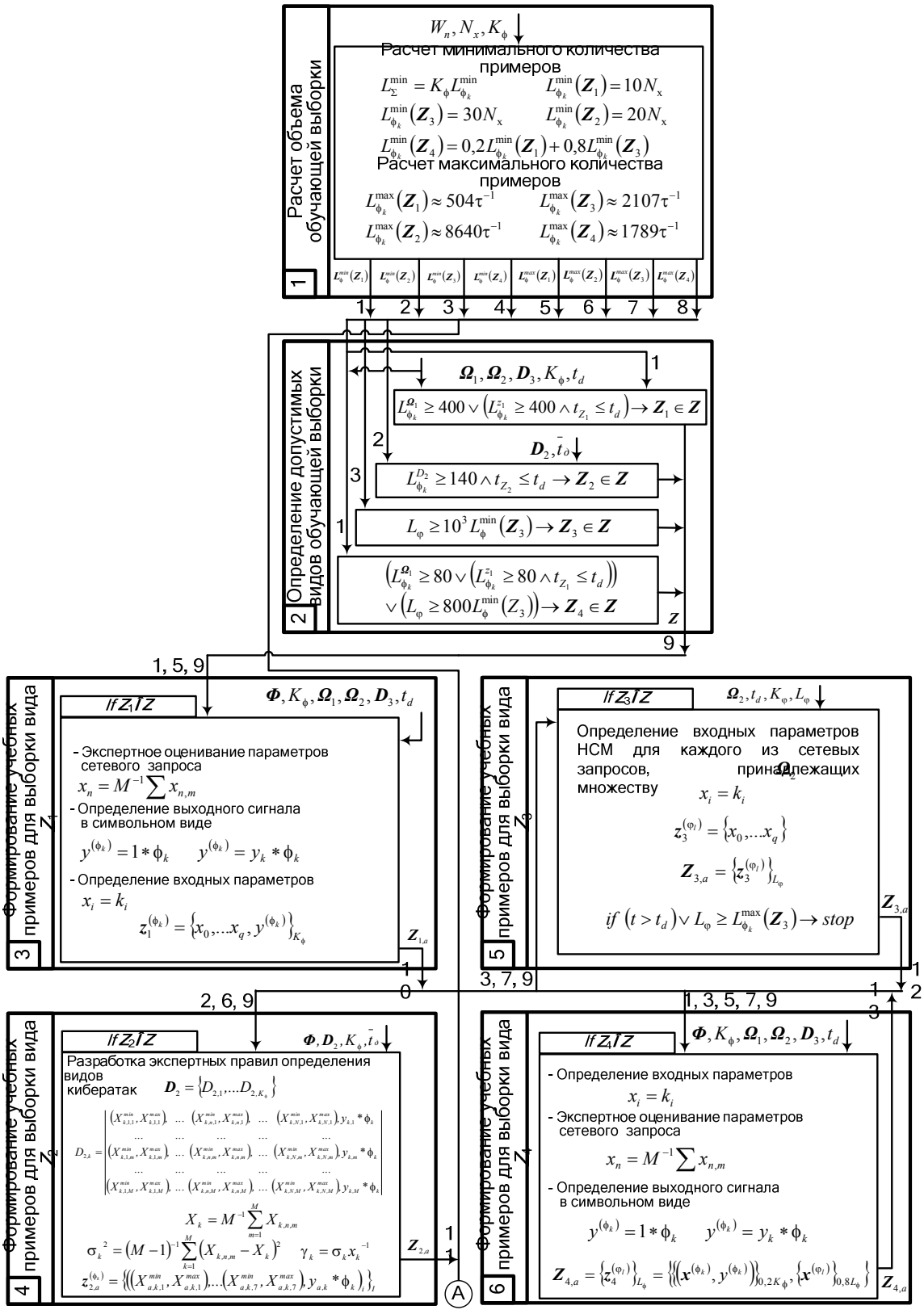

Рисунок 3.5 - Структурно-аналитическая схема метода создания обучающей выборки (этапы 1-6)

Структурно-аналитическая схема последних трех этапов показана на рисунке 3.6

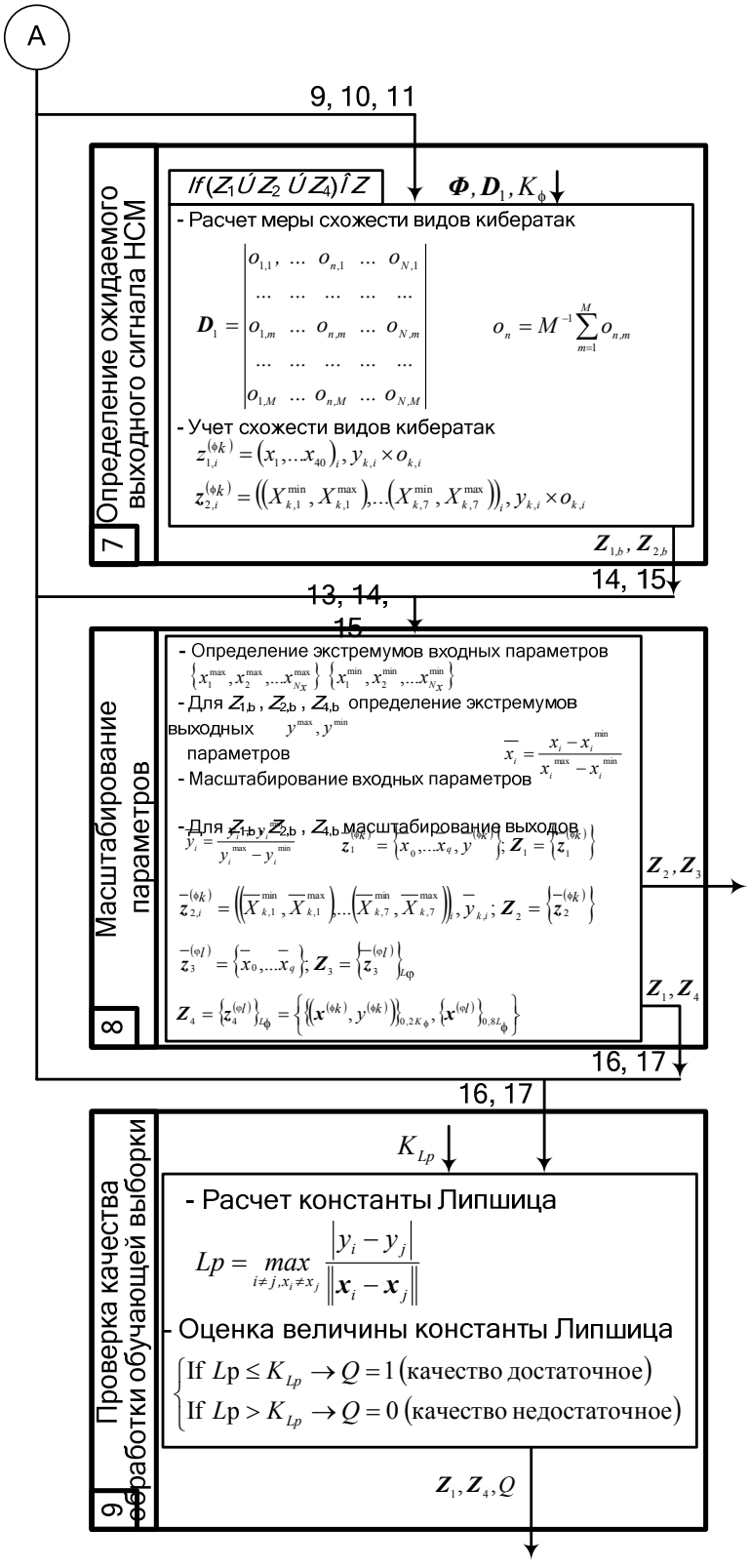

Рисунок 3.6 - Структурно-аналитическая схема метода создания обучающей выборки (этапы 7-9)

Этап 9 – проверка качества предварительной обработки учебных примеров. Входом этапа является обучающей выборка $\boldsymbol{Z}_1$.Для проверки используется константа Липшица вида:

$$Lp = \max_{i \neq j, x_i \neq x_j} \left(\left| y_i - y_j \right| / \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\| \right), \tag{3.60}$$

где $\boldsymbol{x}_i, \boldsymbol{x}_j, y_i, y_j$ – входные и выходные параметры кибератак *i,j*-го видов.

Качество обработки достаточно, при $Lp \leq \mathrm{K}_{Lp}$ ($\mathrm{K}_{Lp} \approx 50$). В противоположном случае следует изменить методы масштабирования, структуру и объем учебной выборки. Выходом этапа является сигнал Q, указывающий на завершение формирования учебной выборки или на необходимость ее модификации. Структурно-аналитическая схема отображения первых шести этапов метода создания учебной выборки показана на рисунке 3.5.

3.4 Метод нейросетевого противодействия сетевым кибератакам

Данный метод является результатом интеграции разработанных элементов методологической базы, метода создания обучающей выборки, НСМ распознавания сетевых кибератак с известными нейросетевыми моделями и методами, применяемыми в области защиты информации. Основной целью его выполнения является определение таких параметров НСР, которые позволяют достичь наибольшей эффективности распознавания сетевых кибератак в заданных условиях эксплуатации. В базовом случае входными данными метода являются множества условий задачи распознавания, экспертных данных и параметров доступных видов НСМ, а выходом является кортеж характеристик та параметров эффективных НСР, состоящий из множеств параметров эффективных НСМ, обучающей выборки и полученных ошибок распознавания. При этом к условиям задачи распознавания сетевых кибератак относятся условия на регистрацию параметров сетевых запросов, ограничения на формирование обучающей выборки, характеристики программно-аппаратного обеспечения системы распознавания, ограничения на ошибку распознавания. Аналитическая модель выполнения методу задана с помощью выражения вида:

$$\left\langle \boldsymbol{\Phi}, \boldsymbol{У}, \boldsymbol{E}, \boldsymbol{Net_a}, \boldsymbol{H}_{Net_a} \right\rangle \rightarrow \left\langle \boldsymbol{Net}_{ve}, \boldsymbol{H}_{Net_{ve}}, \boldsymbol{Z}_{ve}, \boldsymbol{\varepsilon}_{Net_{ve}} \right\rangle, \tag{3.61}$$

где $\boldsymbol{У}$ – множество условий задачи распознавания сетевых кибератак, $\boldsymbol{\Phi}$ – перечень (алфавит) видов кибератак, $\boldsymbol{E}$ – множество экспертных данных, $\boldsymbol{Net_a}$ – множество доступных видов НСМ, $\boldsymbol{H}_{Net_a}$ – параметры доступных видов НСМ, $\boldsymbol{Net}_{ve}$ – множество верифицированных эффективных видов НСМ, $\boldsymbol{H}_{Net_e}$ – параметры верифицированных эффективных видов НСМ, $\boldsymbol{Z}_{ve}$ – множество учебных примеров, соответственных $\boldsymbol{Net}_{ve}$, $\boldsymbol{\varepsilon}$ – множество ошибок распознавания для $\boldsymbol{Net}_{ve}$.

Учитывая разработанную модель правил определения эффективных видов НСМ и метод формирования параметров учебных примеров, для реализации

выражения (3.61), в данном методе предусмотрено выполнение следующих этапов:

1) Определение условий создания и использования НСР.
2) Создание обучающей выборки.
3) Определение допустимых видов НСМ.
4) Определение эффективных видов НСМ.
5) Определение параметров НСМ эффективных видов.
6) Обучение НСМ.
7) Верификация НСР.

Структурная схема разработанного метода показана на рисунке 3.7. Показанные на рисунке 3.7 этапы метода нейросетевого распознавания детализируются так.

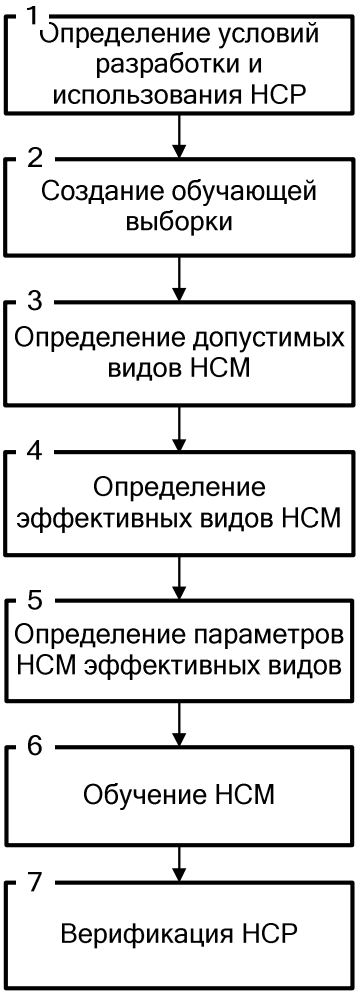

Рисунок 3.7 - Структурная схема метода нейросетевого противодействия кибератакам

Этап 1 – определение условий разработки и использования НСР. Основным назначением данного этапа является определение условий создания и использования НСР распознавания сетевых кибератак. Соответственно результатов второго раздела, эти условия можно охарактеризовать с помощью:

– параметров регистрации параметров сетевых запросов ($\boldsymbol{\Theta}_{\text{сз}}$),
– параметров, которые характеризуют условия формирования обучающей

выборки ($\boldsymbol{\Theta}_{\text{ов}}$) и разработки НСМ ($\boldsymbol{\Theta}_{\text{нсм}}$),

– временных параметров использования НСР ($\boldsymbol{\Theta}_{\text{вп}}$).

Отметим, что элементы множества $\boldsymbol{\Theta}_{\text{сз}}$ характеризуют частоту регистрации сетевых запросов, сетевые протоколы по которым производится регистрация параметров и перечень регистрируемых параметров. Элементы множеств $\boldsymbol{\Theta}_{\text{ов}} = \langle \boldsymbol{\Phi}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, \boldsymbol{Z}_{\text{ов}} \rangle$, $\boldsymbol{\Theta}_{\text{нсм}} = \langle N_x, N_y, \boldsymbol{Net}_{\boldsymbol{a}}, \varepsilon_{\max}, \tau, \boldsymbol{R}_a, \boldsymbol{\alpha} \rangle$ и $\boldsymbol{\Theta}_{апо} = \langle k_{нсс}, W_n \rangle$ детализированы в разработанных моделях правил определения эффективных видов НСМ, формирования параметров учебных примеров, распознавания сетевых кибератак с помощью экспертных знаний, модели ГНС и метод создания обучающей выборки.

Базой для определения указанных условий являются требования к системе распознавания кибератак ($\boldsymbol{Q}_{\text{срк}}$), характеристики программно-аппаратного обеспечения системы распознавания ($\boldsymbol{X}_{\text{пао}}$), а также материально-технические и трудовые ресурсы ($\boldsymbol{R}_{\text{срк}}$), которые выделяются на ее разработку. Так как связь между приведенными категориями ограничений и определенной базой слабо формализована, то в базовом варианте определение ограничений предлагается реализовать с помощью экспертных данных ($\boldsymbol{E}_{срк}$). Обработку экспертных данных предлагается проводить подобно процедуре экспертной оценки близости видов кибератак, которая используется в разработанной модели формирования параметров учебных примеров, а математический аппарат которой задан выражениями (2.50-2.55). В обобщенном виде реализацию данного этапа можно представить с помощью следующего выражения:

$$f\left(\boldsymbol{Q}_{\text{срк}}, \boldsymbol{R}_{\text{срк}}, \boldsymbol{X}_{\text{пао}}, \boldsymbol{E}_{\text{срк}}\right) = \langle \boldsymbol{\Theta}_{сз}, \boldsymbol{\Theta}_{ов}, \boldsymbol{\Theta}_{нсм}, \boldsymbol{\Theta}_{апо}, \boldsymbol{\Theta}_{вп} \rangle. \qquad (3.62)$$

Этап 2 – создание обучающей выборки. Выполнение данного этапа является реализацией метода создания обучающей выборки. Входными данными этапа является кортеж $\langle \boldsymbol{\Phi}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, N_x, N_y, \boldsymbol{\Theta}_{нв}, W_n \rangle$, а выходом $\langle \boldsymbol{Z}_1, \boldsymbol{Z}_2, \boldsymbol{Z}_3 \rangle$. Математическое обеспечение этапа составляют выражения (2.27, 2.28, 2.33, 2.50-2.55, 3.29-3.60).

Этап 3 – определение допустимых видов НСМ. Этап базируется на предложенном принципе допустимости использования вида НСМ и разработанной модели правил определения эффективных видов НСМ. Входными данными этапа является приемлемый срок создания НСР распознавания кибератак (t_d), множество доступных видов НСМ ($\boldsymbol{Net}_{\boldsymbol{a}}$), средний срок создания одного учебного примера ($\bar{t}_v$), допустимая ошибка обучения НСМ (ε), продолжительность одной вычислительной операции процесса обучения (τ), количество входных параметров НС (N_x). Выходом

этапа является множество допустимых видов НСМ ($\boldsymbol{Net_d}$). В общем случае выполнение этапа состоит в расчете разработанных выражений (2.31, 2.32). В случае использования типовых требований к разработке отечественных СРК можно использовать упрощенные выражения (2.36, 2.37).

Этап 4 – определение эффективных видов НСМ. Этап базируется на предложенных принципах определения множества эффективных видов НСМ, оценивания эффективности вида НСМ, предназначенной для распознавания кибератак на сетевые РИС, а также созданной на их основе модели правил определения эффективных видов НСМ. Входными данными этапа является множество допустимых видов НСМ ($\boldsymbol{Net_d}$), условия разработки НСМ ($\boldsymbol{\Theta}_{\text{нсм}}$), экспертные данные, касающиеся разработки системы распознавания и оценки критериев эффективности вида НСМ ($\boldsymbol{E}_{срк}$).

Реализация этапа сводится к тому, что с использованием данных таблицы 2.2 с помощью экспертных данных для каждого допустимого вида НСМ определяются значения критериев эффективности. После этого, базируясь на формализованных условиях задачи распознавания, для каждого *k*-го критерия определяются значения весового коэффициента α_k, с помощью которого учитывается значимость этого критерия. Далее с помощью выражений (2.40, 2.41) проводится определение множества эффективных видов НСМ. Расчет наиболее эффективного вида НСМ реализуется с помощью выражения (2.42).

Этап 5 – определение параметров НСМ эффективных видов. Этап предназначен для определения таких параметров эффективных видов НСМ, которые обеспечивают их максимальную вычислительную мощность (Θ) и минимальную ошибку распознавания (ε). Традиционно Θ оценивается с помощью отношения внесенных в память учебных примеров к количеству синаптических связей. Для определения параметров использовано выражение:

$$\begin{cases} \Theta(\boldsymbol{\Lambda}) \to max \\ \varepsilon(\boldsymbol{\Lambda}) \to min \end{cases}, \tag{3.63}$$

где $\boldsymbol{\Lambda} = \{\lambda_1, \lambda_2, ...\}$ – множество параметров НСМ.

Расчет величин $\{\lambda_1, \lambda_2, ...\}$ предлагается проводить методами, специфичными для вида НСМ. Например, при использовании МСП к указанным параметрам относятся количество скрытых нейронных слоев, количество нейронов в каждом скрытом слое, структура связей между нейронами. Для расчета этих параметров можно использовать результаты [33, 34]. Выходом этапа является множество эффективных видов НСМ с определенными параметрами ($\boldsymbol{Net}_e^o$).

Этап 6 – обучение НСМ. Этап предназначен для расчета весовых коэффициентов синаптических связей НСМ, которые входят во множество $\boldsymbol{Net}_e^o$. Для этого используется обучающая выборка, сформированная в

результате выполнения этапа 2. Расчет реализуется с помощью методов, характерных для вида НСМ. Для обучения разработанной НСМ распознавания сетевых кибератак с помощью экспертных знаний применяются выражения (3.1 - 3.13), а для обучения разработанной глубокой НСМ – выражения (3.14 - 3.22). Выходом этапа является $\boldsymbol{H}_{Net_e^o}$ – множество параметров эффективных видов НСМ, которые входят в состав $\boldsymbol{Net}_e^o$.

Этап 7 – верификации НСР. Предлагается проводить верификацию разработанных НСР с позиций допустимости ошибки распознавания Входными данными этапа является множество эффективных НСМ с определенными параметрами ($\boldsymbol{Net}_e^o$), параметры указанных НСМ ($\boldsymbol{H}_{Net_e^o}$), максимально допустимая ошибка распознавания (ε_{max}).Отметим, что множества $\boldsymbol{H}_{Net_e}, \boldsymbol{Z}_{ve}$, $\boldsymbol{\varepsilon}_{Net_{ve}}$ и $\boldsymbol{\xi}_{Net_{ve}}$ формируются на основе соответствующих данных $net_i \in \boldsymbol{Net}_{ve}$.

Выполнения этапа состоит в реализации правила:

$$If \;\; net_i \in \boldsymbol{Net}_e^o \wedge \varepsilon_i \le \varepsilon_{\max} \rightarrow net_i \in \boldsymbol{Net}_{\text{ve}}, \qquad (3.64)$$

где ε_i – ошибка распознавания net_i.

Выход этапа, который также является и выходом метода состоит из множеств $\boldsymbol{Net}_{ve}$, $\boldsymbol{H}_{Net_e}$, $\boldsymbol{\varepsilon}_{Net_{ve}}$.

В данном разделе решалась научная задача разработки нейросетевых моделей и методов распознавания кибератак на сетевые ресурсы информационных систем. Основные результаты раздела следующие:

– Получили дальнейшее развитие нейросетевые модели распознавания сетевых кибератак, в которых за счет реализации предложенных принципов и разработанного математического аппарата, обеспечивается возможность оперативного создания достаточно точных нейросетевых средств.

– Впервые разработан метод создания обучающей выборки, который за счет применения предложенных принципов использования нейронных сетей для распознавания сетевых кибератак и разработанной модели формирования параметров учебных примеров, обеспечивает адаптацию параметров обучающей выборки к условиям ее формирования и использования, а также предоставляет возможность уменьшения количества учебных итераций применяемой нейросетевой модели.

– Получил дальнейшее развитие метод нейромережевого распознавания сетевых кибератак, который являясь симбиозом известных решений с разработанными моделями использования нейросетевых средств, и разработанном методе создания обучающей выборки, обеспечивает достаточную оперативность и точность распознавания с учетом ограничений, касающихся создания обучающей выборки.

4 РАЗРАБОТКА НЕЙРОСЕТЕВОЙ СИСТЕМЫ ПРОТИВОДЕЙСТВИЯ КИБЕРАТАКАМ И ЭКСПЕРИМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ

4.1 Архитектура нейросетевой системы

Архитектура предложенной НСС СПА разработана с позиций интеграции решений разработанного метода нейросетевого распознавания с решениями, применяемыми в известных СОА. Как показано на рисунке 4.1. структура системы состоит из 20 модулей, объединенных в 4 подсистемы и отдельного модуля управления. Назначением модуля управления системой МУС является перевод системы в следующие режимы эксплуатации: РОУЭ – определение условий эксплуатации, РОН – определения настроек, РР – распознавания сетевых кибератак, РО – остановка.

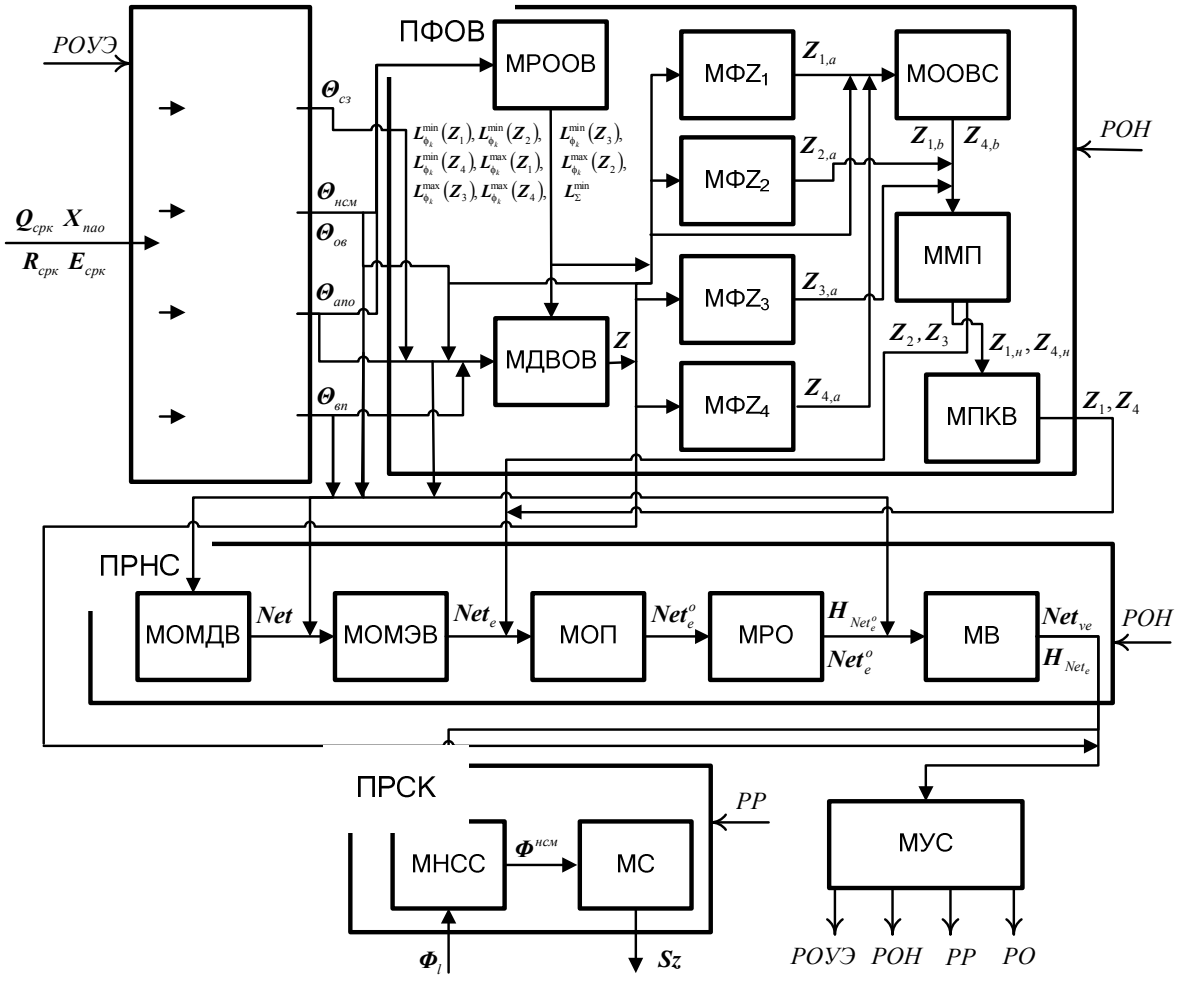

Рисунок 4.1 - Структура нейросетевой системы противодействия

Назначение подсистем:

– ПОУ – подсистема определения условий создания НСР, предназначенная для формирования параметров, характеризующих указанные условия и являющиеся базисом для расчетов в других подсистемах;

– ПФОВ – подсистема формирования адаптивной обучающей выборки,

которая предназначена для определения ее минимального и максимального объема, допустимых видов, определения ожидаемого выходного сигнала НСМ для эталонов кибератак, определения и масштабирования входных и выходных параметров НСМ;

– ПРНС – подсистема разработки НСМ, основным назначением которой является определение параметров НСМ, которые позволяют наиболее эффективно распознавать сетевые кибератаки;

– ПРСК – подсистема распознавания сетевых кибератак, которая, базируясь на результатах работы других подсистем, позволяет реализовать распознавание сетевых кибератак.

Назначение отдельных модулей разработанной НСС описано в таблице 4.1.

Таблица 4.1 - Состав нейросетевой системы противодействия сетевым кибератакам

Подсистема	Модуль	Назначение модуля
ПОУ	МПФСЗ	Определение параметров фиксации и предварительной обработки сетевых запросов
	МУФОВ	Определение условий формирования обучающей выборки и разработки НСМ
	МОАПО	Определение условий использования АПО, которые реализуют НСР
	МВП	Определение временных параметров использования НСР
ПФОВ	МРООВ	Расчет объема обучающей выборки
	МДВОВ	Определение допустимых видов обучающей выборки
	МФZ_1	Формирование обучающей выборки вида Z_1
	МФZ_2	Формирование обучающей выборки вида Z_2
	МФZ_3	Формирование обучающей выборки вида Z_3
	МФZ_4	Формирование обучающей выборки вида Z_4
	МООВС	Определение ожидаемого выходного сигнала учебных примеров вида Z_1, Z_2, Z_4
	ММП	Масштабирования параметров учебных примеров
	МПКВ	Проверка качества обработки примеров обучающей выборки
ПРНС	МОМДВ	Определение множества допустимых видов НСМ
	МОМЭВ	Определение множества эффективных видов НСМ
	МОП	Определение параметров НСМ, которые входят во множество эффективных видов
	МРО	Реализация обучения НСМ
ПРНС	МВ	Верификация НСМ
ПРСК	МНСР	Нейросетевое распознавание
	МС	Сигнализация о распознанных кибератаках

Разработанная НСС начинает функционировать в режиме определения параметров, которые характеризуют условия эксплуатации. При этом срабатывают модули ПОУ. Источником данных для ПОУ являются требования к СРК ($\boldsymbol{Q}_{срк}$), характеристики программно-аппаратного обеспечения СРК ($\boldsymbol{X}_{пао}$) и ресурсы ($\boldsymbol{R}_{срк}$), выделяемые на ее разработку. Для их обработки в ПОУ

также подается множество соответствующих экспертных данных ($\boldsymbol{E}_{срк}$). Результат срабатывания модулей ПОУ:

– МПФЗС – кортеж параметров обработки сетевых запросов ($\boldsymbol{\Theta}_{сз}$), определенный с использованием разработанной модели формирования параметров учебных примеров и данных [33].

– МУФОВ – кортеж параметров, характеризующих условия создания обучающей выборки $\boldsymbol{\Theta}_{ов} = \langle \boldsymbol{\Phi}, \boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3, \boldsymbol{Z}_{ов} \rangle$, компоненты которого определены в (3.23) и кортеж $\boldsymbol{\Theta}_{нсм} = \langle N_x, N_y, \boldsymbol{Net}_{\boldsymbol{a}}, \varepsilon_{\max}, \tau, \boldsymbol{R}_a, \boldsymbol{\alpha} \rangle$, компоненты которого определены в (2.12, 2.46, 2.17, 2.6, 2.15, 2.16, 2.40).

– МВАПС – кортеж параметров, характеризующих использование АПО $\boldsymbol{\Theta}_{апо} = \langle k_{нсс}, W_n \rangle$, компоненты которого определены в (3.23).

– МВП – кортеж параметров, характеризующих временные характеристики использования НСС СРК $\boldsymbol{\Theta}_{вп} = \langle t_d, \bar{t}_v, k_1, k_2, t_{\max} \rangle$, компоненты которого определены в (2.4, 2.13, , 2.17, 2.18, 2.33).

Если параметры, характеризующие условия использования АПО, не отвечают заданным ограничениям, то МУС переводит НСС в режим останова. В противоположном случае система переводится в режим определения настроек, в котором срабатывают подсистемы ПФОВ и ПРНС.

Первым в ПФОВ срабатывает МРООВ. На его вход подаются W_n, N_x и K_ϕ. Выходом являются $L_\Sigma^{\min}$, $\boldsymbol{L}_{\phi_k}^{\min}(\boldsymbol{Z}_1)$, $\boldsymbol{L}_{\phi_k}^{\min}(\boldsymbol{Z}_2)$, $\boldsymbol{L}_{\phi_k}^{\min}(\boldsymbol{Z}_3)$, $\boldsymbol{L}_{\phi_k}^{\min}(\boldsymbol{Z}_4)$, $\boldsymbol{L}_{\phi_k}^{\max}(\boldsymbol{Z}_1)$, $\boldsymbol{L}_{\phi_k}^{\max}(\boldsymbol{Z}_2)$, $\boldsymbol{L}_{\phi_k}^{\max}(\boldsymbol{Z}_3)$, $\boldsymbol{L}_{\phi_k}^{\max}(\boldsymbol{Z}_4)$. Для их расчетов используются (3.29-111). Определенные в МРООВ параметры которые характеризуют минимально допустимый объем обучающей выборки и определенные в ПОУ K_ϕ, $\bar{t}_\partial$, $\boldsymbol{\Phi}$, $\boldsymbol{\Omega}_1$, $\boldsymbol{\Omega}_2$, $\boldsymbol{D}_2, \boldsymbol{D}_3$ передаются в МДВОВ, где с помощью (3.44-3.49) формируется множество допустимых видов обучающей выборки $\boldsymbol{Z}$, которая и является выходом этого модуля.

Если множество $\boldsymbol{Z}$ пустое, то НСС переводится в режим останова. В противоположном случае состав $\boldsymbol{Z}$ определяет срабатывание модулей, в которых реализуется формирование обучающей выборки – МФZ_1, МФZ_2, МФZ_3, МФZ_4. Математическое обеспечение этих модулей составляют (3.50-3.55). Выходом являются множества $\boldsymbol{Z}_{1,a}$, $\boldsymbol{Z}_{2,a}$, $\boldsymbol{Z}_{3,a}$, $\boldsymbol{Z}_{4,a}$ соответственно.

Если хотя бы одна из обучающих выборок вида $\boldsymbol{Z}_1$, $\boldsymbol{Z}_4$ является допустимой, то срабатывает МООВС. Кроме $\boldsymbol{Z}_{1,a}$ и $\boldsymbol{Z}_{4,a}$, на его вход подаются $\boldsymbol{\Phi}$, $\boldsymbol{D}_1$, K_ϕ. Выходом являются множества учебных примеров вида $\boldsymbol{Z}_{1,b}$, $\boldsymbol{Z}_{4,b}$ в которых ожидаемый выходной сигнал представлен в числовом виде. Для расчета используется (2.50-2.55, 3.56-3.58). Следующим срабатывает ММП, который реализует масштабирование параметров учебных примеров. Для нормализации используется (3.59). Выход модуля – множества $\boldsymbol{Z}_{1,н}$, $\boldsymbol{Z}_2$, $\boldsymbol{Z}_3$, $\boldsymbol{Z}_{4,н}$. Если $\boldsymbol{Z}_{1,н}$ и/или $\boldsymbol{Z}_{4,н}$ входят в состав допустимых, то срабатывает МПКВ, в

котором с помощью (3.60) проверяется качество предварительной обработки учебных примеров. Если качество достаточное, то выходом $\boldsymbol{Z}_1 = \boldsymbol{Z}_{1,н}$, $\boldsymbol{Z}_4 = \boldsymbol{Z}_{4,н}$, в противоположном случае $\boldsymbol{Z}_1 = \varnothing$, $\boldsymbol{Z}_4 = \varnothing$.

После этого в порядке МОМДВ→МВМЭВ→МОП→МРО→МВ срабатывают модули ПРНС. Назначением МОМДВ является определение множества допустимых видов НСМ ($\boldsymbol{Net}$). Входными данными модуля являются величины $\bar{t}_v, \tau, t_d, \boldsymbol{net}_1, \boldsymbol{net}_2$. Выходом является множество $\boldsymbol{Net}$, для определения которого используются (2.31,2.32, 2.36, 2.37). Получив $\boldsymbol{Net}$, $(\boldsymbol{R}_a, \boldsymbol{\alpha}) \in \boldsymbol{\Theta}_{нмм}$ в МВМЭВ определяется множество эффективных видов НСМ – $\boldsymbol{Net}_e$ и наиболее эффективный вид НСМ – $net_e^{max} \in \boldsymbol{Net}_e$. Для этого используются (2.39-2.42). $\boldsymbol{Net}_e$ и net_e^{max} передаются в МОП, где с помощью (3.3-3.6, 3.12, 3.13) и данных [33] определяются параметры таких НСМ. Выходом МВП является $\boldsymbol{Net}_e^o$ – множество эффективных видов НСМ с оптимизированными параметрами, которые вместе с $\boldsymbol{Z}_1$, $\boldsymbol{Z}_2$, $\boldsymbol{Z}_3$, $\boldsymbol{Z}_4$ подаются в МРО. В этом модуле проводится обучение НСМ, из состава $\boldsymbol{Net}_e^o$. Для этого используются данные [76], в которых отображены методы обучения различных видов НСМ. В случае использования модифицированных НСМ вида PNN используются результаты раздела 3.1. Выходом МРО является $\boldsymbol{H}_{Net_e^o}$ – множество параметров эффективных видов НСМ. Последним в ПРНС срабатывает МВ в котором в соответствии с седьмым этапом разработанного метода нейросетевого распознавания проводится верификация разработанных НСМ. На вход модуля подаются $\boldsymbol{Net}_e^o$, $\boldsymbol{H}_{Net_e^o}$, $\varepsilon_{max} \in \boldsymbol{\Theta}_{нмм}$, $\boldsymbol{\xi}^{\partial} \in \boldsymbol{\Theta}_{апз}$, $T_з \in \boldsymbol{\Theta}_{чп}$ та $\boldsymbol{\xi}^{max}$, а выходом является $\boldsymbol{Net}_{ve}$ – множество верифицированных НСМ и $\boldsymbol{H}_{Net_e}$ – множество параметров верифицированных НСМ. После получения соответствующего сигнала МУС переводит систему в режим распознавания.

В режиме распознавания полученное из хранилища множество параметров сетевых запросов $\boldsymbol{\Phi}_l$ подается в МНСС ПРСК. В этом модуле с помощью $\boldsymbol{Net}_{ve}$ и $\boldsymbol{H}_{Net_e}$ формируется множество результатов нейросетевого сравнения $\boldsymbol{\Phi}^{нсм}$. Если в состав $\boldsymbol{Net}_{ve}$ входит только одна НСМ, то составляющие $\boldsymbol{\Phi}^{нсм}$ имеют вид:

$$\phi_i^{нсм} = (i, y), \tag{4.1}$$

где i –номер кибератаки, y – величина выходного сигнала НСМ.

Если в состав $\boldsymbol{Net}_{ve}$ входит несколько НСМ, то составляющие $\boldsymbol{\Phi}^{нсм}$ определены так:

$$\phi_i^{нмм} = (i, y_1, y_2, ... y_N), \quad (4.2)$$

где y_n – величина выходного сигнала *n*-ой НСМ, N – количество элементов множества $\boldsymbol{Net}_{ve}$. $\boldsymbol{\Phi}^{нсм}$ передается в МС, где формируется сигнал ***Sz*** о результатах распознавания.

4.2 Экспериментальная установка

Экспериментальная установка представляет собой аппаратно-программный комплекс, предназначенный для проведения экспериментальных исследований разработанных моделей и методов, а также созданной на их основе нейросетевой системы распознавания сетевых кибератак.

Вычислительные возможности и конфигурация аппаратного обеспечения экспериментальной установки определялись с позиций обеспечения минимально допустимых требований к универсальным СРК типа Snort, приспособленных для разворачивания на операционных системах семейств Windows и Linux [68, 78]. Также учтено, что сетевые возможности аппаратного обеспечения должны обеспечивать потенциальную возможность перехвата сетевого трафика, соответствующего стеку протоколов TCP/IP. Поэтому в базовой конфигурации использован универсальный персональный компьютер на основе процессора Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz с оперативной памятью объемом 3,7 ГБ, жестким диском объема 1 ТБ и сетевой картой Attansic L1 Gigabit Ethernet 10/100/1000Base-T. При этом следует отметить, что для функционирования программного обеспечения экспериментальной установки достаточно 20 ГБ постоянной памяти.

Требования к функциональности программного обеспечения экспериментальной установки определены с позиций обеспечения:

– Регистрации параметров сетевых запросов, передаваемых в соответствии со стеком протоколов TCP/IP.
– Предварительной обработки параметров зарегистрированных сетевых запросов для приведения их к виду, пригодному для использования в качестве входных параметров НСМ.
– Возможности использования общедоступных БД для обучения НСМ распознавания сетевых кибератак.
– Возможности использования общедоступных БД для формирования продукционных правил распознавания сетевых кибератак.
– Реализации процесса обучения и тестирования общеизвестных НСМ распознавания сетевых кибератак.
– Реализации процесса обучения и тестирования разработанных на базе ГНС и PNN НСМ распознавания сетевых кибератак.
– Сравнения эффективности разработанных моделей и методов с известными аналогичными моделями и методами.

Для обеспечения указанных требований использовано несколько БД, программных библиотек и приложений, основные характеристики которых показаны в таблице 4.2.

Таблица 4.2 - Основные характеристики программных средств экспериментальной установки

Название	Источник получения	Назначение
KDD-99	В свободном доступе (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)	Источник данных для обучения НСМ.
NSL-KDD	В свободном доступе (http://www.unb.ca/cic/research/datasets/nsl.html)	
Snort	В свободном доступе (http://www.snort.org)	- Перехват сетевых запросов, передаваемых в соответствии со стеком протоколов TCP/IP. - Предварительная обработка параметров перехваченных сетевых запросов. - Распознавание сетевых кибератак
Win Sniffer	В свободном доступе (http://www.inattack.ru/download.php?id=5)	Перехват параметров сетевых запросов, передаваемых в соответствии со стеком протоколов TCP/IP.
Tensorflow	В свободном доступе (https://www.tensorflow.org/install)	Программная библиотека, в которой содержатся компоненты для реализации НСМ на базе ГНС.
MPNNpro	Собственная разработка	Реализация НСМ на базе MPNN.
kddNeural	Собственная разработка	Реализация НСМ на базе PNN.
DNETpro	Собственная разработка	Реализация НСМ вида ГНС.
NeuroPro	В свободном доступе (http://www.neuropro.ru)	Реализация НСМ вида МСП

Отметим, что часть приложений находятся в свободном доступе, а часть приложений разработана под руководством автора на основании математических моделей, предложенных в данной диссертационной работе.

В большинстве экспериментов основным источником учебных примеров для обучения и тестирования НСМ являлась БД KDD-99 и ее более современный вариант БД NSL-KDD, что объясняется полнотой и достоверностью представленных в данных [105]. В указанных БД содержаться параметры сетевых запросов соответствующих следующим состояниям защищенности:

- Distributed Denial of Service (DDoS) – кибератаки нацеленные на блокирования сети.
- Probe — кибератаки нацеленные на сканирование информации или на определение уязвимостей сети для дальнейшего ее использование для атак на другие сети.
- Remote to Local (U2L) – кибератака нацеленная на создание удаленного неавторизированного соединения с сетью с помощью посылки пакетов в эту сеть.
- User to Root (U2R) – кибератака нацеленная на получение обычным пользователем прав администратора.
- Normal – нормальный сетевой запрос (кибератака отсутствует)

При этом, как показано в таблице 4.3, каждый представленный класс сетевых кибератак подразделяется на несколько видов.

Таблица 4.3 - Характеристики базы данных KDD-99

Класс кибератаки	Вид кибератаки	Количество записей
DoS	neptune	1072017
	smurf	2807886
	Pod	264
	teardrop	979
	land	21
	back	2203
U2R	buffer_overflow	33
	loadmodule	9
	perl	3
	rootkit	10
R2L	guess_passwd	53
	ftp_write	8
	imap	12
	phf	4
	multihop	7
	warezmaster	20
	warezclient	1020
	spy	2
Probe	portsweep	10413
	ipsweep	12481
	satan	15892
	nmap	2316

Атрибутами полей БД являются 41 параметр сетевых запросов, соответствующих стеку протоколов TCP/IP, а также название состояния защищенности (normal или вид кибератаки). Указанные атрибуты объединяются в 4 группы:
- Базовые атрибуты – параметры TCP/IP соединения (1-10).

- Временные атрибуты трафіка – это атрибуты, которые оцениваются на протяжении 2 секунд соединения (22-31).
- Атрибуты контента (11-21).
- Атрибуты хоста трафика (32-41).

База данных содержит значения каждого из атрибутов для распознавания следующих видов сетевых кибератак:

- Distributed Denial of Service (DDoS) — кибератаки нацеленные на блокирования сети.
- Probe — кибератаки нацеленные на сканирование информации или на определение уязвимостей сети для дальнейшего ее использование для атак на другие сети.
- Remote to Local (U2L) — кибератака нацеленная на создание удаленного неавторизированного соединения с сетью с помощью посылки пакетов в эту сеть.
- User to Root (U2R) — кибератака нацеленная на получение обычным пользователем прав администратора.
- Частично характеристики атрибутов БД представлены в таблице 4.4. Номера указанных атрибутов соответствуют номерам входных нейронов НСМ.

Таблица 4.4 - Характеристика атрибутов баз данных KDD-99 и NSL-KDD

Название атрибута	Описание атрибута
duration	Время соединения в секундах
protocol_type	Тип протоколу(TCP, UDP)
service	Сетевой сервис (http, telnet, etc)
flag	Статус соединения (соединение, ошибка)
src_bytes	Количество переданной информации от источника к получателю в байтах
dst_bytes	Количество переданной информации от получателю к источнику в байтах

За исключением DNETpro и MPNNpro использованные программы приспособлены к функционированию на операционных системах семейства Windows. DNETpro и MPNNpro ориентированы на операционные системы семейства Linux, хотя после незначительных переделок могут быть адаптированы к Windows.

Основной частью экспериментальной установки являются программы DNETpro, kddNeural и MPNNpro.

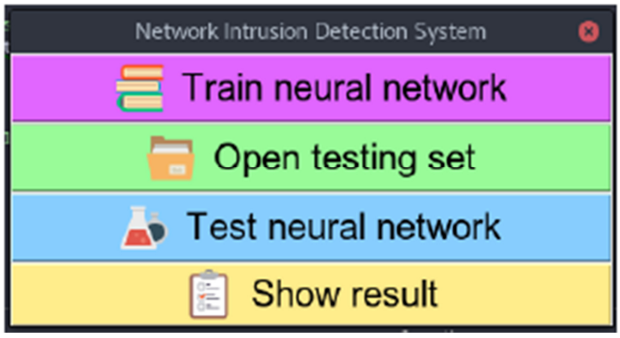

Рисунок 4.2 - Главное окно программы DNET_pro

DNETpro разработана на базе предложенной в разделе 3.2 модели ГНС, адаптированной к распознаванию сетевых кибератак. Использован язык программирования Python и библиотека Tensorflow. Выбор указанных инструментальных средств разработки объясняется их высокой эффективностью при реализации НСМ на базе ГНС [106, 107]. Листинг программного кода DNETpro показан в приложении А, а главное окно показано на рисунке 4.2.

Программа DNETpro позволяет обучить НСМ, провести тестирование сети и в графическом виде отобразить результаты тестирования.

Процесс обучения DNETpro запускается с помощью кнопки «Train neural network». При этом обучающая выборка должна находится в каталоге «\model\data\», который в свою очередь должен быть размещена в каталоге в котором находится запускающий файл с названием main.py. Также следует отметить, что файл с обучающей выборкой должен иметь название train. Формат файла .csv. Окончание процесса обучения сигнализируется сообщением, показанным на рисунке 4.3

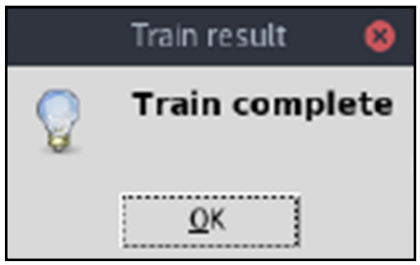

Рисунок 4.3 - Информационное сообщение о завершении процесса обучения

После обучения DNETpro можно использовать анализа зарегистрированных параметров сетевых запросов. Отметим, что и обучающая тестовая выборка также должна быть записана в файл формата .csv. Для указания целевого файла с тестовой выборкой следует воспользоваться кнопкой «Open testing set». В ответ открывается диалоговое окно запроса (Рисунок 4.4).

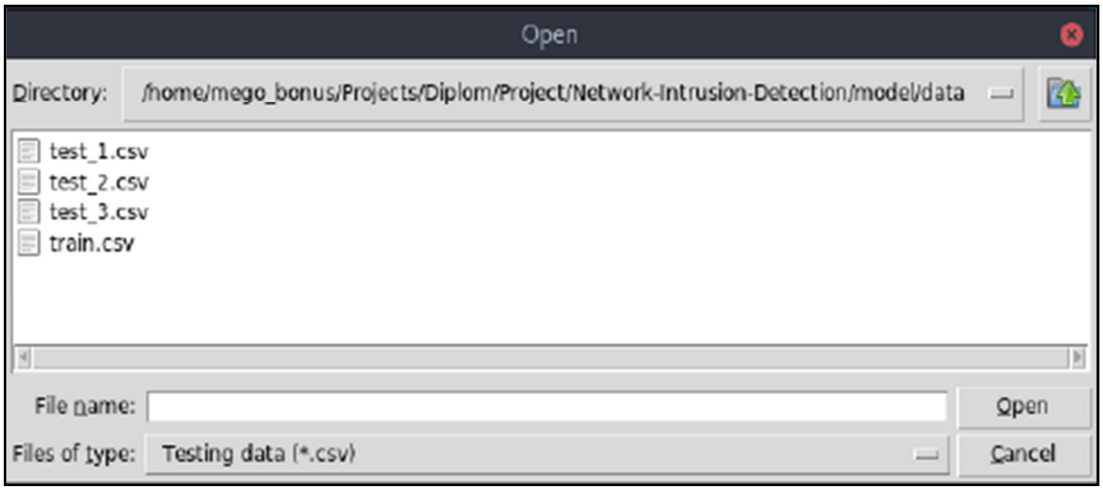

Рисунок 4.4 - Диалоговое окно указания файла с тестовой выборкой

Рисунок 4.5 - Сигнализация о завершении процесса распознавания

Визуализацию полученных результатов распознавания предусмотрено реализовать в виде диаграммы, пример которой показан на рисунке 4.6. Вход в режим просмотра осуществляется с помощью кнопки «Show result». Отметим, что на рисунке 4.6. отображена почасовая диаграмма распознанных разнотипных сетевых кибератак на протяжении 24 часов.

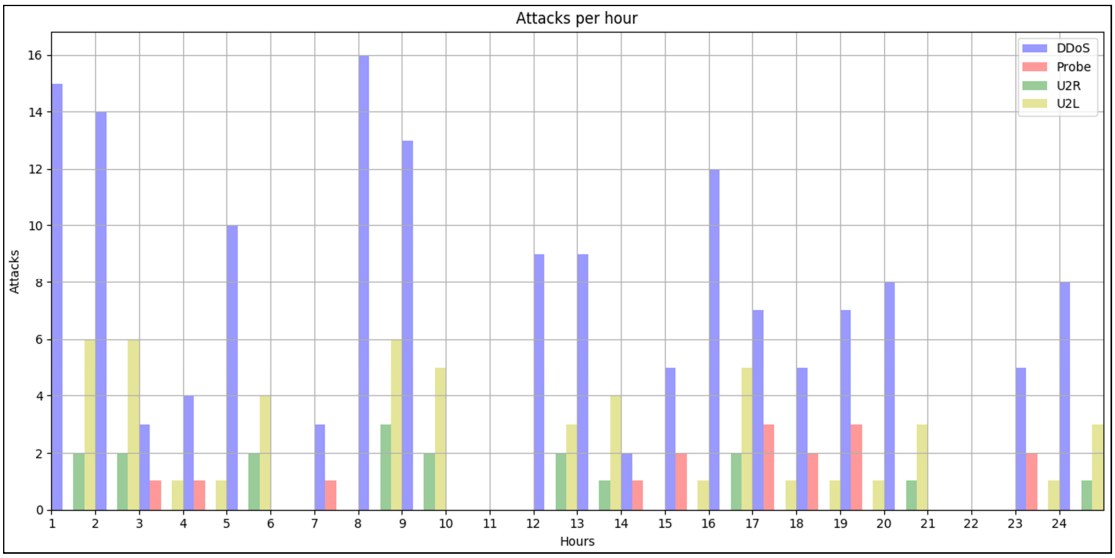

Рисунок 4.6 - Отображение результатов распознавания

Программы MPNNpro и kddNeural базируются на разработанной в разделе

3.1 НСМ распознавания сетевых кибератак с помощью экспертных знаний. Программный код MPNNpro, листинг которого показан в приложении Г, написан на языке программирования Python. Программа ориентирована на управление из командной строки, поэтому не имеет графического интерфейса. Также отметим, что особенностью НСМ, которая реализуется MPNNpro является возможность обучения с помощью продукционных правил. Пример такого правила для распознавания кибератаки типа neptune имеет вид:

Если duration = 0 ∧ protocol_type = tcp ∧ service= private mnet sql_net ∧ flag =SF S0 ∧ src_bytes= 0 ∧ dst_bytes= 0 ∧ land = 0 ∧ wrong_fragment = 0 ∧ urgent = 0 ∧ hot = 0 ∧ num_failed_logins = 0 ∧ logged_in = 0 ∧ num_compromised = 0 ∧ root_shell = 0 ∧ su_attempted= 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login= 0 ∧ is_guest_login = 0 ∧ count= от 6 до 141 ∧ srv_count = от 1 до 19 ∧ serror_rate = 1 ∧ srv_serror_rate = 1 ∧ rerror_rate = 0 ∧ srv_rerror_rate= 0 ∧ same_srv_rate = від 0.01 до 1 ∧ diff_srv_rate = от 0.05 до 0.33 ∧ srv_diff_host_rate = 0 ∧ dst_host_count= от 1 до 131 ∧ dst_host_srv_count = от 1 до 19 ∧ dst_host_same_srv_rate = от 0.01 до 1 ∧ dst_host_diff_srv_rate = от 0 до 0.27 ∧ dst_host_same_src_port_rate = от 0.01 до 1 ∧ dst_host_srv_diff_host_rate = от 0 до 0.33 ∧ dst_host_serror_rate = 1 ∧ dst_host_srv_serror_rate = 1 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Файл с тестовой выборкой должен находится в каталоге с программой и иметь название kddcup.csv. Результаты распознавания записывается в текстовый файл result.txt, а также могут быть отображены в виде графиков.

Программа kddNeural написана на языке программирования C#. Соответствующий листинг показан в приложении Д. Программа ориентирована на обучение НСМ на базе PNN с помощью данных БД KDD-99. Для распознавания также необходимо подать на вход программы данные в формате указанной БД. При этом в соответствии с разработанными НМС распознавания сетевых кибератак с помощью экспертных знаний kddNeural позволяет:

- Классифицировать сетевые запросы по типу – нормальный или кибератака.
- Распознавать классы сетевых запросов – normal, u2r, probe, r2l, dos.
- Распознавать следующие типы сетевых кибератак – back, buffer_overflow, ftp_write, guess_passwd, imap,ipsweep, land, loadmodule, multihop, neptune, nmap, normal, perl, phf, pod, portsweep, rootkit, satan, smurf, spy, teardrop, warezclient, warezmaster.

Главное окно программы показано на рисунке 4.7.

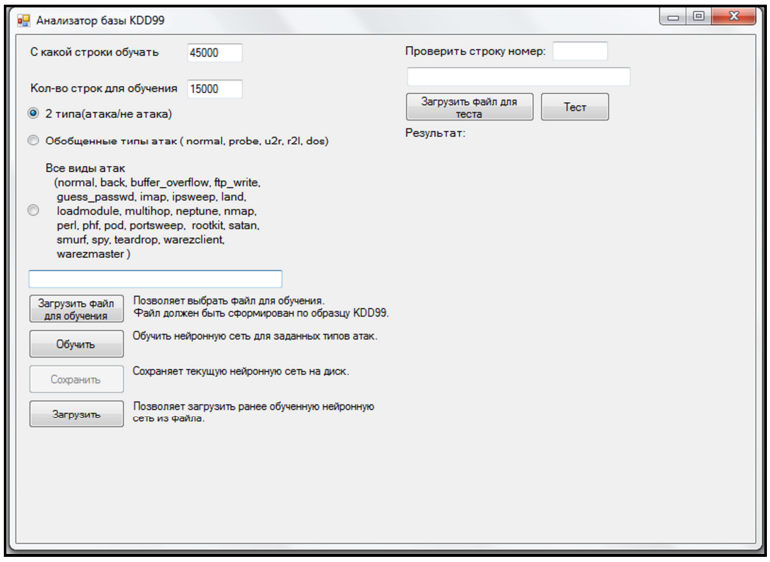

Рисунок 4.7 - Главное окно программы kddNeural

В соответствии с архитектурой PNN, функционал kddNeural предусматривает два режима эксплуатации: обучение и распознавание. Также предусмотрена сервисная возможность сохранения, а в дальнейшем и использования параметров обученной НСМ. Также с целью повышения гибкости проведения экспериментов программа kddNeural позволяет обучать НСМ на определенной части файла, содержащего обучающую выборку.

Результатом работы kddNeural в режиме распознавания являются вероятности принадлежности анализируемого сетевого запроса к каждому из заранее определенных классов. Для примера на рисунке 4.8 отображен результат распознавания программой kddNeural сетевого запроса как одного из классов: normal, u2r, probe, r2l, dos. На рисунке 4.9 показаны вероятности принадлежности этого же запроса к классам, отображающим типы сетевых кибератак.

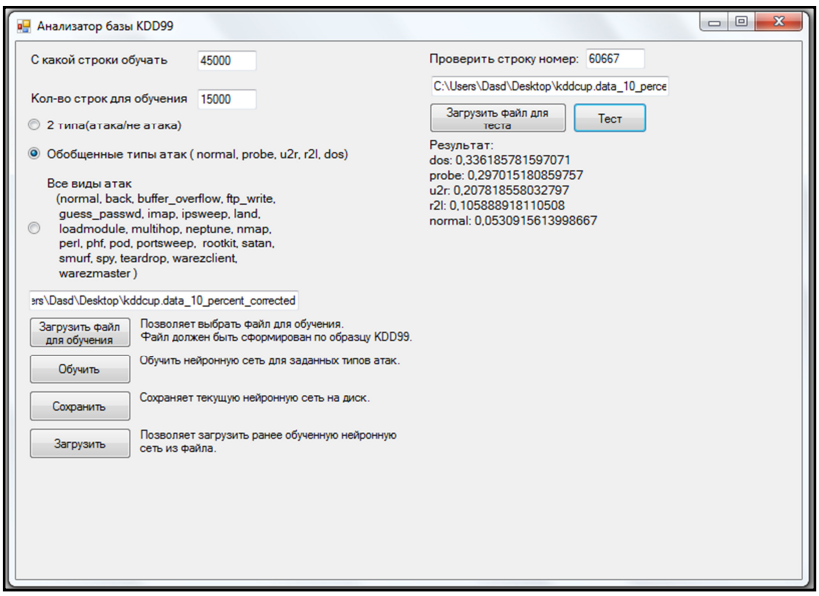

Рисунок 4.8 Результат распознавания класса сетевого запроса

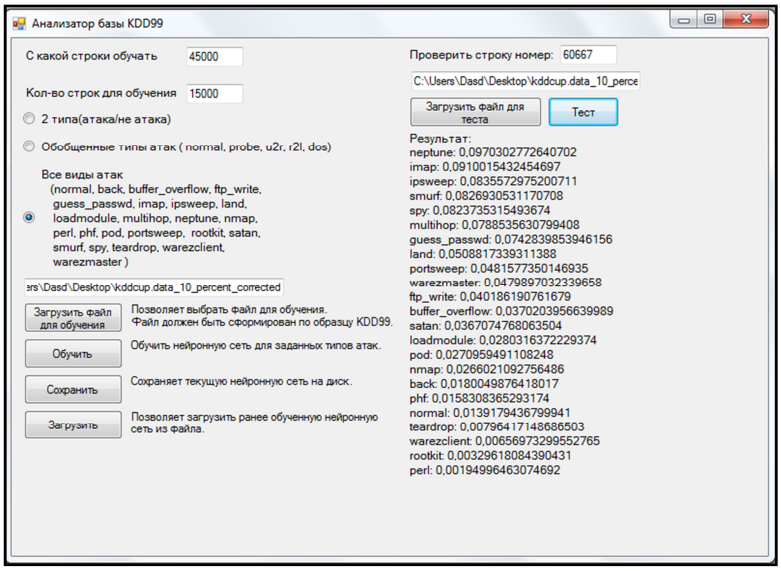

Рисунок 4.9 Результат распознавания типа сетевой кибератаки

4.3 Экспериментальные исследования

Главной целью проведения экспериментальных исследований было подтверждения достоверности основных результатов диссертационного исследования. Для этого проверялись гипотезы о том, что использование разработанных методов и системы позволит обеспечить эффективное распознавание сетевых кибератак. Проведенные исследования разделены на три этапа.

На первом этапе для проверки эффективности разработанного метода создания обучающей выборки были проведены эксперименты с целью обучить НСМ типа ДСП классифицировать сетевые запросы на normal, buffer_overflow, loadmodule, perl и rootkit [80].

ДСП обучался как на примерах обучающей выборки вида $\boldsymbol{Z}_1$, которая была создана с помощью предложенного метода, так и с помощью обучающей выборки вида $\boldsymbol{Z}_g$, что была сформирована в соответствии с общеизвестными методами [50].

Основным отличием учебных примеров, которые входили в состав множества $\boldsymbol{Z}_g$, стало кодирование ожидаемого выходного сигнала НСМ в соответствии алфавитному ранжированию названий распознаваемых классов.

Эксперименты проводились с помощью программного комплекса NeuroPro. Результаты экспериментов показали, что для достижения допустимой средней ошибки обучения $\varepsilon \approx 0{,}01$ при использовании выборки вида $\boldsymbol{Z}_1$ необходимое количество учебных итераций приблизительно в 2,4 раза меньше, чем при использовании выборки вида $\boldsymbol{Z}_g$. Это подтверждает справедливость сформулированной гипотезы о эффективности предложенного метода создания обучающей выборки НСМ для распознавания сетевых кибератак.

На втором этапе исследований была проверена гипотеза о том, что использование разработанного метода нейросетевого распознавания сетевых кибератак позволит создать на его основе НСС, эффективность которой выше,

чем у известных систем. Для проверки указанной гипотезы использовано разработанный в [6] метод оценки эффективности применения НСР распознавания кибератак. Математическое обеспечение этого метода составляют выражения (1.6, 1.7).

Основные результаты исследований представлены в таблице 4.4. В процессе исследований приняты величины коэффициентов значимости параметров эффективности $\boldsymbol{A}^{nz} = \{0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 0.7, 1\}$. Отметим, что увеличение коэффициента значимости для $E_{квс}$ объясняется тем, что выполнение соответствующей процедуры (использование различных видов обучающей выборки) определяет принципиальную возможность применения НСР для распознавания сетевых кибератак.

Анализ данных таблицы 4.4 позволяет утверждать, что использование предложенного метода позволяет в 1,35 раз повысить эффективность применения НСР распознавания сетевых кибератак по отношению к наилучшим известным НСР. Также определено, что основным направлением усовершенствования созданной НСС ПКА является оптимизация метода обучения.

Таблица 4.4 - Оценка эффективности НСР распознавания сетевых кибератак

Метод	Параметр									
	$E_{по}$	$E_{ота}$	$E_{опа}$	$E_{омн}$	$E_{вэп}$	$E_{пна}$	$E_{одв}$	$E_{ов}$	$E_{квс}$	E_{Σ}
АППТ	1	0	0	0	0	0	0	0	0	0,7
НСОВ	0	1	0	0	0	0	0	0	0	0,7
ТОСА	1	1	0	0	0	0	0	0	0	1,4
РАСТ	0	1	1	0	0	0	0	0	0	1,4
ВСА	0	1	1	0	0	0	0	0	0	1,4
ПСКТ	1	0	0	0	0	0	0	0	0	0,7
ПВСА	1	1	0	1	0	0	0	0	0	2,1
АСОА	1	1	1	0	0	0	0	0	0	2,1
СОД	0	1	0	1	0	0	0	0	0	1,4
БНМ	0	1	0	1	0	0	0	1	0	1,4
ОКСА	1	0	0	0	0	0	0	1	0	1,4
МОВ	1	0	0	0	0	0	0	0	0	0,7
НСОК	1	0	0	0	0	0	0	0	0	0,7
НСГС	1	0	0	0	0	0	0	1	0	1,4
СОСА	1	0	0	0	0	0	0	1	0	1,4
НМОПБ	1	1	1	1	1	1	1	0	0	4,9
СОВНС	1	0	1	0	0	0	0	0	0	1,4
СОАСТ	1	1	1	1	0	0	0	0	0	2,8
НСМРЧ	1	0	0	1	0	0	0	0	0	1,4
НССАСТ	1	0	1	1	0	0	0	0	0	2,1
СОВГНС	1	0	1	1	0	1	0	0	0	2,8
НСС ПКА	1	1	1	1	1	1	1	1	1	6,6

Третий этап исследований был нацелен на проверку гипотезы о адекватности разработанной НСС ПКА к вариативности ожидаемых условий использования. Адекватность оценивалась с помощью соответствия величины ошибки распознавания кибератак установленным требованиям. Вариативность условий использования НСС ПКА определялась используемой обучающей выборки, вид которой также определял вид наиболее эффективной НСМ.

Вначале рассмотрен наиболее вероятный на практике вариант использования обучающей выборки вида $\boldsymbol{Z}_4$. Отметим, что в такой выборке около 20% учебных примеров являются маркированными, то есть содержат значение ожидаемого выходного сигнала. Указанное значение может быть определено, например, экспертным путем, или взято из БД кибератак. 80% учебных примеров такой выборки значения ожидаемого выходного сигнал не содержат. Наиболее эффективно использовать обучающую выборку вида $\boldsymbol{Z}_4$ для распознавания сетевых кибератак могут, разработанные в третьем разделе, НСМ типа ГНС. Указанная модель обучена с использованием учебных примеров БД NSL-KDD. Базируясь на результатах [3], в соответствии со структурой и объемом БД использована следующая структура ГНС: количество входных нейронов – 40, количество скрытых слоев - 2, количество нейронов в каждом скрытом слое – 300, количество выходных нейронов 4. Для моделирования использована разработанная программа kddNeural. После обучения, разработанная НСМ использована для распознавания примеров, которые не применялись для обучения. Тестовые выборки содержали примеры, описывающие сетевые соединения на протяжении 24 часов. Результаты распознавания показаны на рисунке 4.6 и рисунке 4.10.

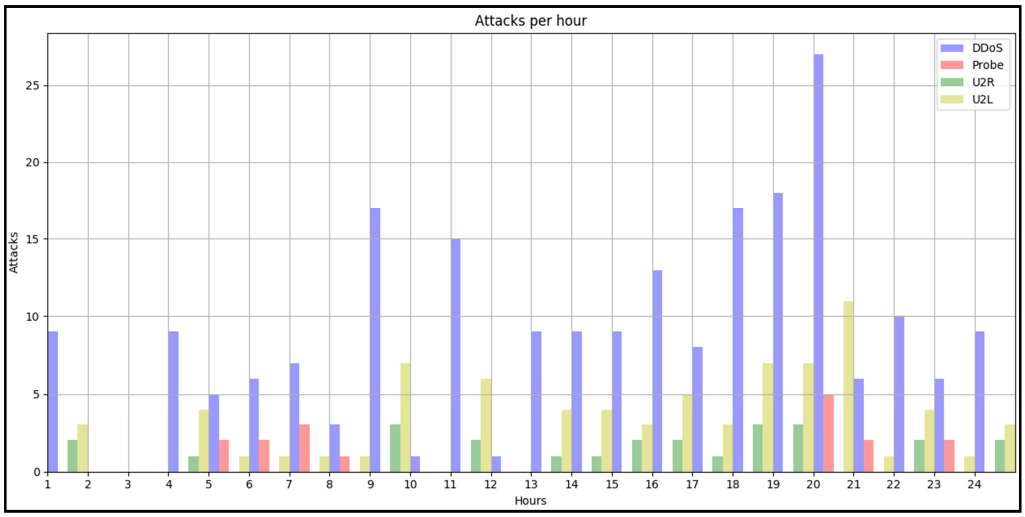

Рисунок 4.10 – Результаты распознавания ГНС тестовой выборки

В среднем точность распознавания ГНС около 90%, что соответствует точности распознавания кибератак с помощью известных СРК [77].

Рассмотрен вариант использования обучающей выборки вида $\boldsymbol{Z}_2$. Отметим, что такой вид обучающей выборки характерен для случая ограниченной статистики. Принята необходимость распознавания сетевых

кибератак типа neptune, smurf, pod, teardrop, land, back, buffer_overflow, loadmodule, perl и rootkit. С использованием данных БД KDD-99, разработано 10 соответствующих продукционных правил. Пример правила для распознавания кибератаки типа smurf имеет вид:

Если duration = 0 ∧ protocol_type imcp ∧ service = ecr_i ∧ flag = SF ∧ src_bytes = 1032 ∧ dst_bytes = 0 ∧ land – 0 ∧ wrong_fragment – 0 ∧ urgent – 0 ∧ hot = 0 ∧ num_failed_logins = 0 ∧ logged_in = 0 ∧ num_compromised = 0 ∧ root_shell = 0 ∧ su_attempted = 0 ∧ num_root = 0 ∧ num_file_creations = 0 ∧ num_shells = 0 ∧ num_access_files = 0 ∧ num_outbound_cmds = 0 ∧ is_host_login = 0 ∧ is_guest_login = 0 ∧ count = от 300 до 500 ∧ srv_count = от 300 до 500 ∧ serror_rate = 0 ∧ srv_serror_rate = 0 ∧ rerror_rate = 0 ∧ srv_rerror_rate = 0 ∧ same_srv_rate = 1 ∧ diff_srv_rate = 0 ∧ srv_diff_host_rate = 0 ∧ dst_host_count = 255 ∧ dst_host_srv_count = 255 ∧ dst_host_same_srv_rate = 1 ∧ dst_host_diff_srv_rate = 0 ∧ dst_host_same_src_port_rate 1 ∧ dst_host_srv_diff_host_rate 0 ∧ dst_host_serror_rate = 0 ∧ dst_host_srv_serror_rate = 0 ∧ dst_host_rerror_rate = 0 ∧ dst_host_srv_rerror_rate = 0.

Для распознавания использована, разработанная НСМ распознавания сетевых кибератак с помощью экспертных знаний. Эксперименты, направленные на распознавание сетевых кибератак типа smurf, проводились при помощи разработанной программы MPNNpro. Отметим, что ошибка распознавания тестовых примеров для кибератак этого типа менее 0,04, что соответствует лучшим аналогам [97].

Также рассмотрен вариант использования учебной выборки вида $\boldsymbol{Z}_1$, объем которой не позволяет использовать для распознавания НСМ типа МСП. В соответствии с предложенным методом нейросетевого распознавания кибератак использована разработанная НСМ типа PNN. Обучение такой НСМ реализуется с помощью продукционных правил вида (3.1), каждое из которых является аналогом отдельной записи БД KDD-99 или NSL-KDD. Для экспериментов использована разработанная программа kddNeural. Обучение проводилось на тренировочной части данных NSL-KDD. Для распознавания использованы тестовые данные NSL-KDD, а также статистика сетевых запросов зарегистрированных с помощью программы Win Sniffer. Промежуточные результаты тестирования частично показаны на рисунке 4.8 и рисунке 4.9. Средняя ошибка распознавания обобщенных классов кибератак составила менее 0.03.

Таким образом, результаты проведенных исследований подтверждают возможность повышения эффективности СРК за счет использования в них разработанных нейросетевыых моделей и методов.

Данный раздел посвящен решению научно-практических задач разработки нейросетевой системы распознавания сетевых кибератак и проведения экспериментальных исследований, направленных на подтверждение достоверности основных результатов диссертационной работы. Основные результаты раздела:

– Впервые разработана архитектура нейросетевой системы распознавания сетевых кибератак, в которой в отличии от известных предусмотрено использование подсистем определения условий создания нейросетевых средств, формирования адаптивной обучающей выборки и разработки нейросетевых моделей, что обеспечивает достаточную точность распознавания и адаптацию к условиям разработки и применения.
– Разработана экспериментальная установка, которая обеспечивает возможность проведения экспериментов, направленных на проверку достоверности основных результатов диссертационной работы.

Проведенные исследования позволяют утверждать:
– Использование разработанного метода создания обучающей выборки позволяет примерно в 2,4 раза уменьшить количество вычислительных операций нейросетевой модели для достижения допустимой ошибки обучения.
– Использование разработанного метода нейросетевого распознавания кибератак позволяет приблизительно в 1,35 раз повысить эффективность нейросетевых средств распознавания сетевых кибератак.
– При ожидаемых условиях применения разработанная нейросетевая система позволит обеспечить ошибку распознавания сетевых кибератак в пределах 0,05, что достаточно для практического использования.

ЗАКЛЮЧЕНИЕ

Краткие выводы по результатам диссертационных исследований.

В диссертационной работе решена актуальная научно-прикладная задача разработки эффективных нейросетевых моделей, методов и средств распознавания кибератак, адаптированных к условиям эксплуатации и способных оперативно распознавать новые виды сетевых кибератак.

Проведенные исследования позволяют сформулировать следующие выводы:

1) В результате анализа научно-практических работ посвященных разработке и эксплуатации систем распознавания кибератак на сетевые ресурсы информационных систем общего назначения показано, что одним из основных путей их развития указанных систем является внедрение методов анализа сетевого трафика, базирующихся на современных решениях теории искусственных нейронных сетей. Для этого необходимо развить методологическую базу нейросетевого распознавания сетевых кибератак и разработать на этой базе метод создания обучающей выборки и метод создания, соответствующих нейросетевых средств. Для апробации предложенных решений целесообразно разработать нейросетевую систему и провести исследование ее эффективности.

2) Получила дальнейшее развитие методологическая база нейросетевого распознавания кибератак на сетевые ресурсы информационных систем, которая за счет учета условий создания таких средств, обеспечила возможность создания эффективных нейросетевых моделей и методов распознавания.

3) Получили дальнейшее развитие нейросетевые модели распознавания, которые за счет возможности обучения с помощью экспертных данных и использования комбинированной обучающей выборки, позволяют оперативно реагировать на новые типы сетевых кибератак.

4) Впервые разработан метод создания обучающей выборки для нейросетевого распознавания сетевых кибератак, который за счет определения параметров допустимых видов выборки и учета в выходном сигнале близости эталонов видов кибератак, позволяет определить круг допустимых видов нейросетевых моделей и обеспечить уменьшение количества учебных итераций приблизительно в 2,4 раза.

5) Впервые разработан метод нейросетевого распознавания кибератак на сетевые ресурсы информационных систем, который за счет использования разработанных нейросетевых моделей и разработанного метода создания обучающей выборки, позволяет расширить функциональные возможности и обеспечить достаточную точность распознавания.

6) С использованием предложенных моделей и методов, разработана и исследована нейросетевая система распознавания кибератак на сетевые ресурсы информационных систем. В отличии от известных нейросетевых средств, в данной предусмотрено использование подсистем определения условий создания нейросетевых средств, формирования адаптивной обучающей выборки и разработки нейросетевых моделей, что обеспечивает достаточную

точность распознавания и адаптацию к условиям разработки и применения. Экспериментальным путем показано, что эффективность предложенной нейросетевой системы приблизительно в 1,35 раз выше по отношению к наилучшим подобным системам, а при ожидаемых условиях применения ее применение позволит обеспечить ошибку распознавания сетевых кибератак в пределах 0,05, что достаточно для практического использования. Также показано, что использование разработанного метода создания обучающей выборки позволяет примерно в 2,4 раза уменьшить количество вычислительных операций нейросетевой модели для достижения допустимой ошибки обучения

7) Указанные результаты внедрены в учебный процесс на кафедре безопасности информационных технологий Национального авиационного университета (Киев, Украина) и на кафедре информационные системы Западно-Казахстанского аграрно-технического университета имени Жангир хана (акт внедрения от 04.09.2017).

Оценка полноты решений поставленных задач.

В результате выполнения диссертационных исследований все поставленные задачи решены в полном объеме:

– Проведен анализ возможностей нейросетевых средств распознавания кибератак на сетевые ресурсы информационных;

– Реализовано развитие методологической базы нейросетевого распознавания кибератак на сетевые ресурсы информационных систем;

– Разработаны нейросетевые модели и методы распознавания сетевых кибератак;

– Разработана нейросетевая система распознавания сетевых кибератак и проведены экспериментальные исследования, направленные на верификацию предложенных диссертационных решений.

Рекомендации и исходные данные по конкретному использованию результатов.

Результаты научных исследований могут быть использованы при разработке новых и модернизации существующих средств противодействия кибератакам на сетевые ресурсы информационных систем как общего назначения, так и на сетевые ресурсы объектов критической инфраструктуры. Это позволит повысить эффективность средств противодействия за счет более точного распознавания фактов реализации кибератак. В качестве исходных данных для конкретного применения могут использоваться требования к системе распознавания кибератак, характеристики программно-аппаратного обеспечения системы распознавания кибератак и ресурсы, выделяемые на ее разработку. Кроме этого, полученные результаты могут использоваться при подготовке специалистов, которые занимаются разработкой и эксплуатацией средств противодействия сетевым кибератакам.

Оценка технико-экономической эффективности внедрения.

Полученные в диссертационной работе результаты исследований могут принести существенный технико-экономический эффект при разработке отечественных комплексных систем защиты информации. Базируясь на

приведенных в диссертационной работе результатах экспериментов можно обосновано предположить двукратное уменьшение затрат на аппаратное обеспечение контура распознавания сетевых кибератак.

Оценка научного уровня выполненной работы в сравнении с лучшими достижениями в данной области.

Оценка научного уровня выполненной работы в сравнении с лучшими достижениями в данной области проведена на основании анализа научно-практических литературных источников посвященных тематике «Нейросетевые модели и методы противодействия атакам на сетевые ресурсы информационных систем». Выбор индекса классификации и глубина поиска 10 лет соответствующие теме обеспечили надежность и достоверность поиска актуальных информационных материалов. В результате проведенного анализа определено, что научный уровень выполненной диссертационной работы обладает достаточной новизной и в целом соответствует мировому техническому уровню и тенденциям развития технологий распознавания кибератак на сетевые ресурсы информационных систем.

Автор выражает глубокую благодарность сотрудникам Национального авиационного университета, профессорско-преподавательскому составу кафедры «Кибербезопасность, обработка и хранение информации» КазНИТУ имени К.И. Сатпаева за консультации и техническую помощь в выполнении экспериментов и анализов по настоящей диссертационной работе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Концепция кибербезопасности («Киберщит Казахстана»): утв. Постановлением Правительства Республики Казахстан 30 июня 2017 года, № 407 [Электронный ресурс] // http://mdai.gov.kz/ru/pages/ koncepciya-kiberbezopasnosti-kibershchit-kazahstana.23.01.2018.

2 Абрамов Е.С. Разработка и исследование методов построения систем обнаружения атак: дис. … канд. техн. наук: 05.13.19. – Таганрог, 2005. – 199 с.

3 Айтчанов Б.Х., Бапиев И.М. Разработка процедуры определения ожидаемого выходного сигнала нейросетевой модели распознавания кибератак // Международный журнал прикладных и фундаментальных исследований. – М.: Академия естествознания, 2017. – Ч. 1. – № 5. – С. 8 – 11.

4 Аль-Мехди С.Т., Евланенкова О. Применение нейронных сетей для обнаружения вторжений // Доклады ТУСУР. – 2014. – №4. – С. 28-33.

5 Архипов А., Ишутин А. Применение моделей обнаружения аномалий для выявления атак // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні : наук.-техн. конф., 1-3 берез. 2006 р. : тези доп. – К., 2006. – С. 71-72.

6 Ахметов Б.Б., Корченко А.Г., Терейковский И.А. Параметры оценки эффективности нейросетевых средств распознавания кибератак на сетевые ресурсы информационных систем // Reports of the national academy of sciences of the republic of Kazakhstan. – 2017. – Volume 2. – Number 312. – ISSN 2224-5227.

7 Бапиев И.М., Ахметов Б.С., Корченко А.Г. Применение нейронной сети с радиальными базисными функциями для распознавания скриптовых вирусов // II международная научно-практическая конференция «Актуальные вопросы обеспечения кибербезопасности и защиты информации». – Киев: Европейский университет, 2016. – С. 21-24.

8 Бапиев И.М., Корченко А.Г., Терейковский И.А. Разработка критериев оценки эффективности нейросетевых средств распознавания кибератак на сетевые ресурсы информационных систем // IV International scientific conference «Global and regional problems of informatizition in society and nature using». – Kyiv: National University of Life and Environmental Sciences of Ukraine. – 2016. – P. 80-82.

9 Бапиев И.М. Правила для определения эффективных видов нейросетевых моделей распознавания кибератак на сетевые ресурсы // III Международная научно-практическая конференция «Актуальные вопросы обеспечения кибербезопасности и защиты информации». – Киев: Европейский университет, 2017. – С. 27-30.

10 Бапиев И.М., Терейковский О.И. Исследование продукционных правил для обучения нейросетевых средств распознавания кибератак // Стан та удосконалення безпеки інформаційно-телекомунікаційних систем (SITS’2017). – Миколаїв - Коблево: МТУ «Миколаївська політехніка», 2017, – С. 8-9.

11 Безобразов С.В., Головко В.А. Алгоритмы исскуственных имунных систем и нейронних сетей для обнаружения вредоносных программ // Нейроинформатика. – 2010. – №7. – С. 273-288.

12 Беляев А. Петренко С. Системы обнаружения аномалий: новые идеи в защите информации // Экспресс-Электроника. – 2004. – № 2. – С. 12–14.

13 Большев А.К. Алгоритмы преобразования и классификации трафика для обнаружения вторжений в компьютерные сети: авторефер. … канд. техн. наук: 05.13.19 – Методы и системы защиты информации, информационная безопасность. – Санкт-Петербург, 2011. – 36 с.

14 Браницкий А.А. Котенко И.В. Обнаружение сетевых атак на основе комплексирования нейронных, иммунных и нейронечетких классификаторов // Информационно-управляющие системы. – 2015. – №3. – С. 69-77.

15 Васильев В.И., Хафизов А.Ф. Нейронные сети при обнаружении атак в сети Internet (на примере атаки SYNFLOOD) // Нейрокомпьютеры в информационных и экспертных системах. – М.: Радиотехника, 2007. – №6. – С. 34-38.

16 Вилков А.С. Информационная безопасность персональных ЭВМ и мониторинг компьютерных сетей. – М.: МИНИТ ФСБ России, 2005. – 210 с.

17 Галушкин А.И. Теория нейронных сетей. – М.: ИПРЖР, 2000. – 416 с.

18 Горбань А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. – Новосибирск: Наука, 1996. – 276 с.

19 Гришин А.В. Нейросетевые технологии в задачах обнаружения компьютерных атак // Информационные технологии и вычислительные системы. – 2011. – №1. – С. 53 -64.

20 Довлад О.А. Дослідження та розробка моделі процесу атаки та трафіку локальної мережі // Захист інформації. – 2009. – № 1 – С. 83–86.

21 Ежов А.А., Шумский С.А. Нейрокомпьютинг и его применения в экономике и бизнесе. – М.: МИФИ, 1998. – 224 с.

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

23 Жульков Е. Поиск уязвимостей в современных системах IDS // Открытые системы. – 2003. – № 7–8. – С. 16–18.

24 Заенцев И.В. Нейронные сети: основные модели. – Воронеж: Воронежский гос. ун-т, 1999. – 76 с.

25 Закер К. Компьютерные сети / пер. с англ. – СПб.: БХВ-Петербург, 2000. – 1008 с.

26 Зима В.М., Молдовян А.А., Молдовян Н.А. Безопасность глобальных сетевых технологий. – СПб.: БХВ-Петербург, 2000. – 450 с.

27 Зиновьев А.Ю., Питенко А.А. Визуализаци данных методом упругих карт // Радіоелектроніка. Інформатика. Управління. – 2000. – № 1. – С. 76–85.

28 Иванов А.И. Быстрое обучение искусственных нейронных сетей в системах биометрической аутентификации личности: авторефер. дисс. … док. техн. наук : 05.13. 01. – Пенза, 2000. – 36 с.

29 Каллан Р. Основные концепции нейронных сетей / пер. с англ.; под ред. А.Г. Сивака. – М.: Вильямс, 2003. – 288 с.

30 Касперски К. Техника и философия хакерских атак. – М.: Солон, 2001. – 256 с.

31 Комар М.П. Метод построения совокупного классификатора трафика информационно-телекоммуникационных сетей для иерархической классификации компьютерных атак // Системи обробки інформації. – 2012. – Випуск 3 (101) . – Том 1 – С.134-138.

32 Комар М.П., Палий И.О., Шевчук Р.П. Нейросетевой подход к обнаружению сетевых атак на компьютерные системы // Інформатика та математичні методи в моделюванні. – 2011. – Том 1. – №2. – С. 156-160.

33 Корченко А.Г., Терейковский И.А., Карпинский Н., Тынымбаев С. Нейросетевые модели, методы и средства оценки параметров безопасности интернет-ориентированных информационных систем: монография. - К.: ТОО «Наш Формат». – 2016. – 275 с.

34 Корченко А.А., Стасюк А.И. Базовая модель параметров для построения систем выявления атак // Захист інформації. – 2012. – № 2 (55). – С. 47-51.

35 Корченко О.Г., Терейковський І.А., Казмірчук С.В. Верифікація нейромережевих методів розпізнавання кібератак // Науково-технічний збірник «Управління розвитком складних систем» Київського національного університету будівництва і архітектури. – 2014. – Випуск 17. – С. 168-172.

36 Корченко О.Г. Системи захисту інформації: монографія. – К.: НАУ, 2004. – 264 с.

37 Корченко А.Г., Терейковский И.А., Терейковская Л.А., Ахметов Б.Б. Определение эффективных видов нейросетевых моделей распознавания кибератак на сетевые ресурсы // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2016. – Вип. 2 (32).

38 Крамер Г. Математические методы статистики / пер. с англ.; под ред. А.С. Монина. – М.: Мир, 1976. – 648 с.

39 Красоткин А. Обнаружение сетевых атак // Мир ПК. – 2003. – № 6. – С. 24–26.

40 Крыжановский А.В. Применение искусственных нейронных сетей в системах обнаружения атак [Электронный ресурс] // Доклады ТУСУРа. – 2008. – № 2 (18). – Часть 1. – С. 37-41. https://cyber leninka.ru/article/n/primenenie-iskusstvennyh-neyronnyh-setey-v-sis temah-obnaru zheniya-atak. 19.12.2017.

41 Круглов В.В., Дли М.И., Голунов Р.Ю. Нечеткая логика и искусственные нейронные сети. – М.: Горячая линия-Телеком, 2004. – 242 с.

42 Кузнецов Г.В., Иванов А.М. Классификация и анализ систем и методов обнаружения атак // Захист інформації. – 2004. – № 4 – С. 4–11.

43 Кузнецов Г.В., Иванов А.М. Методы анализа данных для обнаружения атак в компьтерных сетях банковских структур // Защита информации: сб. науч. трудов. – К.: НАУ, 2004. – С. 45–50.

44 Лукацкий А.В. Обнаружение атак. – СПб.: БХВ-Петербург, 2003. – 624 с.

45 Люгер Ф. Искусственный интеллект: стратегии и методы решения сложных проблем, 4-е издание / пер. с англ.; под ред. Н.И. Галагана. – М.: Вильямс, 2003. – 864 с.

46 Магницкий Ю.Н. Использование бинарной нейронной сети для обнаружения атак на ресурсы распределенных информационных систем // Динамика неоднородных систем. – 2008. – С. 200-205.

47 Міхайленко В.М., Терейковська Л.О., Терейковський І.А., Ахметов Б.Б. Нейромережеві моделі та методи розпізнавання фонем в голосовому сигналі в системі дистанційного навчання: монографія. – К.: ЦП «Компринтр», 2017. – 252 с.

48 Мелкумян К.В. COM как средство для реализации достоверной вычислительной базы // Защита информации: Сб. науч. трудов. – К.: КМУГА, 1999. – С. 104–106.

49 Менаске Д., Виргилио А. Производительность Web-служб. Анализ, оценка и планирование / пер. с англ. – СПб.: ДиаСофтЮп, 2003. – 480 с.

50 Мустафаев А.Г. Нейросетевая система обнаружения компьютерных атак на основе анализа сетевого трафика [Электронный ресурс] // Вопросы безопасности. – 2016. – № 2. – С.1-7. DOI: 10.7256/2409-7543.2016.2.18834. http://e-notabene.ru/nb/article_18834.html. 19.12.2017.

51 Нейман Дж. Теория самовоспроизводящихся автоматов / пер. с англ. – М.: Мир, 1971. – 384 с.

52 Нейман Дж., Моргенштерн О. Теория игр и экономическое поведение / пер. с англ. – М.: Наука, 1970. – 326 с.

53 Нижник Е.И. Математическое моделирование производительности файловых систем: автореф. дис. … канд. техн. наук: 05.13.18. – М., 2007. – 24 с.

54 Новак Дж., Норткатт О., Маклахен Д. Как обнаружить вторжение в сеть. Настольная книга специалиста по системному анализу / пер. с англ.; под ред. И. Дранишникова. – М.: Лори, 2012. – 384 с.

55 Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344 с.

56 Паркер Т., Сиян К. TCP/IP для профессионалов / пер. с англ.; под ред. Е. Матвеева. – СПб.: Питер, 2004. – 859 с.

57 Петров А.С., Украинский А.П. Перспективы применения искусственных нейронных сетей в компьютерных системах // Системи обробки інформації. – 2010. – №3 (84). – С. 72-74.

58 Погорелов В.В., Бапиев И.М., Терейковский О.И. Современные нейросетевые средства распознавания кибератак на ресурсы компьютерных сетей // V International scientific conference «Global and regional problems of informatizition in society and nature using'2017». – Kyiv: NULES of Ukraine, 2017. – С. 68-69.

59 Поликарпов С.В., Дергачев В.С., Румянцев К.Е., Голубчиков Д.М. Новая модель искусственного нейрона: кибернейрон и области его применения [Электронный ресурс]. http://arxiv.org/ftp/arxiv/papers /0907/0907.0229.pdf. 19.12.2017.

60 Пучков Н.В. Использование искусственных нейронных сетей для контроля корректности информационно-технологического процесса // Новые промышленные технологи. – 1999. – № 3. – С. 79–84.

61 Рассел С., Норвиг П. Искусственный интеллект: современный подход, 2-е изд / пер.с англ.; под ред. К.А. Птицына. – М.: Вильямс, 2007. – 1408 с.

62 Резник А.М. О природе интеллекта // Математические машины и системы. – 2008. – №1. – С. 23-45.

63 Розенблат Ф. Аналитические методы изучения нейронных сетей / пер. с англ. – М.: Зарубежная радиоэлектроника, 1965. – 150 с.

64 Руденко О.Г., Бодянський Є.В. Штучні нейронні мережі: навч. посіб. – Харків: ТОВ "Компанія СМІТ", 2006. – 404 с.

65 Свинцов В.И. Смысловой анализ и обработка текста. – М.: Книга, 1979. – 272 с.

66 Сенешова М.Ю. Погрешности нейронных сетей. Вычисление погрешностей весов синапсов // Методы нейроинформатики: сб. науч. трудов. – Красноярск: КГТУ. – 1998. – С. 204 – 212.

67 Слеповичев И.И., Ирматов П.В., Комарова М.С., Бежин А.А. Обнаружение DDoS-атак нечеткой нейронной сетью // Известия Саратовского университета. Серия «Математика. Механика. Информатика». – 2009. – Т. 9. – Вып. 3. – С. 84-89.

68 Стаханов А.А. Linux. – СПб.: БХВ-Петербург, 2004. – 912 с.

69 Сухов В.Е. Система обнаружения аномалий сетевого трафика на основе искусственных иммунных систем и нейросетевых детекторов // Вестник РГРТУ. – 2015. – № 54. – Ч.1. – С. 84-90.

70 Талалаев А.А., Тищенко И.П., Фраленко В.П., Хачумов В.М. Разработка нейросетевого модуля мониторинга аномальной сетевой активности // Нейрокомпьютеры: разработка и применение. – 2011. – № 7. – С. 32-38.

71 Таненбаум Э. Компьютерные сети / пер. с англ.; под ред. А. Леонтьева. – СПб.: Питер, 2002. – 848 с.

72 Тейлор Дж. Введение в теорию ошибок / Тейлор Дж.; пер. с англ. Л. Г. Деденко. – М. : Мир, 1985. – 272 с.

73 Терейковский И.А. Моделирование профилей нормального поведения компьютерных систем // Защита информации: сб. науч. трудов НАУ. – 2006. – Выпуск 13. – С. 103-108.

74 Терейковський І. Нейронні мережі в засобах захисту комп’ютерної інформації: монографія. – К.: ПоліграфКонсалтинг. – 2007. – 209 с.

75 Терейковський І.А. Оптимізація архітектури нейронної мережі, призначеної для діагностики стану комп’ютерної мереж // Науково-технічний збірник "Управління розвитком складних систем" Київського національного університету будівництва і архітектури. – 2011. – Випуск 6. – С. 155-158.

76 Терейковський І.А. Нейромережева методологія розпізнавання інтернет-орієнтованого шкідливого програмного забезпечення // Безпека інформації. – 2013. – Т. 19. – № 1. – С. 24-28.

77 Тимофеев А., Браницкий А. Исследование и моделирование нейросетевого метода обнаружения и классификации сетевых атак // International Journal Information Technologies & Knowledge. – 2012. – Vol.6. – Number 3. – P. 257-265.

78 Уэйнпрат П. Apache для профессионалов / пер. с англ.; под ред. И. Дранишников. – М.: Лори, 2001. – 473 с.

79 Уэнстром М. Организация защиты сетей Cisco / пер. с англ. – М.: Вильямс, 2005. – 768 с.

80 Федотов Е.В. Механизмы возможных атак в сети Internet // Защита информации: сб. науч. трудов. – К.: НАУ, 2001. – С. 30–42.

81 Хайкин С. Нейронные сети: полный курс, 2-е изд., испр. / пер. с англ.; под ред. Н.Н. Куссуль – М.: Вильямс, 2006. – 1104 с.

82 Хафизов А.Ф. Нейросетевая система обнаружения атак на WWW-сервер: дис. ... канд. техн. наук: 05.13.11. – Уфа, 2004. –172 с.

83 Царегородцев В.Г. Редукция размеров нейросети не приводит к повышению обобщающих способностей // Материалы XII Всеросс. семинара «Нейроинформатика и ее приложения». – Красноярск: КГТУ, 2004. – С. 163–165.

84 Царегородцев В.Г. Упрощение нейронных сетей – цели, идеи и методы // Нейрокомпьютеры: разработка, применение. – 2002. – № 4. – С. 5-13.

85 Частикова В.А., Власов К.А., Картамышев Д.А. Обнаружение ddos-атак на основе нейронных сетей с применением метода роя частиц в качестве алгоритма обучения // Фундаментальные исследования. – 2014. – № 8-4. – С. 829-832.

86 Щеглов А.Ю. Защита компьютерной информации от несанкционированного доступа // Наука и техника. – СПб.: 2004. – 384 с.

87 Aitchanov B., Bapiyev I., Terejkowski I. Calculation of expected output signal of neural network model for detecting of cyberattack on network resources // The 15th International Scientific Conference Information Technologies and Management. – Riga: ISMA University, April 27 - 28, 2017.

88 Basant S., Santosh B., Sushanta K. A Neural Network based system for Intrusion Detection and attack classification // IEEE Communication (NCC), 2016 Twenty Second National Conference, 4-6 March 2016.

89 Bezobrazov S., Golovko V. Neural Networks for Artificial Immune Systems: LVQ for Detectors Construction // International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications. – Dortmund, 2007. – P. 180-184.

90 Bivens A., Palagiri C., Smith R., Szymansky B. Network – Based Intrusion Detection Using Neural Networks // Proc. Intelligent Engineering Systems through Artificial Neural Networks ANNIE. – 2002. – St. Louis, MO. – Volume 12. – New York: ASME Press, 2002. – P. 579-584.

91 Chen Y., Narayanan A., Shaoning P., Ban T. Multiple sequence alignment and artificial neuralnetworks for malicious software detection // Natural Computation. – 2012. – P. 261-265.

92 Hnatiuk S. Cyberterrorism: History of current trends and countermeasures // Privacy Notice. – 2013. – Volume 9. – № 2. – P. 118-129.

93 Anderson J. Computer security threat monitoring and surveillance [Electronic resource] // Computer Security Resource Center of National Institute of Standards and Technology / Computer Security Laboratory Department of Computer Science University of California at Davis. – Electronic data. – Gaithersburg, MD, USA: NIST, 1980. – URL: http://csrc.nist.gov/publications /history/ande80.pdf – Language: English. – Description based on home page (viewed on 19.12.17).

94 Gavrilis D., Dermatas E. Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features // Computer Networks. – 2005. – № 48. – P. 235-245.

95 IBM Proventia Network Anomaly Detection System [Electronic resource] / IBM // IBM. – Electronic data. – [Armonk, New York, USA]: IBM, [2011]. –URL: http://www.ibm.com/ru/services/iss/proventia_network_anomaly_detection_system.html – Language: English. – Description based on home page (viewed on 19.12.17).

96 Koch R., Stelte B., Golling M. Attack Trends in Present Computer Networks // 4th International Conference on Cyber Conflict [CYCON 2012]. – Tallinn, Estonia, 2012. – 2012. – P. 225-236.

97 Korchenko O.G., Terejkowski I.A. Modern methods and neural network model parameter estimation of information systems security // Aviation in the XXI-st century. Safety in Aviation And Space.

98 Kotov V., Vasilyev V. Detection of web server attacks using principles of immunocomputing // Proc. of 2nd World Congress on Nature and Biologically Inspired Computing. – 2010. – P. 25-30.

99 Kang M., Kang J. Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security // PLoS One 2016; 11(6).

100 Planquart J.P. Application of neural networks to intrusion detection [Electronic resource] // SANS Information Security Reading Room. – Electronic data. – [USA]: SANS Institute, 2001. – URL: http://www.sans.org/reading_room/whitepapers/detection/application-neural-networks-intrusion-detection_336. – Language: English. – Description based on home page (viewed on 19.12.17).

101 Reznik A.M. "Non-Iterative Learning for Neural Networks" Proceedings // International Joint Conference on Neural Networks, Washington DC, July 10-16, 1999, №548.

102 Shyrochin V., Mukhin V. Adaptive security mechanisms for the computer networks based on risk analysis // Journal of Qafqaz University: AZN. Mathematics and Computer Science. – Num. 1. – Vol. 1. – 2013. – P. 11-16.

103 The Bro Network Security Monitor [Electronic resource] / The Bro Project. – Electronic data. – [USA]: The Bro Project, 2011. – URL: http://www.bro-ids.org/ – Language: English. – Description based on home page (viewed on 19.12.17).

104 The White House, Cyber space policy review. Assuring a Trusted and Resilient Information and Communications Infrastructure, 2010. – 76 p. http://msisac.cisecurity.org/awareness/documents/Cyberspace_Policy_Review_final.pdf. Description based on home page (viewed on 19.12.17).

105 Tripwire [Electronic resource] / [Tripwire, Inc.]. – Electronic data. – [Portland, OR, USA]: [Tripwire, Inc.], 2011. – URL: http://www.tripwire.org/ – Language: English. – Description based on home page (viewed on 19.12.17).

106 Krizhevsky A., Hinton G. Using very deep autoencoders for content-based image retrieval // Conf. ESANN 2011, 19th Europ. Sympos. on Artificial Neural Networks. – Bruges, Belgium. – 2011. – P. 44-51.

107 Kryzhevsky A. ImageNet Classification with Deep Convolutional Neural Networks. – NIPS, 2012.

ПРИЛОЖЕНИЕ А
Критерии эффективности вида НСМ

Критерий	Требование
R_1	Возможность использования нейронной сетью обучающих примеров с различным количеством входных параметров
R_2	Минимизация объема обучающей выборки
R_3	Возможность использования обучающей выборки, в которой количество примеров непропорционально количеству распознаваемых классов
R_4	Возможность использования обучающих примеров в которых отсутствует ожидаемых выходной сигнал
R_5	Возможность использования обучающей выборки, примеры которой коррелированны между собой
R_6	Приспособленность к дообучению без потери первоначальной учебной информации
R_7	Приспособленность к обучению отдельными частями
R_8	Обеспечение низкой ошибки обучения
R_9	Обеспечение короткого срока обучения
R_{10}	Обеспечение автоматического обучения
R_{11}	Минимизация объема вычислительных ресурсов при обучении
R_{12}	Обеспечение стабильности обучения
R_{13}	Возможность подачи в НСМ явных экспертных знаний.
R_{14}	Максимизация отношения объема памяти НСМ к количеству синаптических связей
R_{15}	Минимизация ошибки обобщения
R_{16}	Минимизация срока распознавания
R_{17}	Минимизация вычислительных ресурсов при распознавании
R_{18}	Возможность вербализации НСМ.
R_{19}	Апробированость в задачах распознавания кибератак

ПРИЛОЖЕНИЕ Б
Значения критериев эффективности для апробированных видов нейросетевых моделей

Критерий	Вид НСМ				
	МСП	ГНС	ТК	PNN	РБФ
1	2	3	4	5	6
R_1	0	0	0,2	0,2	0
R_2	0,2	0,2	0,5	0,9	0,7
R_3	0,3	0,5	0,5	0,5	0,7
R_4	0	0	1	0	0
R_5	0,8	0,8	0,3	0,5	0,5
R_6	0,2	0,9	0,2	1	0,7
R_7	0,1	0,9	0,1	0,1	0,9
R_8	0,9	0,9	0,5	0,9	0,9
R_9	0,5	0,5	0,8	0,9	0,9
R_{10}	0,9	0,9	0,7	0,9	0,7
R_{11}	0,7	0,2	0,8	0,9	0,9
R_{12}	0,8	0,8	0,7	0,9	0,9
R_{13}	0	0	0	0,9	0
R_{14}	0,9	0,9	0,4	0,3	0,5
R_{15}	0,9	0,9	0,4	0,3	0,4
R_{16}	0,7	0,7	0,4	0,9	0,5
R_{17}	0,9	0,9	0,4	0,5	0,5
R_{18}	0,5	0	0	0	0
R_{19}	0,9	0,9	0,5	0,5	0,5

ПРИЛОЖЕНИЕ В
Листинг программы для распознавания сетевых кибератак с помощью глубокой нейронной сети

Файл main.py

```
from model import Model
from view import View
from viewmodel import ViewModel

class Main:
    def __init__(self):
        model = Model()
        viewmodel = ViewModel(model)
        view = View(viewmodel)


if __name__ == '__main__':
    main = Main()
```

Файл main.spec

```
# -*- mode: python -*-

block_cipher = None


a = Analysis(['main.py'],
             pathex=['/home/mego_bonus/Projects/Diplom/Project/Network-
Intrusion-Detection'],
             binaries=[],
             datas=[],
             hiddenimports=[],
             hookspath=[],
             runtime_hooks=[],
             excludes=[],
             win_no_prefer_redirects=False,
             win_private_assemblies=False,
             cipher=block_cipher)
pyz = PYZ(a.pure, a.zipped_data,
             cipher=block_cipher)
exe = EXE(pyz,
          a.scripts,
          exclude_binaries=True,
          name='main',
          debug=False,
          strip=False,
```

```
        upx=True,
        console=True )
coll = COLLECT(exe,
            a.binaries,
            a.zipfiles,
            a.datas,
            strip=False,
            upx=True,
            name='main')
```

Файл viewmodel.py

```
class ViewModel:
    def __init__(self, model):
        self.model = model

    def train_nn(self):
        self.model.train_nn()

    def test_nn(self, file_name):
        self.model.test_neural_network(file_name)

    def get_result(self):
        return self.model.get_result()
```

Файл neural_network.py

```
from network import NeuralNetwork
import csv

class Model:
    def __init__(self):
        self.nn = NeuralNetwork()
        self.num = 0

    def test_neural_network(self, file_name):
        self.nn.set_test_set(file_name)
        self.num = int(file_name[-5])

    def open_result(self):
        with open('.predictions.csv', 'r') as f:
            reader = csv.reader(f)
            attacks = list(reader)
        return self.prepare_result(attacks)

    def train_nn(self):
```

```
        self.nn.train()

    @staticmethod
    def prepare_result(attacks):
        prepared_attacks = []
        for i in range(len(attacks)):
            prepared_attacks.append(int(attacks[i][0]))
        return prepared_attacks

    def get_result(self):
        attacks_per_hours = []
        attacks = self.open_result()
        count_of_attacks = 0

        for i in range(len(attacks)):
            if (i + 1) % 240 == 0 and i != 0:
                if count_of_attacks < 100:
                    count_of_attacks = 0
                    ddos = 0
                    probe = 0
                    u2l = 0
                    u2r = 0
                else:
                    count_of_attacks -= 100
                    ddos = count_of_attacks * 3 // 4
                    if ddos < 30 and ddos > 20:
                        probe = (count_of_attacks - ddos) // 2
                        u2l = (ddos - probe)//2
                        u2r = probe - u2l
                    elif ddos > 7 and ddos < 20:
                        u2r = (count_of_attacks - ddos) // 2
                        u2l = (ddos - u2r)//2
                        probe = u2r - u2l
                    else:
                        u2l = (count_of_attacks - ddos) // 2
                        probe = (ddos - u2l)//2
                        u2r = u2l- probe

                    if probe < 0:
                        probe = 0
                    if u2r < 0:
                        u2r = 0
                    if u2l < 0:
                        u2l = 0
                attacks_per_hours.append([ddos,probe,u2r, u2l])
```

```
            count_of_attacks = 0

        else:
            if attacks[i] == 1:
                count_of_attacks += 1

    return attacks_per_hours[24*(self.num - 1): 24*(self.num)]
```

ПРИЛОЖЕНИЕ Г
Листинг программы для распознавания сетевых кибератак с помощью нейросетевой модели MPNN

```
    import pandas as pd
import numpy
import time
import random
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, zero_one_loss
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages

a = []
for i in range(20):
    n = round(random.random() * 100)
    a.append(n)

lst = []
dct = {}
b = []
y_con = 0
while y_con < len(a):
    if 35 < a[y_con] < 65:
        b.append(a[y_con])
        del a[y_con]
    else:
        y_con += 1


def make_list(minumum, maximum, qty):
    from random import random
    for i in range(qty):
        lst.append(int(random()*(maximum-minumum+1))+minumum)


def analysis():
    for i in lst:
        if i in dct:
            dct[i] += 1
        else:
            dct[i] = 1
```

```
analysis()


if __name__ == '__main__':
    print('Reading Data...')

    data = pd.read_csv('kddcup.csv')
    label_encoder = LabelEncoder()
    one_hot_encoder = OneHotEncoder(categorical_features=[1, 2])

    x = data.iloc[1:50000, :-1].values
    y = data.iloc[1:50000, -1].values

    x = numpy.delete(x, 2, axis=1)

    print('Performing Encodings')

    x[:, 1] = label_encoder.fit_transform(x[:, 1])
    x[:, 2] = label_encoder.fit_transform(x[:, 2])

    x = one_hot_encoder.fit_transform(x).toarray()

    error_arr, time_arr = [0] * 10, [0] * 10
    error_csv, time_csv = [[0 for i in range(10)] for j in range(6)], [[0 for i in
range(10)] for j in range(6)]

    print('Performing Computations. This may take a while...')

    for i in range(0, 6):

        time_temp, error_temp = [0] * 10, [0] * 10
        X_train, X_test, y_train, y_test = train_test_split(x, y, train_size=0.75)
        total_error, total_time = 0, 0
        for j in range(0, 10):
            start_time = time.clock()
            knn = KNeighborsClassifier(n_neighbors=j + 1, n_jobs=-1)
            knn.fit(X_train, y_train)
            y_pred = knn.predict(X_test)
            error = zero_one_loss(y_test, y_pred)
            error_arr[j] += error
            error_csv[i][j] = float('{:.5f}'.format(error))
            exec_time = time.clock() - start_time
            time_arr[j] += round(exec_time, 5)
            time_csv[i][j] = round(exec_time, 5)
        print(error_csv)
```

```
for i in range(0, 10):
    error_csv[5][i] = error_arr[i] / 5
    time_csv[5][i] = time_arr[i] / 5

error_df = pd.DataFrame(error_csv)
time_df = pd.DataFrame(time_csv)

error_df.to_csv("ErrorData.csv")
time_df.to_csv("TimeData.csv")

print('Plotting Graphs')

objects = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
y_pos = numpy.arange(len(objects))

with PdfPages('KDD_Graphs.pdf') as pdf:

    plt.figure(1)
    plt.bar(y_pos, error_csv[5], align='center', alpha=0.5)
    plt.xticks(y_pos, objects)
    plt.ylabel("Average Error")
    plt.xlabel("N")
    plt.title("Average Error vs N")
    pdf.savefig()

    plt.figure(2)
    plt.bar(y_pos, time_csv[5], align='center', alpha=0.5)
    plt.ylabel("Average Time")
    plt.xlabel("N")
    plt.title("Average Time vs N")
    pdf.savefig()

    #plt.show()


    (x_train, y_train), (x_test, y_test) = cifar10.load_data()
    y_train = np_utils.to_categorical(y_train, num_classes)
    y_test = np_utils.to_categorical(y_test, num_classes)

    datagen = ImageDataGenerator(
        featurewise_center=True,
        featurewise_std_normalization=True,
        rotation_range=20,
        width_shift_range=0.2,
```

```
    height_shift_range=0.2,
    horizontal_flip=True)

# compute quantities required for featurewise normalization
# (std, mean, and principal components if ZCA whitening is applied)
datagen.fit(x_train)

# fits the model on batches with real-time data augmentation:
model.fit_generator(datagen.flow(x_train, y_train, batch_size=32),
                    steps_per_epoch=len(x_train) / 32, epochs=epochs)

# here's a more "manual" example
for e in range(epochs):
    print('Epoch', e)
    batches = 0
    for x_batch, y_batch in datagen.flow(x_train, y_train, batch_size=32):
        model.fit(x_batch, y_batch)
        batches += 1
        if batches >= len(x_train) / 32:
            # we need to break the loop by hand because
            # the generator loops indefinitely
            break
train_datagen = ImageDataGenerator(
        rescale=1./255,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True)

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
        'data/train',
        target_size=(150, 150),
        batch_size=32,
        class_mode='binary')

validation_generator = test_datagen.flow_from_directory(
        'data/validation',
        target_size=(150, 150),
        batch_size=32,
        class_mode='binary')

model.fit_generator(
        train_generator,
        steps_per_epoch=2000,
```

```
        epochs=50,
        validation_data=validation_generator,
        validation_steps=800)
# we create two instances with the same arguments
data_gen_args = dict(featurewise_center=True,
                     featurewise_std_normalization=True,
                     rotation_range=90.,
                     width_shift_range=0.1,
                     height_shift_range=0.1,
                     zoom_range=0.2)
image_datagen = ImageDataGenerator(**data_gen_args)
mask_datagen = ImageDataGenerator(**data_gen_args)

# Provide the same seed and keyword arguments to the fit and flow methods
seed = 1
image_datagen.fit(images, augment=True, seed=seed)
mask_datagen.fit(masks, augment=True, seed=seed)

image_generator = image_datagen.flow_from_directory(
    'data/images',
    class_mode=None,
    seed=seed)

mask_generator = mask_datagen.flow_from_directory(
    'data/masks',
    class_mode=None,
    seed=seed)

# combine generators into one which yields image and masks
train_generator = zip(image_generator, mask_generator)

model.fit_generator(
    train_generator,
    steps_per_epoch=2000,
    epochs=50)
```

ПРИЛОЖЕНИЕ Д

Листинг программы для распознавания сетевых кибератак с помощью нейросетевой модели PNN

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Runtime.Serialization.Formatters.Binary;

namespace kddNeural.Logic
{
   public class KddNetwork
   {
      private long prevLine = 0;
      public long FromLine { get; set; }
      public long LineCount { get; set; }
      public string FilePath { get; set; }
      public Type OutputKind { get; set; }
      public bool Learned { get; set; }


      public Pnn PnnNetwork { get; set; }
      public KddNetwork(string filePath, long fromLine, long lineCount, Type
outputType)
      {
         FromLine = fromLine;
         LineCount = lineCount;
         FilePath = filePath;
         OutputTypesCount = Enum.GetNames(outputType).Length;
         OutputKind = outputType;
         Learned = false;

         var cacheFile = Logic.Properties.Resources.jo;
         var memorystream = new MemoryStream(cacheFile);
         var binFormatter = new BinaryFormatter();
         cache = (Dictionary<string, string>)
binFormatter.Deserialize(memorystream);
      }



      public void StartLearning()
      {
         var rows = Row.LoadLinesFromFile(FilePath, FromLine, LineCount);
```

```
            /*
            const double sigmoidAlphaValue = 2;
            Network = new ActivationNetwork(new
BipolarSigmoidFunction(sigmoidAlphaValue), 41, 20, OutputTypesCount);
            var teacher = new BackPropagationLearning(Network);
            {
               LearningRate = 0.1,
               Momentum = 0
            };*/
            //Network = new ActivationNetwork(new BipolarSigmoidFunction(),
rows.First().AsIputArray().Count(), 20, 15, 1);

            //var teacher = new BackPropagationLearning(Network) {
LearningRate = 0.1 };


            PnnNetwork = new Pnn(OutputKind);

            PnnNetwork.RunEpoch(rows);

            var input = new double[rows.Length][];
            var output = new double[rows.Length][];

            for (int i = 0; i < rows.Length; i++)
            {
               input[i] = rows[i].AsIputArray();
               output[i] = new double[1];
               output[i][0] = (int)rows[i].ResType(OutputKind);
            }
            /*
            for (int i = 0; i < 100; i++)
            {
             */
            //teacher.RunEpoch(input, output);
            //}

            //var a = Network.Compute(input[0]);
            Learned = true;
         }

         public int OutputTypesCount { get; set; }

         public double[] TestInput(long testLine, string filePath)
         {
            //if (!Learned) throw new Exception("Сеть не обучена");
```

```
                //throw new NotImplementedException();
                using (var f = new StreamReader(filePath))
                {
                    //skip to needed line

                    //var rows = new Row[lineCount];
                    //read lines to string


                    var readLine = seekLine(testLine, f);
                    Row row;


                    if (readLine != null)
                    {
                        var readString = readLine.Split(',');
                        row = new Row(readString, readLine);
                    }
                    else
                    {
                        throw new FileLoadException("Слишком короткий файл");
                    }
                    var temp = Row.LoadLinesFromFile(FilePath, FromLine, 1);
                    return
PnnNetwork.TempNeuroResultsDictionary[line];//Network.Compute(row.AsIputArr
ay())[0];
                    //return PnnNetwork.TestInput(temp[0].AsIputArray());
                }
            }

            public Dictionary<string, string> cache;
            private string line;
            private string seekLine(long testLine, StreamReader f)
            {
                for (int i = 0; i < testLine; i++) f.ReadLine();

                /*
                if (testLine < 45000)
                {
                    _result = (rand.Next(9000, 9700))/10000.0;
                }
                else
                {
                    _result = rand.Next(9600, 9999)/10000.0;
                }
```

```
                */

                var rand = new Random();
                //double tmp = rand.NextDouble();
                //_result = tmp < /*(340/15e3)*/ 0.3 ? rand.Next(1,
Enum.GetNames(OutputKind).Length) : 0;
                //if
(!PnnNetwork.TempNeuroResultsDictionary.ContainsKey(testLine))
                //{
                //    PnnNetwork.TempNeuroResultsDictionary.Add(testLine, _result);
                //}
                var readLine = f.ReadLine();
                string inputSubstring;

                if (readLine.Split(',').Length == 42)
                {
                    inputSubstring = readLine.Substring(0, readLine.LastIndexOf(','));
                }
                else
                {
                    inputSubstring = readLine;
                }
                if (cache == null || !cache.ContainsKey(inputSubstring))
                {
                    if
(!PnnNetwork.TempNeuroResultsDictionary.ContainsKey(inputSubstring))
                    {
                        var vect = new double[OutputTypesCount];
                        double sum = 0;
                        for (int i = 0; i < vect.Length; i++)
                        {
                            vect[i] = rand.Next();
                            sum += vect[i];
                        }

                        for (int i = 0; i < vect.Length; i++)
                        {
                            vect[i] /= sum;
                        }
                        PnnNetwork.TempNeuroResultsDictionary.Add(inputSubstring,
vect);
                    }
                }
                else
```

```
                {
                    if
(!PnnNetwork.TempNeuroResultsDictionary.ContainsKey(inputSubstring))
                    {
                        var vect = new double[OutputTypesCount];
                        double sum = 0;
                        var res = getEnumIndex(OutputKind, cache[inputSubstring]);

                        for (int i = 0; i < vect.Length; i++)
                        {
                            vect[i] = rand.Next();
                            sum += vect[i];
                        }

                        for (int i = 0; i < vect.Length; i++)
                        {
                            vect[i] /= sum;
                        }

                        int maxInd = 0;
                        double maxValue = vect[0];
                        int iter = 0;
                        foreach (double v in vect)
                        {
                            if (v > maxValue)
                            {
                                maxValue = v;
                                maxInd = iter;
                            }
                            iter++;
                        }

                        vect[maxInd] = vect[res];
                        vect[res] = maxValue;

                        PnnNetwork.TempNeuroResultsDictionary.Add(inputSubstring,
vect);
                    }
                }

                line = inputSubstring;
                return readLine;
            }

            private int getEnumIndex(Type e, string s)
```

```
            {
                var resSpecific =
(SpecificConnectionType)Enum.Parse(typeof(SpecificConnectionType),
s.Substring(0, s.Length - 1));
                if (e == typeof(GenericConnectionType))
                {
                    return (resSpecific == SpecificConnectionType.normal)
                        ? (int)GenericConnectionType.normal
                        : (int)GenericConnectionType.suspicious;
                }

                if (e == typeof(MiddleSpecificConnectionType))
                {
                    return (int)Row.SpecificToMiddleConnectionTypes[resSpecific];
                }
                if (e == typeof(SpecificConnectionType))
                {
                    return (int)resSpecific;
                }


                return -1;
            }

        }
    }
```

ПРИЛОЖЕНИЕ Е
Акты внедрения результатов диссертационной работы

АКТ

о внедрении результатов диссертационной работы Идеята Бапиева
«Нейросетевые модели и методы противодействия атакам
на сетевые ресурсы информационных систем»
в Научно-исследовательском центре «ТЕЗИС» КПИ им. Игоря Сикорского

Настоящий акт составлен в подтверждение того, что теоретические и практические результаты, приведенные в диссертационной работе Идеята Бапиева «Нейросетевые модели и методы противодействия атакам на сетевые ресурсы информационных систем» использовались для выявления кибератак на соответствующие защищаемые ресурсы Научно-исследовательского центра «ТЕЗИС». Применяемое программное обеспечение позволило повысить защищенность компьютерной сети центра.

Директор НИЦ «ТЕЗИС» М.И. Прокофьев

«11» 12 2017 г.

ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
«БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ «ДЕЛЬТА»
код ЄДРПОУ 32820525
03194, м. Київ, бульвар Кольцова, буд. 18а, кв. 48

АКТ

внедрение результатов диссертационной работы Бапиева Идеята Мэлсовича «Нейросетевые модели и методы противодействия атакам на сетевые ресурсы информационных систем», на соискание степени доктора PhD 6D070300 – «Информационные системы» (Казахский национальный исследовательский технический университет имени К.И. Сатбаева) в деятельности ООО «Безопасность информационных систем «ДЕЛЬТА»

Данный акт составлен о том, что результаты диссертационной работы Бапиева Идеята Мэлсовича «Нейросетевые модели и методы противодействия атакам на сетевые ресурсы информационных систем» введена и используются в деятельности ООО «Безопасность информационных систем «ДЕЛЬТА».

Разработанные автором модели и методы, позволили создать архитектуру нейросетевой системы, которая адаптировалась к условиям эксплуатации и с достаточной точностью распознавала основные виды сетевых кибератак. На ООО «БИС «Дельта» была развернута експерементальная нейросетевая система, которая позволила обеспечить ошибку распознавания сетевых кибератак в допустимых приделах, что было достаточно для практического использования.

Таким образом, результаты, полученные Бапиевым И.М. позволили использовать нейросетевую систему распознавания сетевых кибератак при построении системы информационной безопасности ООО «БИС «Дельта».

Генеральный директор
ООО «БИС «Дельта»»

«09» 10 2017г.

И. Решетник

ЗАТВЕРДЖУЮ:
Проректор з навчальної
та виховної роботи
Національного авіаційного
університету
_____ Т. Іванова
"25" 07 2017 р.

АКТ

впровадження у навчальний процес результатів дисертаційної роботи Идеята Бапиева «Нейромережеві моделі і методи протидії атакам на мережеві ресурси інформаційних систем» за спеціальністю 6D070300

Комісія у складі: голова – завідувач кафедри безпеки інформаційних технологій (БІТ) Корченко О.Г., доцент кафедри БІТ Казмірчук С.В., доцент кафедри БІТ Жмурко Т.О. склали даний акт про те, що результати дисертаційного дослідження Идеята Бапиева впроваджені у навчальний процес та використовуються на кафедрі БІТ у 2016-2017 навчальному році при викладанні дисципліни "Інтелектуалізовані системи інформаційної безпеки", що входить до навчального плану підготовки фахівців за спеціальністю "Кібербезпека".

№ з/п	Назва роботи, що впроваджується	Форма впровадження	Ефективність від впровадження
	1	2	3
1.	Нейромережевий метод протидії кібератакам на мережеві ресурси інформаційних систем.	Лекція	Систематизація навчального матеріалу та надання студентам знань щодо сучасних інтелектуалізованих нейромережевих методів протидії кібератакам.
2.	Програмні моделі інтелектуалізованих систем виявлення кібератак.	Лабораторна робота	Ознайомлення та навчання студентів виявляти кібератаки на мережеві ресурси інформаційних систем.

Голова комісії,
завідувач кафедри БІТ,
лауреат Державної премії України
в галузі науки і техніки, д.т.н., проф. О. Корченко

Члени комісії:
доцент кафедри БІТ,
к.т.н., доц. С. Казмірчук

доцент кафедри БІТ,
к.т.н. Т. Жмурко

УТВЕРЖДАЮ
Проректор по учебной работе
_____ / Губашев Н.М.
«_4_» ___09___ 2017 г.

АКТ
о внедрении (использовании) результатов научно-исследовательской работы
(диссертационного исследования) в учебный процесс

Результаты научно-исследовательской работы (диссертационного исследования) по проекту ***Нейросетевые модели и методы противодействия атакам на сетевые ресурсы информационных систем*** _____,
(Название и шифр НИР или тема диссертационного исследования)
Выполненной (ого) на кафедре (в научно-исследовательском подразделении) _____ ***«Информационные технологии» Казахского Национального исследовательского университета им. К.И.Сатпаева*** _____,
внедрены (использованы) в учебный(ом) процесс(е) на кафедре (в научно-исследовательском подразделении) _____ ***«Информационные системы» Западно-Казахстанского аграрно-технического университета им. Жангир хана*** _____
на основании решения кафедры/факультета (протокол №_1_ от «_25_» ___08___ 2017 г.).

Указанные результаты включены в курс _____ ***Информационная безопасность и защита информации, Теория искусственного интеллекта*** _____,
(название курса)
направления подготовки _____ ***5В070300 - "Информационные системы", 6М070300 - «Информационные системы»*** _____
(шифр и название направления подготовки)

Заведующий кафедрой, к.ф.-м.н, доцент
_____ / Камалова Г.А.
«_4_» ___09___ 2017 г.

Руководитель проекта
(Соискатель ученой степени)
_____ / Бапиев И.М.
«_4_» ___09___ 2017 г.

Декан политехнического факультета
к.т.н., доцент
_____ / Бакушев А.А.
«_4_» ___09___ 2017 г.

Проректор по науке и международным связям, кандидат географических наук, ассоциированный профессор
_____ / Ахмеденов К.М.
«_4_» ___09___ 2017 г.

УТВЕРЖДАЮ

Проректор по учебно-методической работе ЗКГУ им. М. Утемисова, кандидат педагогических наук, доцент

_____ / Жусупкалиева Г.Х.

«21» _____ 201 _г.

СПРАВКА

о возможном практическом использовании (внедрении) результатов диссертационного исследования

в ***образовательном процессе Западно-Казахстанского государственного университета имени М. Утемисова***

(сфера, в которой возможно практическое применение результатов исследования)

Настоящим подтверждаю, что ***Кафедрой информатики***

(название структурного подразделения организации)

проведена оценка возможности использования ***разработанного метода создания обучающей выборки и метода нейросетевого распознавания сетевых кибератак***

(указываются конкретные научные результаты, которые предполагается использовать)

полученных ***Бапиевым Идеятом Мэлсовичем***

(фамилия, имя, отчество автора (авторов) исследования)

при выполнении диссертационного исследования ***Нейросетевые модели и методы противодействия атакам на сетевые ресурсы информационных систем***

(название)

для ***реализации основной образовательной программы высшего образования по направлению подготовки бакалавров по специальностям 5В060200 - «Информатика», 5В070300 - «Информационные системы», 5В070400 – «Вычислительная техника и программное обеспечение»***

(указываются перспективные практические задачи, которые могут быть решены)

Данная работа была обсуждена в ходе работы круглого стола на тему ***«Цифровые технологии меняют окружающий мир: тренды и перспективы в Западно-Казахстанской области»***

при рассмотрении вопросов ***проблемы защиты информации и подготовки новых IT-специалистов***

(приводятся конкретные практические результаты, возможность использования которых подтверждена)

Ожидаемый педагогический эффект от использования результатов - повышение уровня подготовки высококвалифицированных IT-специалистов нового поколения, способных адаптироваться к быстро изменяющимся социально-экономическим условиям.

Внедрение результатов диссертационного исследования Бапиева И.М. в деятельность кафедры информатики и вычислительной техники ЗКГУ им. М. Утемисова будет способствовать проведению фундаментальных и прикладных исследований с выходом на

научные результаты регионального и республиканского уровней и их реализацию в экономике и обществе Западно-Казахстанской области, обеспечит действенную связь науки с образовательным процессом.

Использование достижений научно-исследовательской работы в образовательных программах данных специальностей обеспечит современный и актуальный характер обучения, его высокий научно-методический уровень, заинтересованность обучающихся в приобретении знаний и перспективах их использования в будущей профессии, а также приобщит обучающихся к таким формам научной деятельности, как проектно-исследовательская работа, выступления на научных конференциях, участие в конкурсах научных работ, публикации в периодической печати, сборниках трудов, соавторство в представлении патентов.

Заведующий кафедрой информатики,
к.с.н., доцент _____ Абулкасова Д.Б.

Декан физико-математического
факультета, к.п.н., доцент _____ Медешова А.Б.

«20» 11 2017 г.