МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»
Кафедра систем информационной безопасности

«ДОПУЩЕН К ЗАЩИТЕ»
Зав. кафедрой к.п.н., доцент Р. Ш. Бердибаев
_____ «_____» _____2019 г.
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Применение нейросетевых технологий при обнаружении вторжений
Специальность: 5В100200 – «Системы информационной безопасности»
Выполнил: Бекмухамбетов Алмаз Ахсенович Группа: СИБ-15-2
Научный руководитель: Аскарова Нурсанат Темирбековна

Консультант:
по экономической части:
к. э. н., профессор Арнабаева Ж. Т.
(ученая степень, звание, Ф.И.О)
(подпись) «27» мая 2019 г.

по безопасности жизнедеятельности:
д. т. н. ст. преподаватель Бердибаев Ш. Ш
(ученая степень, звание, Ф.И.О)
(подпись) «27» мая 2019 г.

по применению вычислительной техники:
ст. преподаватель Аскарова Н. Т.
(ученая степень, звание, Ф.И.О)
(подпись) «11» июня 2019 г.

Нормоконтролер:
ст. преподаватель Аскарова Н. Т.
(ученая степень, звание, Ф.И.О)
(подпись) «12» июня 2019 г.

Рецензент:
к. т. н. Тасмуханов Сырым Бимуратович
(ученая степень, звание, Ф.И.О)
(подпись) «14» июня 2019 г.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ»

Институт систем управления и информационных технологий

Кафедра систем информационной безопасности

Специальность 5В100200 - Системы информационной безопасности

ЗАДАНИЕ
на выполнение дипломной работы

Студенту Бекмухамбетову Алмазу Ахсеновичу

Тема работы: Применение нейросетевых технологий при обнаружении вторжений .

Утверждена приказом по университету №_124_ от «_26_» _10_ 2019 г.

Срок сдачи законченного проекта «_10_» _июня_ 2019 г.

Исходные данные к работе (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): проект подразумевает разработку искусственной нейросети в среде Matlab, позволяющую определять сетевые атаки, их тип.

Перечень вопросов, подлежащих разработке в дипломном проекте, или краткое содержание дипломного проекта: дипломный проект включает в себя 4 главы, разделенных на подглавы, каждая из которых освещает определенную тематику, используемую при разработке нейросетевого инструмента защиты.

В первой главе дипломного проекта сделать теоретический обзор по нейросетевым технологиям.

Во второй главе дипломного проекта рассмотреть сетевые атаки, их виды, а также разработать ИНС в среде Matlab, произвести обучение нейросети на обнаружение сетевых атак (DoS).

В третей главе работы провести технико-экономическое обоснование, показывающее актуальность разработки нейросетевого инструмента защиты

В четвертой главе дипломной работы определить необходимые условия для комфортной разработки программного продукта.

Конструкции по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант	Сроки	Подпись
Экономика	Арибаева Ж. Г.	04.03. – 27.05.19	
БЖД	Бейбасова Ш. Ч	702 - 8.05.19	
Основная часть	Макарова А. П	20.12.18 - 01.04.19	

График
подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
1. Подбор теоретического материала о нейросетях	10.01.19 - 20.01.19	
2. Выбор способа обучения нейросети	20.01.19 - 30.01.19	
3. Выбор инструментария для построения нейросетей	30.01.19 - 15.02.19	
4. Изучение основ Matlab	15.02.19 - 28.02.19	
5. Изучение нейросетей в Matlab	1.03.19 - 15.03.19	
6. Изучение типов сетевых атак	15.03.19 - 30.03.19	
7. Обучение нейросети сетевыми атаками в Matlab	30.03.19 - 15.04.19	
8. Тестирование нейросети	15.04.19 - 30.04.19	
9 Расчёт экономической части	01.05.19 - 15.05.19	
10. Расчет части по безопасности жизнедеятельности	15.05.19 - 30.05.19	

Дата выдачи задания «9» 04 2019 г.

Заведующий кафедрой _____ (_____)
(Подпись) (Ф.И.О)

Научный руководитель Проекта _____ (Аскерова Л. Ю.)
(Подпись) (Ф.И.О)

Задание принял к исполнению студент _____ (Бекмухамбетов Я.А.)
(Подпись) (Ф.И.О)

АННОТАЦИЯ

В дипломном проекте исследованы искусственные нейросети, их классификация, строение. Описаны способы обучения нейросетей. Также были рассмотрены сетевые атаки, их виды, база данных атак, на основе которой обучалась нейросеть. В практической части описываются DoS атаки, среда Matlab.

В среде Matlab R2019a, с помощью пакета Neural Network Toolbox, была развернута нейросеть, представлено ее обучение DoS атакам и получены результаты. Был проведен сравнительный анализ и выбрана нейросеть с оптимальным количеством нейронов. Также было проведено тестирование нейросети на обнаружение атак.

АҢДАТПА

Дипломдық жобада жасанды нейрожелілер, олардың жіктелуі, құрылысы зерттелген. Нейрожелілерді оқыту тәсілдері сипатталған. Сондай-ақ, желілік шабуылдар, олардың түрлері, нейрожеліге негізделген шабуылдардың деректер базасы қарастырылды. Практикалық бөлімде DoS шабуылдар, Matlab ортасы сипатталады.

Matlab R2019a ортасында, Neural Network Toolbox арқылы, нейрожелі құрылып, оны DoS шабуылдарға үйрету ұсынылды және нәтижелер алынды. Салыстырмалы талдау жүргізілді және нейрондардың оптималды санымен нейрожелі таңдалды. Сондай-ақ, шабуылдарды анықтау үшін нейрожеліні тестілеу жүргізілді.

ANNOTATION

The thesis project investigated artificial neural networks, their classification, structure. Methods of neural networks training are described. Network attacks, their types, the database of attacks which was used as the basis of the training of the neural network were also considered. The practical part describes DoS attacks, Matlab environment.

The neural network was deployed in the Matlab R2019a environment using the Neural Network Toolbox package, its training to DoS attacks was presented and the results were obtained. A comparative analysis was carried out and a neural network with an optimal number of neurons was selected. The neural network was also tested to detect attacks.

Содержание

Введение

В настоящее время все большую актуальность обретает направление по разработке новых способов и методов защиты информации. Одновременно с развитием технологии защиты, также получают аналогичное развитие технологии взлома и получение доступа к защищенным данным. Особую ценность представляют методы интеллектуальной защиты. К их числу можно отнести искусственные нейронные сети (ИНС).

В системах обеспечения информационной безопасности ИНС весьма эффективны при решении задач анализа трафика, аудита баз данных, эвристического детектирования вредоносных атак и новых типов вирусов, анализа поведения злоумышленника и выявления аномалий в действиях пользователей.

Для обнаружения сетевых атак наряду с общепризнанными сигнатурными подходами к определению сетевых атак, широко применяются методы с использованием искусственного интеллекта, генетических алгоритмов, нейронных сетей, механизмов нечеткого вывода и других методов. Из-за неограниченного разнообразия атак специальные базы данных с правилами или сигнатурами для обнаружения атак нуждаются в непрерывном администрировании, необходимо добавлять все новые правила. Одним из путей устранения данной проблемы является использовать нейронные сети в качестве механизма для обнаружения сетевых атак. В отличии от сигнатурного подхода, нейронная сеть проводит анализ информации и предоставляет информацию об атаках, которые она научена распознавать. Кроме этого, нейронные сети обладают неоспоримым преимуществом – они способны адаптироваться к ранее неизвестным атакам и обнаруживать их. Обнаружение сетевых атак связано с выделением определенных параметров, которые необходимы для выявления той или иной атаки. Из каждой атаки можно выделить большое количество различных параметров, но не все они требуются для выявления этой атаки. Поэтому стоит задача классификации параметров обнаружения сетевых атак, а также уменьшения их числа для более быстрой работы системы обнаружения

1 Нейронные сети

Искусственные нейронные сети за свою более чем полувековую историю существования оказали существенное влияние на ряд наук, положив начало развитию новых, а также созданию подразделов в уже существующих науках. Это открывает новые перспективы их использования, способствует решению трудноформализуемых задач.

Искусственная нейронная сеть – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма [7].

Первая математическая модель нейрона (базового элемента мозга) была создана в 1943 году, американским ученым Уорреном Маккаллоком (McCulloch W.) и его учеником У. Питтсом (Pitts W.). Они первые:

1) Разработали модель нейрона как простейшего процессорного элемента, выполнявшего вычисление переходной функции от скалярного произведения вектора входных сигналов и вектора весовых коэффициентов;

2) Предложили конструкцию сети таких элементов для выполнения логических и арифметических операций;

3) Сделали основополагающее предположение о том, что такая сеть способна обучаться, распознавать образы, обобщать полученную информацию [10].

В 1958 г. Ф. Розенблатт предложил свою модель нейронной сети (НС). Он ввел в модель У. Маккаллока и У. Питтса способность связей к модификации, что сделало ее лучше обучаемой. Эта модель была названа персептроном. Также Ф. Розенблатт реализовал эти принципы в электронном устройстве «Марк-1», способном распознавать печатные буквы и обучаться на примерах. Этой работой заинтересовались военные, что дало мощный импульс развитию нейронных сетей. В 1969 г. было доказано, что такие ограничения возможностей НС непреодолимы. Это привело к потере интереса к исследованию в области НС, но благодаря работам Хопфилда и Хехт-Нильсена, интерес к нейросетям возобновился в начале 80-х годов прошлого столетия, и наработки в этой области стали внедряться в практику. Примерно в это же время был найден универсальный алгоритм обучения нейросетей, который известен как метод обратного распространения (Back Propagation of Error или backprop). Американские математики Румельхарт, Хинтон и Вильямс первые сформулировали основные принципы этого алгоритма.

ИНС представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. В то же время соединённые в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи [9].

В качестве образов могут выступать различные по своей природе

объекты: символы текста, изображения, образцы звуков и т. д. При обучении сети предлагаются различные образцы образов с указанием того, к какому классу они относятся. Образец, как правило, представляется как вектор значений признаков. При этом совокупность всех признаков должна однозначно определять класс, к которому относится образец. В случае, если признаков недостаточно, сеть может соотнести один и тот же образец с несколькими классами, что неверно. По окончании обучения сети ей можно предъявлять неизвестные ранее образы и получать ответ о принадлежности к определённому классу.

Персептрон – это простейшая нейронная сеть, построенная на основе модели МакКаллока-Питса [14], веса и смещения которого могут быть настроены таким образом, чтобы решить задачу классификации входных векторов (Рисунок 1.1).

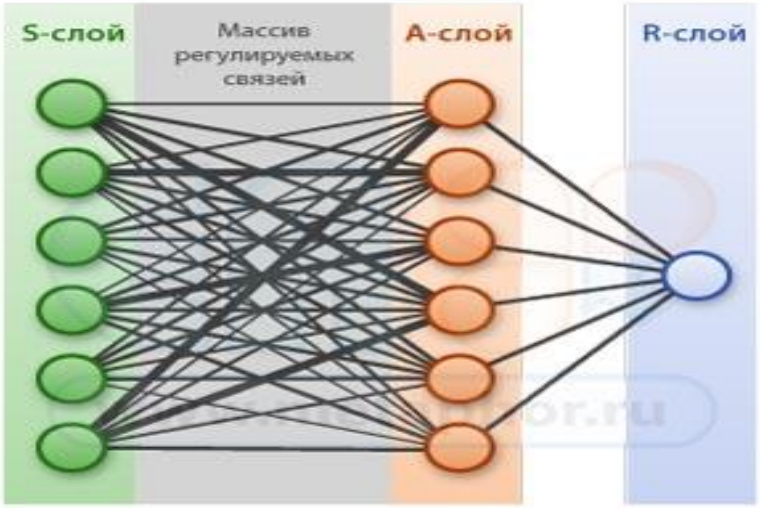

Рисунок 1.1 – Простой персептрон[14],

В персептроне элементы трех видов распределены по трем слоям, обычно обозначаемых буквами S, A и R. В S-слое (рецепторах) находятся элементы, через которые в персептрон поступает входная информация. Далее сигналы от каждого рецептора уходят в следующий A-слой. В A-слое находятся нейроны (сумматор с пороговым эффектом со множеством входов и одним выходом). Нейрон суммирует все сигналы, которые в него поступают через синапсы, и когда сумма достигает определенного порога (L), нейрон активизируется и передает сигнал на R-слой. В R-слое находятся классификатор, который тоже является нейроном и работает также: он суммирует входные сигналы от нейронов в A-слое, также если сумма достигает определенного порога, классификатор срабатывает

1.1 Строение нейронных сетей

Нейронная сеть состоит из нейронов (рисунок 1.2).

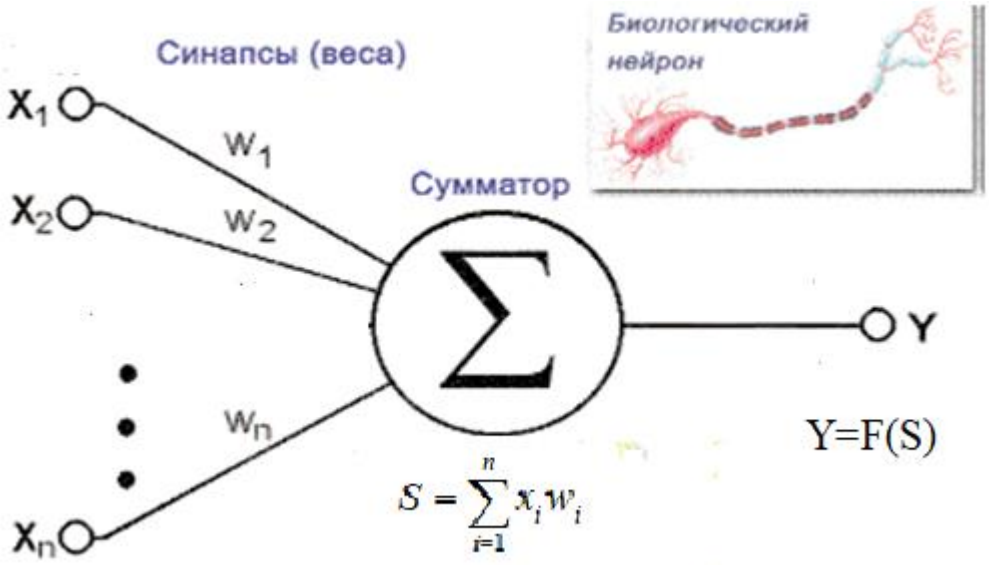

Рисунок 1.2 – Структурная схема нейрона

Структуру нейрона можно представить из следующих блоков:

1) входные сигналы x_n;
2) весовые коэффициенты w_n;
3) сумматор ∑ и его выход S;
4) функция активации нейрона F(x);
5) выходной сигнал Y.

Многослойная нейронная сеть включает в себя:

- выходной слой – слой, который содержит зачастую 1 нейрон и выдает результат расчетов всей нейронной сети. На основании этого сигнала строится дальнейшая логика управления советника;

Скрытые слои – слои обычных нейронов, которые передают сигналы от входа к выходу. Их входом служит выход предыдущего слоя, а выход - входом следующего слоя.

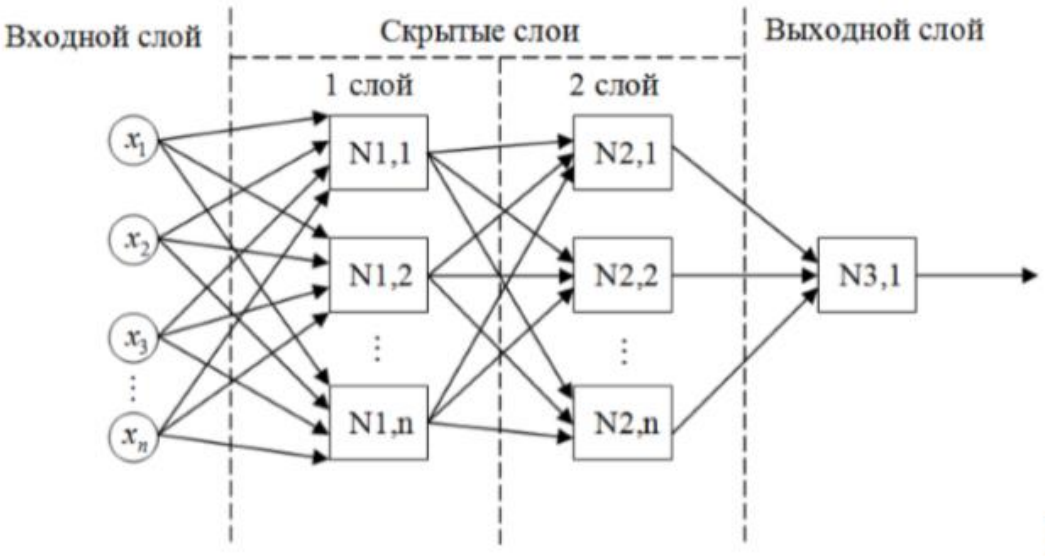

Рисунок 1.3 – Структурная схема многослойной нейронной сети

1.2 Классификация нейронных сетей

Существует несколько способов классификации нейронных сетей в зависимости от критерия: по характеру обучения, по настройке весов, по типу входной информации, по модели нейронной сети (Рисунок 1.4)

Рисунок 1.4 - Классификация нейронных сетей [2}

1) По характеру обучения НС делятся на следующие типы:
   - нейронные сети, проходящие обучение с учителем;
   - нейронные сети, проходящие обучение без учителя.

Обучение с учителем означает, что для каждого входного вектора существует целевой вектор, представляющий собой требуемый выход. Вместе они называются обучающей парой. Обычно сеть обучается на некотором числе таких обучающих пар. Предъявляется выходной вектор, вычисляется выход сети и сравнивается с соответствующим целевым вектором. Далее веса изменяются в соответствии с алгоритмом, стремящимся минимизировать ошибку. Векторы обучающего множества предъявляются последовательно, вычисляются ошибки и веса подстраиваются для каждого вектора до тех пор, пока ошибка по всему обучающему массиву не достигнет приемлемого уровня. Нейронные сети, использующие обучение без учителя. Обучение без учителя является намного более правдоподобной моделью обучения с точки зрения биологических корней искусственных нейронных сетей. Развитая Кохоненом и многими другими, она не нуждается в целевом векторе для выходов и, следовательно, не требует сравнения с предопределенными идеальными ответами. Обучающее множество состоит лишь из входных векторов. Обучающий алгоритм подстраивает веса сети так, чтобы получались согласованные выходные векторы, т. е. чтобы предъявление достаточно близких входных векторов давало одинаковые выходы. Процесс обучения, следовательно, выделяет статистические свойства обучающего множества и группирует сходные векторы в классы.

По настройке весов выделяют следующие НС:

– сети с фиксированными связями – весовые коэффициенты нейронной сети выбираются сразу, исходя из условий задачи;

– сети с динамическими связями – для них в процессе обучения происходит настройка синоптических весов.

По типу входной информации НС делятся на:

– аналоговую (входная информация представлена в форме действительных чисел);

– двоичную (входная информация представляется в виде нулей и единиц).

По модели НС:

– НС прямого распространения;

– рекуррентные НС;

– радиально-базисные функции;

– самоорганизующиеся карты или Сети Кохонена.

В сетях прямого распространения все связи направлены строго от входных нейронов к выходным (простейший персептрон, разработанный Розенблаттом, и многослойный персептрон). В рекуррентных НС сигнал с выходных нейронов или нейронов скрытого слоя частично передается обратно на входы нейронов входного слоя. Радиально базисная сеть обладает следующими особенностями: один скрытый слой, только нейроны скрытого слоя имеют нелинейную активационную функцию и синоптические

веса входного и скрытого слоев равны единице. Самоорганизующиеся карты или Сети Кохонена обучаются без учителя (применяется в задачах распознавания, кластеризации и визуализации многомерных данных). Сети этого класса способны выявлять новизну во входных данных: если после обучения сеть встретится с набором данных, непохожим ни на один из известных образцов, то она не сможет классифицировать такой набор и тем самым выявит его новизну. Сеть Кохонена имеет всего два слоя нейронов: входной и выходной, составленный из радиальных элементов.

1.3 Функции активации

Функция активации Y = F(S) определяет выходной сигнал нейрона как функцию его состояния S. Наиболее распространенными функциями активации являются ступенчатая пороговая, линейная пороговая, сигмоидная, арктангенс, а также линейная и гауссиана, приведенные в таблице 1.1.

Функция активации в общем виде будет выглядеть следующим образом:

$$\boldsymbol{Y = F(S - \theta)}$$

где

– *F(x)* – сама функции активации;

– S – средневзвешенная сумма, полученная на первом этапе вычисления выходного значения нейрона;

– $\boldsymbol{\theta}$ - пороговое значение срабатывания функции активации

Результаты вычислений предаются на выход не напрямую, а через функцию активации.

Процесс обучения сети сводится к изменению весовых коэффициентов.

$$\mathbf{S} = \sum_{\boldsymbol{n}} \boldsymbol{x_n w_n}$$

S в данном случае и есть результат вычислений нейрона.

Таким образом, функция активации нейрона в нейронных сетях – это функция, которая вычисляет выходной сигнал нейрона. Основные виды функций активации отображены в таблице 1.1 и на рисункt 1.5.

Таблица 1.1 – Основные виды функций активации

Название	Определение
Ступенчатая пороговая	$Y = 0$ при $S < a$, $Y = 1$ при $S >= a$
Линейная пороговая	$Y = 0$ при $s < a_1$, $y = ks + b$ при $a_1 <= s < a_2$, $y = 1$ при $s >= a_2$, $a_2 = 1/k + a_1$
Линейная	$Y = KS + B$
Сигмоидная	$Y = (1 + e^{-K(S-A)})^{-1}$

Гиперболический тангенс	$Y = th\ (x) = (e^{x} – e^{-x})/(e^{x} + e^{-x})$
Арктангенс	$Y = 2\ arctg\ (x)/p$
Гауссиана	$Y = e^{-k(S-A)}$

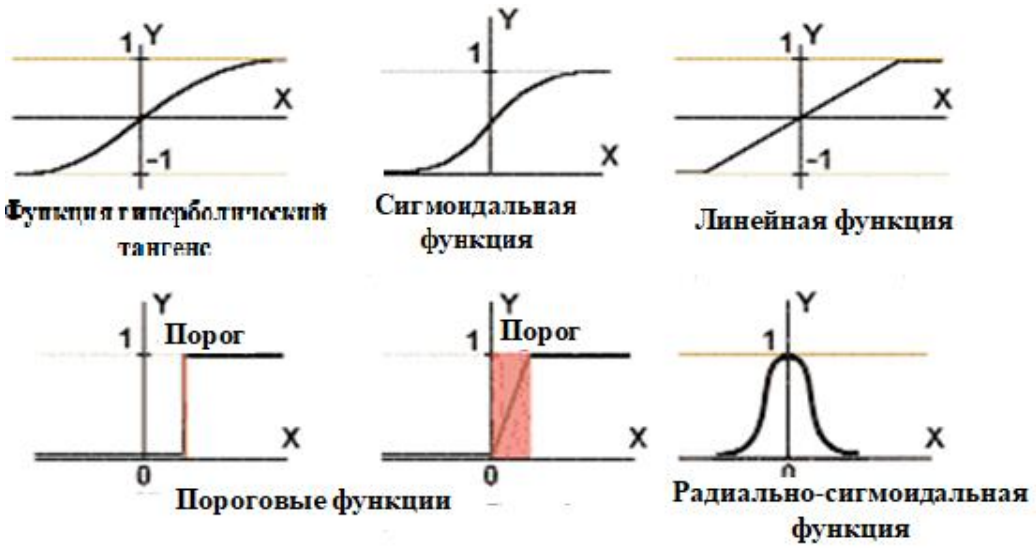

Рисунок 1.5 – Основные виды функций активации

1.4 Обучение нейросетей

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения – одно из главных преимуществ нейронных сетей перед традиционными алгоритмами.

Обучение – это процесс, в котором свободные параметры нейронной сети настраиваются посредством моделирования среды, в которую эта сеть встроена. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

Алгоритм обучения представляет собой следующую последовательность событий:

1) В нейронную сеть поступают стимулы из внешней среды;

2) в результате первого пункта изменяются свободные параметры нейронной сети;

3) после изменения внутренней структуры нейронная сеть отвечает на возбуждения уже иным образом.

Как было сказано выше, обучить нейронную сеть можно разными способами: с учителем, без учителя.

Обучение с учителем. В основном обучение с учителем применяется

для решения двух типов задач: классификации и регрессии.

В задачах классификации алгоритм предсказывает дискретные значения, соответствующие номерам классов, к которым принадлежат объекты. Например, при обучении нейросети распозновать фотографии животных, в обучающем датасете с фотографиями каждое изображение будет иметь соответствующую метку – «кошка», «коала» или «черепаха». Качество алгоритма оценивается тем, насколько точно он может правильно классифицировать новые фото с коалами и черепахами.

Обучение без учителя. В обучении без учителя у модели есть набор данных, и нет явных указаний, что с ним делать. Нейронная сеть пытается самостоятельно найти корелляции в данных, извлекая полезные признаки и анализируя их.

В зависимости от задачи модель систематизирует данные по следующим принципам:

1) Кластеризация – алгоритм подбирает похожие данные, находя общие признаки, и группируют их вместе.

2) Обнаружение аномалий – используется при возникновении аномалий или для нахождения выбросов в данных.

3) Ассоциации – рассматривая пару ключевых признаков объекта, модель может предсказать другие, с которыми существует связь.

4) Автоэнкодеры – принимают входные данные, кодируют их, а затем пытаются воссоздать начальные данные из полученного кода.

В обучении без учителя сложно вычислить точность алгоритма, так как в данных отсутствуют «правильные ответы» или метки. Но размеченные данные часто ненадежные или их слишком дорого получить. В таких случаях, предоставляя модели свободу действий для поиска зависимостей, можно получить хорошие результаты.

Таким образом, известны 4 основных типа правил обучения: коррекция по ошибке, машина Больцмана, правило Хебба и обучение методом соревнования. [4].

Правило коррекции по ошибке. Это обучение имеет место только в случае, когда персептрон ошибается. Известны различные модификации этого алгоритма обучения.

Обучение Больцмана. Целью этого способа обучения является такая настройка весовых коэффициентов, при которой состояния видимых нейронов удовлетворяют желаемому распределению вероятностей.

Правило Хебба. Важной особенностью этого правила является то, что изменение синаптического веса зависит только от активности нейронов, которые связаны данным синапсом.

Обучение методом соревнования. Этот вид обучения позволяет кластеризовать входные данные: подобные примеры группируются сетью в соответствии с корреляциями и представляются одним элементом. При обучении модифицируются только веса "победившего" нейрона. Эффект этого правила достигается за счет такого изменения сохраненного в сети

образца (вектора весов связей победившего нейрона), при котором он становится чуть ближе ко входному примеру.

Различные алгоритмы обучения и связанные с ними архитектуры сетей (список не является исчерпывающим) представлены в таблице 1.2. Каждый алгоритм обучения ориентирован на сеть определенной архитектуры и предназначен для ограниченного класса задач. Кроме рассмотренных, следует упомянуть некоторые другие алгоритмы: Adaline и Madaline, линейный дискриминантный анализ, проекции Саммона, анализ главных компонентов [10].

Таблица 1.2 – Известные алгоритмы обучения [10]

Парадигма	Обучающее правило	Архитектура	Алгоритм обучения	Задача
С учителем	Коррекция ошибки	Однослойный и многослоный персептрон	Алгоритмы обучения персептрона Обратное распространение Adaline, Madaline	Классификация образов Аппроксимация функций Предсказание, управление
	Больцман	Рекуррентная	Алгоритм обучения Больцмана	Классификация образов
	Хебб	Многослойная прямого распространения	Линейный дискриминантный анализ	Анализ данных Классификация образов
	Соревнование	Соревнование	Векторное квантование	Категоризация внутри класса Сжатие данных
		Сеть ART	ARTMap	Классификация образов
Без учителя	Коррекция ошибки	Многослойная прямого распространения	Проекция Саммона	Категоризация внутри класса Анализ данных
	Хебб	Прямого распространения, соревнование	Анализ главных компонентов	Анализ данных Сжатие данных
		Сеть Хопфилда	Обучение ассоциативной памяти	Ассоциативная память

1.5 Области применения нейросистем

Потенциальными областями применения ИНС являются области, где малоэффективен человеческий интеллект, а традиционные вычисления трудоемки или физически неадекватны (т.е. не отражают или плохо отражают реальные физические процессы и объекты). Т.е. актуальность применения НС многократно возрастает когда появляется необходимость решения плохо формализованных задач.

Основные области применения нейронных сетей и нейрокомпьютеров приведены в таблице 1.3.

Таблица 1.3 - Основные области применения нейронных сетей

Основные области применения	Описание
Проектирование и оптимизация сетей связи	С помощью НС успешно решается важная задача в области телекоммуникаций – нахождение оптимального пути трафика между узлами. Кроме управления маршрутизацией потоков, нейронные сети используются для получения эффективных решений в области проектирования новых телекоммуникационных сетей.
Распознавание речи	С помощью НС решаются многие задачи, связанные с распознаванием речи, например системы для дикторо-независимого речевого управления и т.п.
Управление ценами и производством	НС, предназначенные управлять ценами или для планирования затрат при изготовлении продукции, обнаруживает сложные зависимости между затратами на рекламу, объемами продаж, ценой, ценами конкурентов, днем недели, сезоном и т.д. В результате использования системы осуществляется выбор оптимальной стратегии с точки зрения максимизации объема продаж или прибыли.
Анализ потребительского рынка	НС, прогнозирующая свойства потребительского рынка пищевых продуктов, решает задачу кластеризации
Исследование спроса	НС, позволяющие выявлять сложные зависимости между факторами спроса, прогнозировать поведение потребителей при изменении маркетинговой политики, находить наиболее значимые факторы и оптимальные стратегии рекламы, а также очерчивать сегмент потребителей, наиболее перспективный для данного товара.

Продолжение таблицы 1.3

Анализ страховых исков	НС, предназначенные для выявления в реальном времени подозрительных страховых исков, поступающих в связи с повреждениями автомобилей.
Оценка недвижимости	На основе НС используются данные по оценке недвижимости из обзоров риэлтеровских фирм и списков аукционных цен.
Распознавание символов	Применения НС, например при сортировке писем на почте, при идентификации почерка, номеров автомашин
Прогнозирование	Прогнозирование – важнейший элемент современных информационных технологий принятия решений в управлении.
Информационная безопасность	При обнаружении сетевых атак, биометрических данных, ЦВЗ и т.д.
Обслуживание кредитных карт	НС, разработанные для отслеживания операций с краденными кредитными картами и поддельными чеками, позволяет по частоте сделок и характеру покупок выделить подозрительные сделки и сигнализировать об этом в контролирующие службы. Благодаря данным системам отслеживаются более 500 миллионов счетов 16 крупнейших эмитентов кредитных карт, потери банков от таких операций заметно уменьшились.
Нейросеть в здравоохранении	Системы объективной диагностики слуха у грудных детей (Российская компания НейроПроект).
Обнаружение фальсификаций	Тестирование системы обнаружения показывают, что НС позволяет обнаруживать 50,0% мошеннических случаев, в то время как существовавшая ранее экспертная система – только 14,0 %.

1.6 Программные продукты моделирования нейронных сетей

В свете возрастающего интереса к нейросетевым технологиям появляется все большее число проектов, в основе которых применяется данный подход. Такие проекты, как правило, обусловлены большими размерностями и объемами обрабатываемых данных.

В настоящее время существует множество инструментальных средств для работы с нейронными сетями, которые традиционно включают генерацию начальной структуры сети, ее оптимизацию, обучение,

тестирование и применение. Этот инструментарий можно разделить на три группы:

1. Надстройки для программ прикладных вычислений:

- Matlab_Neural_Network – набор нейросетевых расширений для пакета прикладных вычислений Matlab;
- Statistica_Neural_Networks – набор нейросетевых расширений для пакета прикладной статистики Statistica;
- Excel_Neural_Package – набор библиотек и скриптов для электронных таблиц Excel, реализуют некоторые возможности нейросетевой обработки данных.

2. Универсальные нейросетевые пакеты;

- NeuroSolutions – нейропакет предназначен для моделирования широкого круга искусственных нейронных сетей;
- NeuroPro – менеджер обучаемых искусственных нейронных сетей;
- NeuralWorks – нейропакет, в котором основной упор сделан на применение стандартных нейронных парадигм и алгоритмов обучения.

3. Специализированные:

- Neuroshell Trader – одна из наиболее известных программ создания нейронных сетей для анализа рынков.
- Глаз – используется для обработки аэрокосмической информации.

К недостаткам программных средств первой группы можно отнести: высокую стоимость самой среды, богатую функциями, которые очень маловероятно понадобятся при работе с нейронной сетью; необходимость приобретения самой надстройки; для взаимодействия с пользователем используется интерфейс среды, который не всегда удобно использовать для работы с нейронными сетями.

Пакеты второй группы имеют более специализированный интерфейс, но, как и первые, стоят немалых денег, имеют слишком много функций и сложны для освоения.

Основным недостатком программных средств третьей группы является то, что они предназначены для решения только конкретного класса задач или только для решения конкретной задачи. Причем зачастую методы решения данных задач заложены разработчиком и модификаций не допускают.

Существуют также разработки отдельных авторов, распространяемые бесплатно, но каждая из них имеет один или несколько из нижеперечисленных недостатков: неудобный интерфейс; сильно урезанная функциональность; нестабильность в работе; невозможность модификации заложенных алгоритмов.

1.7 Типы аппаратного обеспечения на основе ИНС

Существует большое количество типов рассматриваемых устройств, однако их можно разделить на три основных класса, которые могут

применяться в зависимости от поставленных и выполняемых задач. *Нейрокомпьютеры (Neurocomputer)*. Представители шестого поколения ПК представляют собой комплексную систему, аппаратные составляющие которой полностью основаны на ИНС. Создание таких систем обосновано при необходимости выполнения обработки информации, требующей высоких вычислительных мощностей.[14]

Разного типа *ускорители и другие карты расширения для ПК (PC accelerators)*. Такие устройства представляют собой стандартные карты расширения для шины, например, ISA или PCI, с тем лишь отличием, что обработку данных осуществляет ИНС. Такие устройства обладают некоторыми преимуществами нейрокомпьютеров, но в более узком или специализированном диапазоне выполняемых задач, а, соответственно, и низком ценовом диапазоне.

Чипы (Chips). Тип аппаратной реализации ИНС, применяемый для построения вышеназванных форм реализации, а также предназначенный для совместного использования с другими стандартными устройствами для расширения свойств последних.

Клеточные библиотеки (Cell libraries). Такой тип предназначен для обеспечения совместной работы специализированного чипа и некоторых дополнительных возможностей и функций, предоставляемых другими устройствами. Широко применяется при построении сложных комплексных систем.

Встроенные микрокомпьютеры (Embedded microcomputers). Такие устройства способны выполнять определенный круг задач с помощью ИНС, но без участия периферийных устройств (клавиатуры, монитора и т. д.). Некоторые ускорители могут содержать обычные перепрограммируемые процессоры, повышение производительности которых обеспечивается распараллеливанием вычислительных повторяющихся операций с помощью ИНС. Отметим, что далее сконцентрируемся на устройствах, в которых функциональные возможности самой ИНС непосредственно осуществлены в аппаратном обеспечении

Практическое исполнение и внедрением ИНС в аппаратные средства. В этом случае можно выделить три широких класса: цифровое, аналоговое и гибридное исполнения. В рамках этих категорий используется различная архитектура и методы для реализации необходимых функций.

Цифровое исполнение

В цифровом исполнении все значения, обрабатываемые нейронной сетью, представлены бинарными словами с характерной длиной слова. К преимуществам цифровой технологии перед аналоговой следует отнести независимость от электромагнитных помех, возможность использования RAM для хранения весовых коэффициентов (в течение неопределенного отрезка времени), хорошо отработанные технологии изготовления, высокая точность в вычислительных операциях, а также легкая интегрируемость в уже существующие системы. Однако в этом случае, как и везде,

присутствуют недостатки, среди которых следует отметить более медленные (хотя и более точные) вычисления, а также проблемы, связанные с конвертацией аналогового сигнала. [14]

В случае цифрового исполнения аппаратное обеспечение на основе ИНС может быть реализовано несколькими типами архитектур, наиболее важные из них мы рассмотрим и приведем соответствующие примеры. *Каскадируемая архитектура.* Рассматриваемая архитектура практически идентична методам построения обычных цифровых процессоров, другими словами, нейронная сеть любого размера и архитектуры строится посредством стандартных блоков. Реализованными примерами такой архитектуры могут служить чип Philips Lneuro, MD1220 от Micro Devices, а также Neuralogix NLX-420 Neural Processor.

Мультипроцессорные чипы. В этом случае подход состоит в размещении в одном чипе множества простейших процессоров. Такие решения могут быть разделены на две группы, известные как SIMD (Single Instruction, Multiple Data) и так называемые систолические сети. В случае SIMD, все процессоры выполняют одну и ту же инструкцию параллельно с вектором данных. Во втором случае каждый процессор неоднократно исполняет один шаг вычислений перед передачей результата следующему (или нескольким) процессору в сети. Примерами SIMD-архитектуры являются чип Inova N64000, содержащий 64 элемента обработки, чип HNC 100NAP, включающий в себя 4 обрабатывающих элемента, Siemens внедрила в свой мультипроцессор MA 16 микрочипов. Такая архитектура предназначена, главным образом, для исполнения различных действий над матрицами.

Архитектура RBF (Radial Basis Function). Согласно этой архитектуре, функционирование сети определяется управлением эталонными векторами, определяющими области, на которые влияют данные при обучении. Преимуществом RBF ИНС является их быстрое обучение и относительно простое построение сетей прямого распространения. К коммерческим изделиям относятся чипы IBM ZICS и Nestor Ni1000. Интересным фактом является также и то, что произведенные в США чипы семейства IBM ZICS были разработаны в Европе.

Другие цифровые проекты. Ряд существующих архитектур не подходят ни под одну из вышеназванных категорий. К примеру, разработка фирмы Micro Circuit Engineering MT19003 NISP, по существу, RISC-процессор (Redu-ced Instruction Set Computer, тип архитектуры микропроцессора, ориентированный на быстрое и эффективное выполнение относительно небольшого набора встроенных команд), осуществляющий семь инструкций, оптимизированных для построения многослойных сетей. Еще одним примером, реализующим другой подход, может служить чип Hitachi Wafer Scale Integration. Чипы этого семейства предназначены для реализации сетей обратного распространения и сетей Хопфилда.

Аналоговое исполнение

К преимуществам этой категории аппаратных средств реализации ИНС следует отнести высокие скорости обработки информации и возможности высокой плотности расположения элементов. Однако тут же дают о себе знать и недостатки сложность в получении высокой точности, обусловленная различиями в компонентах из-за системы допусков при производстве, различные характеры тепловых и электромагнитных помех, искажающих полезный сигнал. Еще одной проблемой является сложность в долгосрочном хранении весовых коэффициентов и организации операций аналогового умножения.

В качестве примера можно привести разработку Intel 8017NW ETANN (Electrically Trainable Analogue Neural Networks), содержащий 64 нейрона и 10280 весовых коэффициентов. ИНС, реализованная в продукте Synaptics Silicon Retina, обрабатывает изображение, моделируя процессы, происходящие в сетчатке глаза. Подход заключается в создании аналогового исполнения, где ИНС пытается наиболее точно воспроизвести поведение биологических нейронов. Реализованные аналоговые нейросети представляют набор компонентов, размеры которых меньше размеров биологического нейрона, и предполагается, что вышеназванные недостатки компенсируются взаимосвязями между аналоговыми нейронами.

Гибридное исполнение

Как понятно из названия, эта категория представляет собой комплекс вышерассмотренных систем. Разработчики таких проектов пытаются получить от таких систем преимущества аналогового и цифрового исполнений. По большей части это достигается путем связи между устройствами и датчиками посредством цифровой составляющей, а обработка полностью или частично реализуется аналоговыми методами. В качестве примера приведем чип Bellcore CLNN-32, который хранит весовые коэффициенты в цифровой форме, а производит моделирование ИНС, используя аналоговую схему. Существуют проекты, в которых весовые коэффициенты хранятся в конденсаторах, периодически подзаряжающихся от внутренних источников тока. Также примерами гибридных систем могут служить SU3232 Synapse и NU32 Neuron, разработанные в лабораториях Neural Semiconductor, и RN-100, представленный Ricoh.

Ученые из University of Dusseldorf изучают дисфункцию в искусственных условиях замороженных мозговых клеток крыс, путем размещения их на Microelectrode Arrays (MEAs), параллельно исследуя их реакцию на различные фармакологические препараты

В дальнейшем развитие аппаратных средств на основе ИНС может пойти следующими путями:

1. Путем усовершенствования методов для реализации нейросетевых методов на FPGA (Field Programmable Gate Array, ПЛИС, Программируемая Логическая Интегральная Схема), VLSI (Very Large Scale Integration, СБИС, уровень интеграции, при котором количество элементов на одной микросхеме исчисляется тысячами и миллионами).

2. Благодаря исследованиям и внедрению инновационных алгоритмов построения ИНС, которые осуществимы аппаратными средствами. 3. Разработкой промышленного стандарта нейросетевых алгоритмов высокого уровня в промышленности.

Первые два пункта более-менее понятны, поясним, что подразумевается в последнем. Разработанные методы должны легко адаптироваться к нуждам промышленности, достаточно просто реализовываться. Но для этого необходимо специализированное ПО с полным набором нейросетевых функций (для цифрового, аналогового и гибридного исполнений). Немаловажно и исследование методов внедрения ИНС в уже существующие системы, создания на их основе гетерогенных систем. Вообще говоря, цепь обработки информации может начинаться с аналоговых датчиков и заканчиваться аналоговыми исполнительными устройствами, или система может быть полностью цифровой, в любом случае необходима оптимизация на уровне системы, а не отдельных ее составляющих.

А согласно указанным направлениям развития, все более вероятен переход на новые технологии. Отметим, что Япония по скорости внедрения новых интеллектуальных технологий шагает далеко впереди, обогнав как страны СНГ, так и страны Европы. Особенно это хорошо заметно в области бытовой электроники, где чипы на основе нейронных сетей устанавливаются в микроволновые печи (Sharp), пылесосы, фото- и видеокамеры. Приведем краткий список фирм, уже применяющих ИНС в их аппаратном исполнении: Ericsson (Англия и Швеция), Philips Research (Нидерланды), Siemens AG Munich, Siemens/Nixdorf Bonn, 3M Laboratories (Europe) GmbH Neuss, XIONICS Document Technologies GmbH Dortmund, Robert Bosch GmbH Reutlingen, Spectrum Microelectronics Siek, Fiat, Domain Dynamics Ltd.

В последнее время большое распространение получили методы, основанные на математическом аппарате искусственных нейронных сетей (ИНС). Одно из направлений обработки изображений связано с ИНС, которые эффективно фильтруют данные (сеть Хопфилда), распознают графические образы (сеть Хемминга, многослойный персептрон), кластеризуют объекты (сеть Кохонена), а также выполняют другие немаловажные функции. Существенный вклад в развитие теории нейронных сетей и нейрокомпьютеров внесли А.Н. Горбань, А.И. Галушкин, Е.М. Миркес, В.Г.Редько, В.Г. Яхно и др.

Реализация программного обеспечения может осуществляться в виде пакетов прикладных программ, библиотек прикладных или системных программных средств. На данный момент существует множество программных пакетов, с помощью которых можно проектировать, настраивать и применять ИНС для решения различных задач, например, Neurooffice, Neuro Emulator, BrainMaker, Neural 10, Neural Planner, Mathlab и др. Специально для обработки данных ДЗЗ разработаны современные программные комплексы: ENVI (Environment for Visualizing Images), ER

Mapper, ERDAS Imagine. Кроме этого имеются библиотеки для обработки и распознавания, такие как OpenCV, LTI, VXL и др. Однако перечисленные нейропакеты и программные комплексы не рассчитаны на потоковую обработку снимков и на использование параллельных вычислителей. Решение же практических задач требует быстрого анализа последовательности изображений большого формата, доставляемых СТЗ. С появлением российских мультипроцессорных систем стала актуальной задача построения программных систем, ориентированных на параллельную обработку снимков с целью ускорения процессов выделения и распознавания требуемых объектов.

1.8 Применение нейросетей как систем обнаружения вторжений

Системы обнаружения сетевых вторжений (сов) и обнаружение признаков компьютерных атак на информационные системы уже давно являются одной из линий защиты, необходимых для информационных систем. В настоящее время системы обнаружения вторжений обычно представляют собой программные или аппаратные решения, автоматизируя процесс мониторинга событий, происходящих в компьютерной или сетевой системе, а также независимо анализируя эти события, чтобы найти признаки проблем безопасности.

Из-за способности искусственных нейронных сетей идентифицировать сложные отношения между входными и выходными данными в процессе обучения эти данные явно не являются, поэтому привлекательным инструментом для решения проблемы является защита компьютерной информации. Таким образом, повышение эффективности идентификации событий информационной безопасности через ИНС в рамках работы системы и статистических методов анализа данных является неотложной научной и технической задачей.

Из-за способности искусственных нейронных сетей в процессе обучения улучшать сложные связи между входами и выходами, которые явно отсутствуют, инструмент для получения компьютерной защиты информации представляет интерес. Таким образом, повышение эффективности обнаружения инцидентов информационной безопасности посредством инициативы, взаимодействующей со статистическими методами анализа данных, в контексте функционирования системы, представляет собой актуальную научную и техническую задачу.

При разработке систем обнаружения вторжений серьезной проблемой является выбор времени установки для определения удаленной атаки. Чтобы обойти эту проблему, можно использовать методы сокащения размерности. Метод позволяет отбросить ненужные параметры трафика или же выявить некоторые из наиболее важных параметров.

С точки зрения математики задачу сокращения размерности можно предоставить: дана p-мерная переменная $x = (x_1,x_2,...,x_p)^T$. Нужно найти пространство меньшей размерности, в котором переменная $s = (s_1,s_2,...,s_k)^T$, k

$\leq p$ отражает содержание исходных данных в соответствии с некоторым критерием.

Существует два метода редукции мирового класса: линейные методы и нелинейные методы. Для линейных методов результаты каждого элемента каждой $k \leq p$ компоненты будет линейная комбинация исходныхпеременных:

$s_i = w_{i;1}x_1 + \ldots + w_{i;p}x_p$,

где $i=1 \ldots k$. $s = Wx$.

W_{kxp}– матрица весов линейных преобразований.

Линейные методы более наглядные и значительно проще в использовании, чем более современные методы, основанные на нелинейных преобразованиях.

Отдельно можно методы, ограничивающиеся рассмотрением статистических моментов второго порядка. Метод расчета прост и включает в себя только классические операции с матрицами, которые не требуют декодирования в области параметров преобразования. Это первый и самый продвинутый метод, самый известный классический метод: метод главных компонент и факторный анализ.

Метод главных компонент

Метод Главных Компонент (Principal components analysis, PCA) является одним из основных способов уменьшения размерности данных, при котором теряется наименьшее количество информации [16, 17]. Вычисление главных компонент сводится к вычислению собственных векторов и собственных значений ковариационной матрицы исходныхданных.

Задача анализа главных компонент имеет четыре базовых версии:

- аппроксимировать данные линейными многообразиями меньшей размерности;
- найти подпространства меньшей размерности, в ортогональной проекции на которые разброс данных (т.е. среднеквадратичное отклонение от среднего значения)максимален;
- найти подпространства меньшей размерности, в ортогональной проекции на которые среднеквадратичное расстояние между точкамимаксимально;
- для данной многомерной случайной величины построить такое ортогональное преобразование координат, что в результате корреляции между отдельными координатами обратятся вноль.

Идея метода главных компонент состоит в уменьшении размерности данных путём поиска нескольких ортогональных линейных комбинаций (главных компонент).

Факторныйанализ

Факторный анализ (Factor analyze, FA), также, как и метод главных компонент, является методом второго порядка, работающий с линейными преобразованиями [18, 19]. В факторном анализе используется предположение, что степень отклонения переменных зависит от неизвестных, и часто неизмеримых, общих факторов. Целью метода

факторного анализа является выявление таких зависимостей, которые могут быть использованы для сокращения размерностиданных.

Случайный вектор x с нулевым средним значением удовлетворяет k-фактор модели, если выполняется следующее условие: x = Λf + u, где Λp×k – это матрица констант, fk и up – это, соответственно, скрытые факторы и специфические факторы. Кроме того необходимо, чтобы все скрытые факторы не коррелировали между собой и являлись стандартизованными таким образом, чтобы их дисперсия была равна 1. При помощи данного метода выявляются скрытые зависимости между исходными величинами, в результате.

Метод независимых компонент

Метод независимых компонент – это метод высшего порядка, который ищет наиболее статистически независимые линейные проекции, не обязательно ортогональные между собой [20]. Метод независимых компонент можно считать обобщением концепций метода главных компонент и метода поиска оптимальных проекций. Бесшумный метод независимых компонент является частным случаем поиска оптимальной проекции, с использованием независимости в качестве«интересности» в определении индекса проекции. Модель метода независимых компонент с шумом эквивалентна модели факторного анализа, допускающей негауссовость данных. Получение модели состоит из двух этапов: определение целевой функции (функции затрат) и алгоритм оптимизации целевой функции. Целевые функции могут быть разделены на две группы: многокомпонентные контрастные функции, оценивающие все p независимых компонент сразу, и однокомпонентные контрастные функции, производящие оценку одного независимого компонента.

Метод поиска наилучшей проекции

Поиск наилучшей проекции – это также линейный метод, но он, в отличие от метода главных компонент и факторного анализа, основывается на статистиках высоких порядков, что полезно при негауссовых данных [21]. Этот метод требует большего объёма вычислений, чем методы второго порядка.

Метод многомерного шкалирования

Для n выборок в p-мерном пространстве и $n \times n$ матрицы близости между выборками метод многомерного шкалирования создаёт k-мерное, $k \leq p$, представление выборок таким образом, что расстояния между точками в новом пространстве отражают близость в оригинальных данных [22, 23]. Близость измеряет степень сходства между выборками, т.е. меру длины: чем больше схожи две выборки, тем меньше расстояние между ними. Наиболее известные измерения расстояний – это евклидово расстояние (L_2 нормы), расстояние Манхэттена (L_1, абсолютная норма) и максимум-норма. Результаты многомерного шкалирования являются неоднозначными относительно перемещений, вращений иотражений.

Многомерное шкалирование обычно используется для преобразования

данных в двумерное или трехмерное пространство с дальнейшей визуализацией результата с целью выявления скрытой структуры в данных.

Генетические и эволюционныеалгоритмы

Генетические и эволюционные алгоритмы –-это алгоритмы поиска, которые используются для решения задач оптимизации и моделирования путем последовательного выбора, объединения и поиска изменений параметров с использованием механизмов, аналогичных биологической эволюции

Необходимо отметить, что применение нейросетевых технологий не всегда возможно и сопряжено с определенными проблемами и недостатками:

1. Необходимо как минимум 50, а лучше 100 наблюдений для создания приемлемой модели. Это достаточно большое число данных, и они далеко не всегда доступны. При дефиците информации модели ИНС строят в условиях неполных данных, а затем проводят их последовательное уточнение.

2. Построение нейронных сетей требует значительных затрат труда и времени для получения удовлетворительной модели. Необходимо учитывать, что излишне высокая точность, полученная на обучающей выборке, может обернуться неустойчивостью результатов на тестовой выборке — в этом случае происходит «переобучение» сети. Чем лучше система адаптирована к конкретным условиям, тем меньше она способна к обобщению и экстраполяции и тем скорее может оказаться неработоспособной при изменении этих условий. Расширение объема обучающей выборки позволяет добиться большей устойчивости, но за счет увеличения времени обучения.

Разработка новых методов и средств защиты вычислительных систем от сетевых является актуальной. В данном дипломном предлагается нейросетевая модель фильтрации входящего в вычислительную систему трафика. Исследование предполагает построение адаптивной нейросетевой системы, представляющей собой распознавание сетевого соединения, как атаку или не угрожающее соединение. Для проектирования искусственной нейронной сети был использован пакет Neural Network Toolbox из MATLAB 8.6 (R2019a).

2 Сетевые атаки и обучение нейросети.

2.1 Сетевые атаки

Сетевая атака — это действие злоумышленика, направленное на получение контроля над определенной сетью путем присвоения прав администратора. Конечной задачей хакера является дестабилизация сайтов и серверов, вывод их из строя, получение личных данных каждого пользователя сети. Трудность выявления проведения удалённой атаки и относительная простота проведения из-за избыточной функциональности современных систем выводит этот вид неправомерных действий на первое место по степени опасности и препятствует своевременному реагированию на осуществлённую угрозу, в результате чего у нарушителя увеличиваются шансы успешной реализации атаки.

Сетевые атаки можно классифицировать по разным критериям. Так по характеру воздействия можно выделить пассивные – атаки, не оказывающие прямого воздействия на работу системы, и активные – атаки, которые напрямую влияют на работу системы, что может быть нарушением работоспособности, изменение конфигурации, кража данных и прочее. По цели воздействия сетевые атаки могут быть направлены на нарушение целостности, доступности или конфиденциальности системы.

Существуют такие виды сетевых атак, как mailbombing, переполнение буфера, сетевая разведка, IP-спуфинг, Man-in-the-Middle, XSS-атака, DDOS-атака и другие.

2.2 База NSL-KDD

В дипломной работе рассматривалась база данных атак NSL-KDD, которая содержит более 490 тысяч записей. В этой базе содержались как нормальные вектора, так и вектора аномальной активности. Аномальная активность представляет собой атаку.

Каждая запись в базе данных NSL-KDD представляет собой образ сетевого соединения, т.е. последовательности TCP пакетов за некоторое время, в течение которого данные передаются от IP-адреса источника к IP-адресу получателя по определенному протоколу.

Таблица 2.1 – Типы и классы атак.

Типатаки	Количествоатак	Классатаки
back	956	DOS
land	18	DOS
neptune	41214	DOS
pod	201	DOS
smurf	2646	DOS
teardrop	892	DOS
buffer_overflow	130	U2R
loadmodule	72	U2R
perl	34	U2R

Продолжение таблицы 2.1

rootkit	30	U2R
ftp_write	43	R2L
guess_passwd	1231	R2L
imap	126	R2L
multihop	254	R2L
phf	7	R2L
spy	3	R2L
warezclient	890	R2L
warezmaster	205	R2L
ipsweet	3599	Probe
nmap	1493	Probe
portsweep	2931	Probe
satan	3633	Probe
normal	67343	-

Эта база атак содержит 41 параметр, описание которых приведено в таблице.

Таблица 2.2 – Описание параметров

Номер	Название параметра	Описание	Тип
1	duration	Продолжительность соединения (секунды)	Непрерывный
2	protocol_type	Протокол транспортного уровня	Дискретный
3	service	Сервис прикладного уровня	Дискретный
4	flag	Статус соединения	Дискретный
5	src_bytes	Входящий поток, байт	Непрерывный
6	dst_bytes	Исходящий поток, байт	Непрерывный
7	land	1 если адреса источника и получателя совпадают, иначе 0	Дискретный
8	wrong_fragment	Число неправильных фрагментов	Непрерывный
9	urgent	Число срочных пакетов	Непрерывный
10	hot	Число «горячих» индикаторов	Непрерывный
11	num_failed_logins	Число неудачных попыток входа	Непрерывный
12	logged_in	Успешный вход	Дискретный
13	num_compromised	Число скомпроментированных состояний	Непрерывный
14	root_shell	Доступ с административными полномочиями	Дискретный
15	su_attemped	1 если "su root", иначе 0	Дискретный
16	num_root	Число попыток доступа с правами администратора	Непрерывный
17	num_file_creations	Число операций создания файла	Непрерывный
18	num_shells	Число попыток использования командной строки	Непрерывный

Продолжение таблицы 2.2

19	num_access_files	Число операций с файлами контроля доступа	Непрерывный
20	num_outbond_cmds	Число исходящих команд в ftp сеансе	Непрерывный
21	is_host_login	1 если соединение из списка серверов, иначе 0	Дискретный
22	is_guest_login	1 если соединение из гостевого списка, иначе 0	Дискретный
23	count	Число соединений на тот же сервер за 2 секунды	Непрерывный
24	srv_count	Число соединений на тот же сервис за 2 секунды	Непрерывный
25	serror_rate	Процент соединений, у которых ошибка с флагом SYN	Непрерывный
26	srv_serror_rate	Процент соединений, у которых ошибка с флагом SYN	Непрерывный
27	rerror_rate	Процент соединений, у которых флаг REJ	Непрерывный
28	srv_rerror_rate	Процент соединений, у которых флаг REJ	Непрерывный
29	same_srv_rate	Процент соединений к той же службе	Непрерывный
30	diff_srv_rate	Процент соединений к различным службам	Непрерывный
31	srv_diff_host_rate	Процент соединений к различным хостам	Непрерывный
32	dst_host_count	Число соединений на тот же сервер за 2 секунды	Непрерывный
33	dst_host_srv_count	Число соединений на тот же сервис за 2 секунды	Непрерывный
34	dst_host_same_srv_rate	Процент соединений к той же службе	Непрерывный
35	dst_host_diff_srv_rate	Процент соединений к различным службам	Непрерывный
36	dst_host_same_src_port_rate	Процент соединений с одинаковым портом источника	Непрерывный
37	dst_host_srv_diff_host_rate	Процент соединений к различным хостам	Непрерывный
38	dst_host_serror_rate	Процент соединений, у которых ошибка с флагом SYN	Непрерывный
39	dst_host_srv_serror_rate	Процент соединений, у которых ошибка с флагом SYN	Непрерывный
40	dst_host_rerror_rate	Процент соединений, у которых флаг REJ	Непрерывный
41	dst_host_srv_rerror_rate	Процент соединений, у которых флаг REJ	Непрерывный

Из базы NSL-KDD было выбрано 100 тысяч записей для обучения нейросети. Из сетевых атак были выбраны только DoS-атаки.

DoS-атака (Denial of Service) – это атака на вычислительную систему с целью довести её до отказа в обслуживании, то есть создание таких условий, при которых пользователи системы не могут получить доступ к предоставляемым системой ресурсам, либо этот доступ затруднён. В ходе атак DoS могут использоваться обычные Интернет-протоколы, такие как TCP и ICMP (InternetControlMessageProtocol). Большинство атак DoS опирается 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 этого типа проводится одновременно через множество устройств, мы говорим о распределенной атаке – DDoS [3].

Таблица 2.3 – Параметры DoSатак

Класс атаки	Тип атаки	Количество параметров	Параметры
DOS	back	6	src_bytesdst_byt es count srv_countdst_ho st_count dst_host_srv_count
	neptune	7	count srv_countserror_rates rv_serror_ratedst_hos t_countdst_host_serro r_rate dst_host_srv_serror_rate
	smurf	5	count srv_countdst_hos t_count dst_host_same_src_port_rate dst_host_serror_rate
	teardrop	4	count srv_countdst_hos t_count dst_host_same_src_port_rate

2.3 Обучение нейросети в Matlab

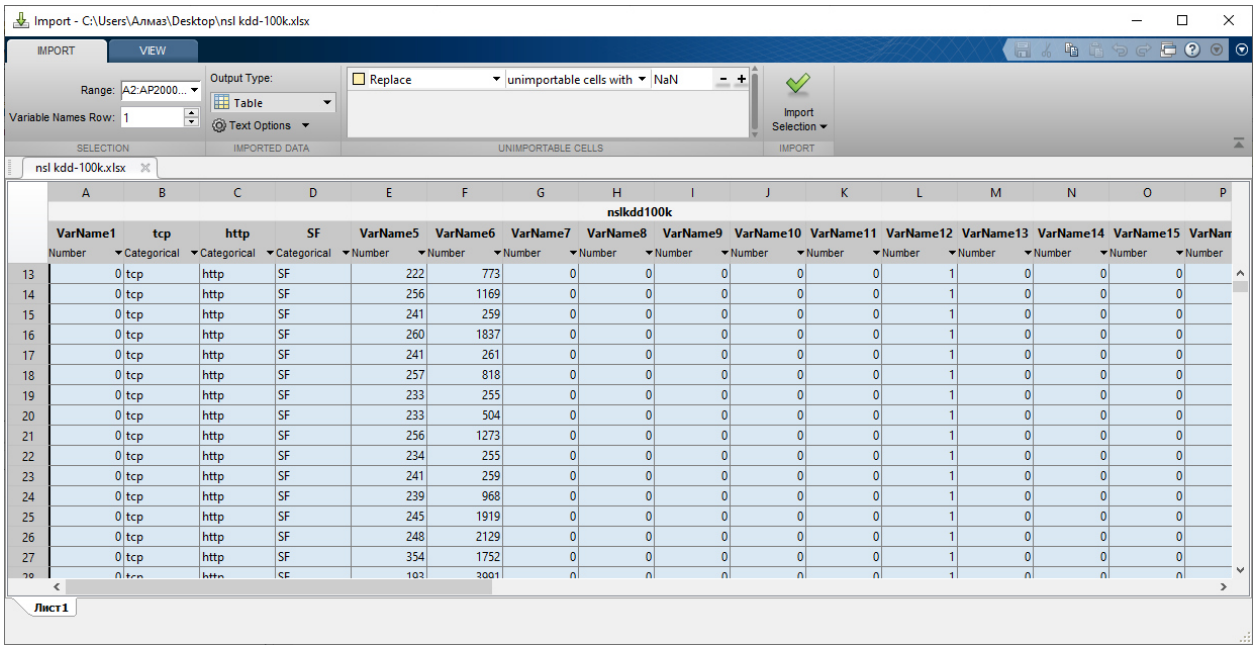

Рисунок 2.1 – Импорт базы данных атак в Matlab

В качестве входных данных будут выступать параметры соединения, а выходные данные – описание соединения: нормальное оно или же представляет собой атаку.

После этого используем утилиту nnstart. Данная утилита позволяет вызвать окошко выбора следующих утилит для дальнейшей работы с нейросетями.

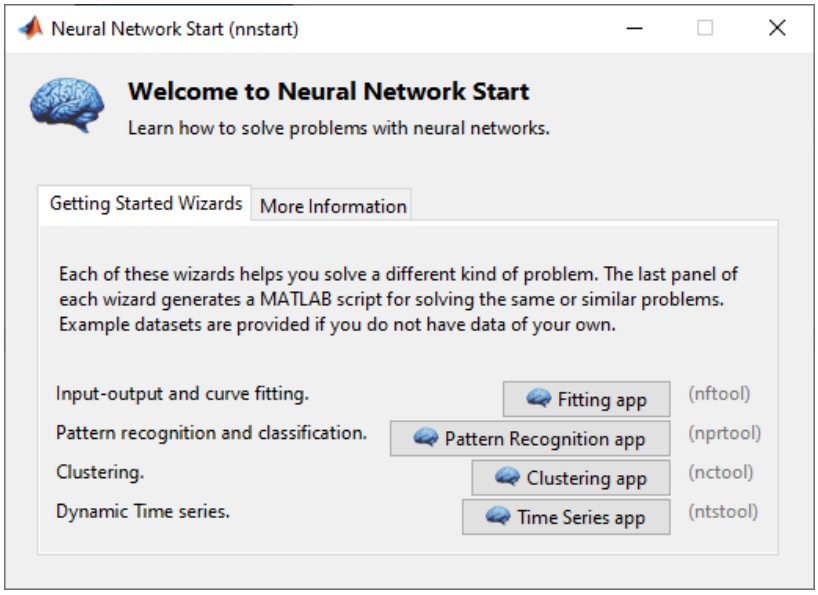

Рисунок 2.2 – Утилита nnstart

Использую утилиту ntftool. Она позволяет конфигурировать двухслойную нейросеть с прямой связью. В качестве функции активации используется сигмоида.

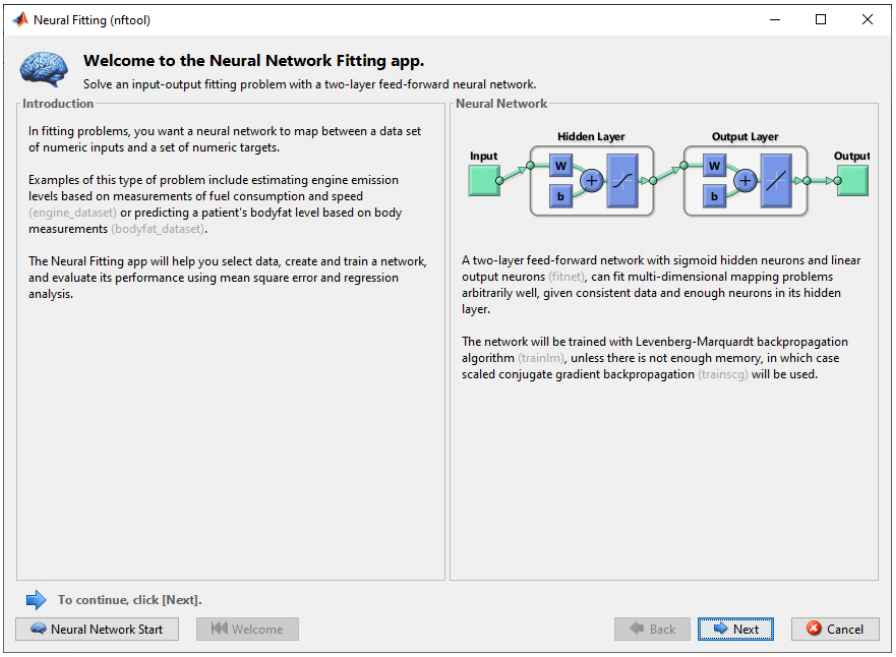

Рисунок 2.3 – Окно ntftool

Следующим шагом был выбор входных данных (inputs) и образцов соответствия (targets). В качестве входных данных были выбраны параметры соединения, а в качестве образцов –тип соединения. Так как матлаб в качестве образцов принимает только числовые данные, то названия типов были изменены в числа: 0 – нормальное соединение, 1 – атака.

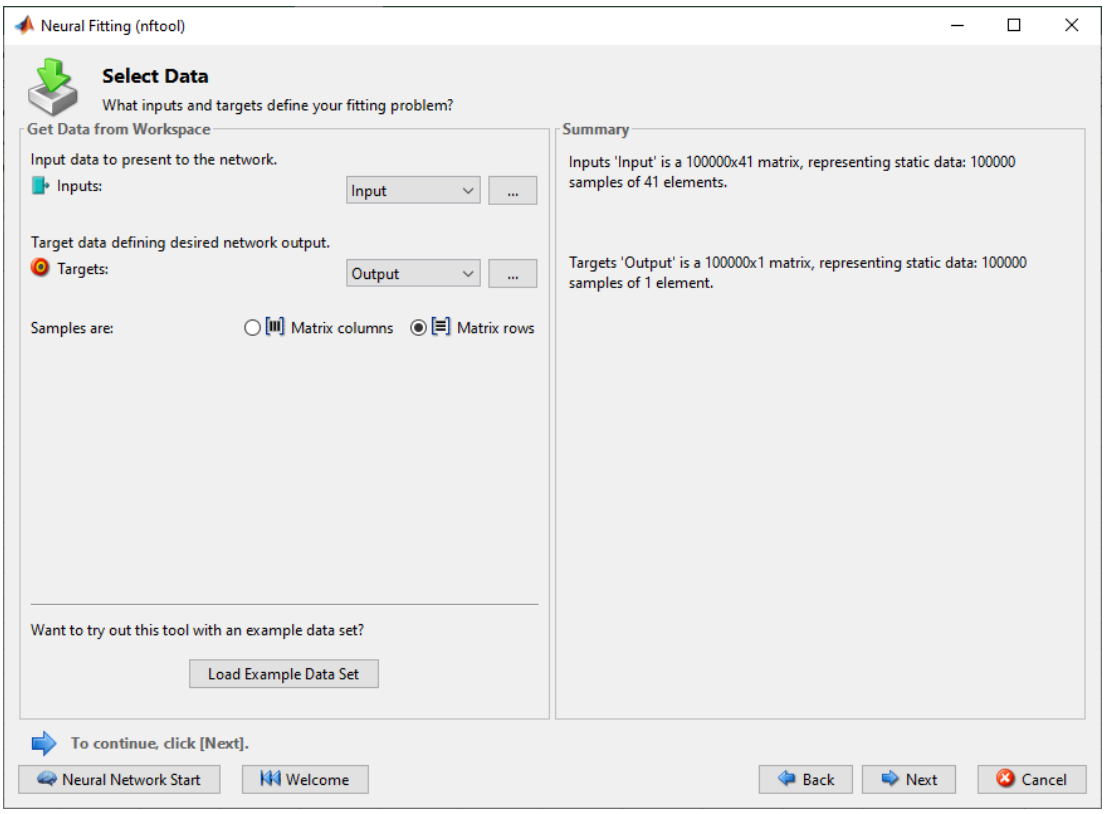

Рисунок 2.4 – Определение входных и выходных данных

Для самого обучения было выбрано 70 тысяч записей, что составляет 70%, для валидации – 15 тысяч записей, что составляет 15% и 15 тысяч записей для проверки, что также составляет 15%/

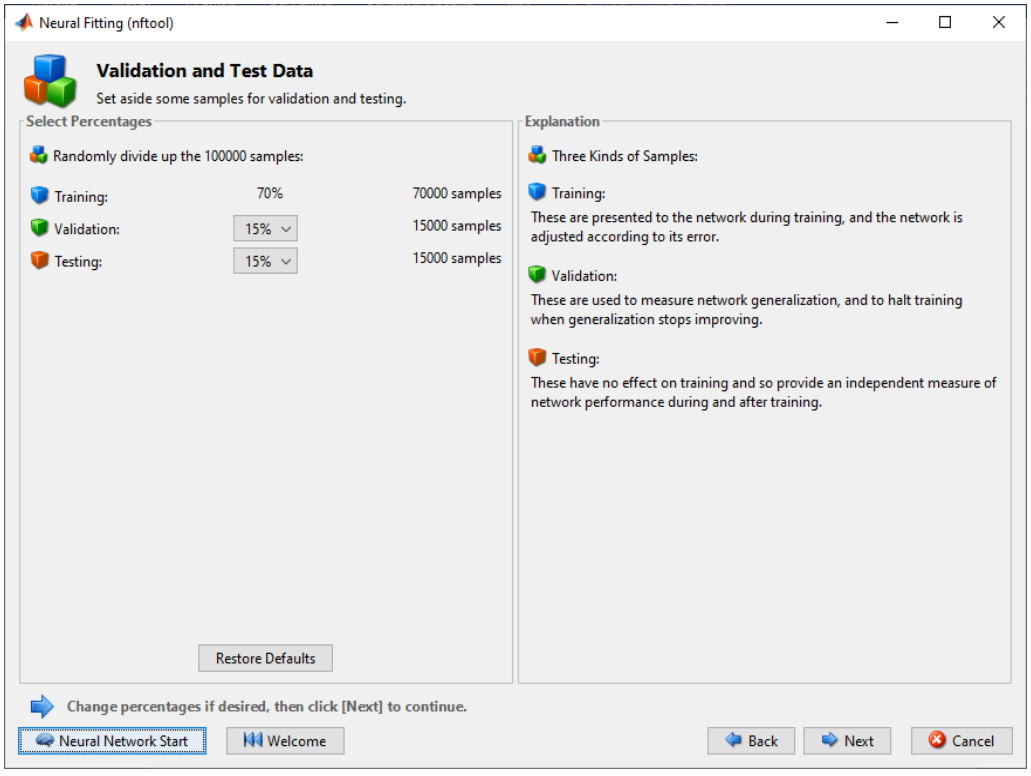

Рисунок 2.5 – выбор количества записей для обучения

В скрытом слое нейросети был выбран 21 нейрон

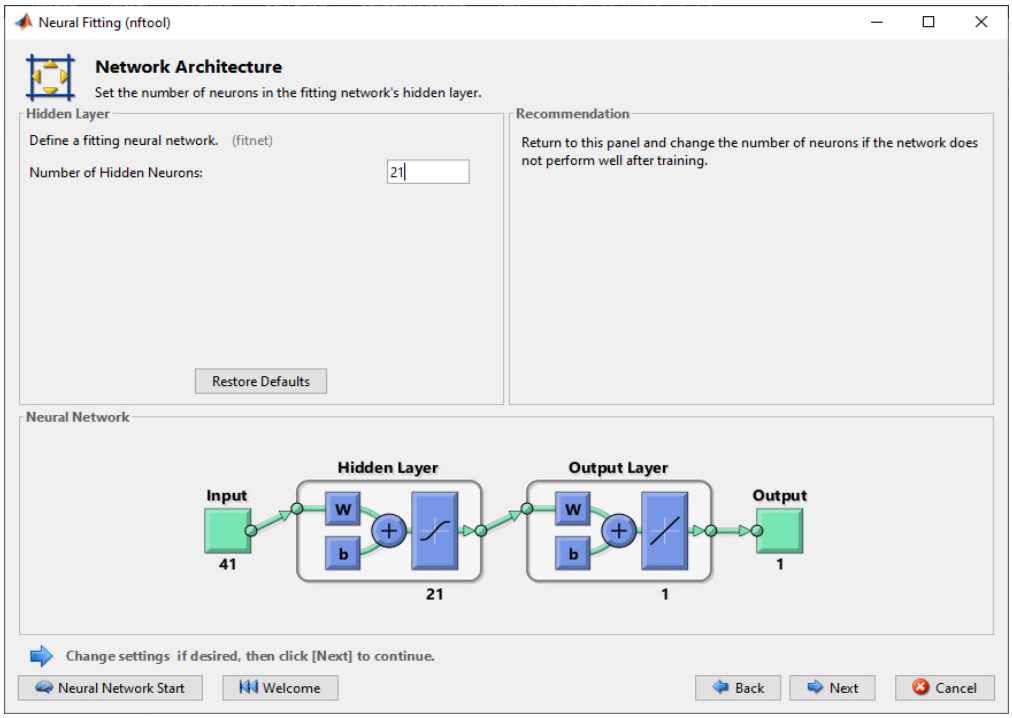

Рисунок 2.6 – Выбор числа нейронов

При выборе алгоритма обучения был выбран метод Левенберга-Марквардта.(Levenberg-Marquardt) Данный алгоритм предназначен для оптимизации параметров нелинейных регрессионных моделей. В данном методе предполагается, что в качестве критерия оптимизации используется среднеквадратичная ошибка модели на обучающей выборке. Алгоритм заключается в последовательном приближении заданных начальных значений параметров к искомому локальному оптимуму [4].

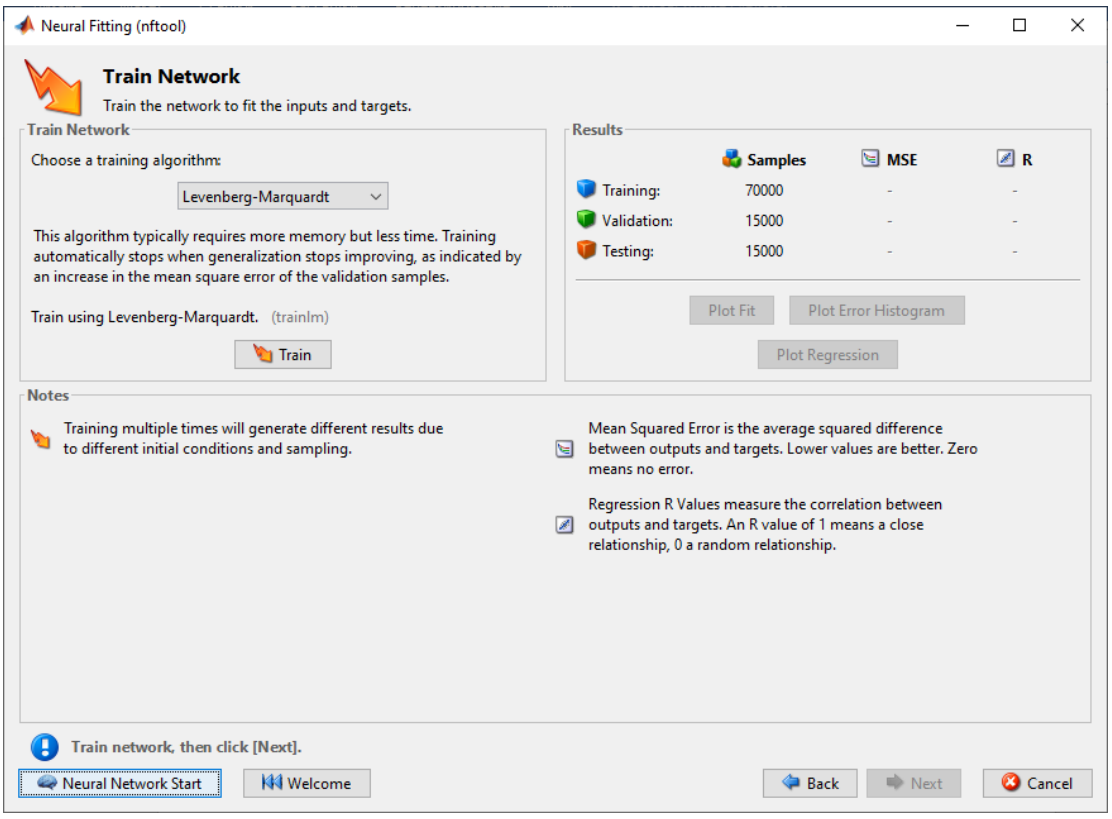

Рисунок 2.7 – Окно обучения

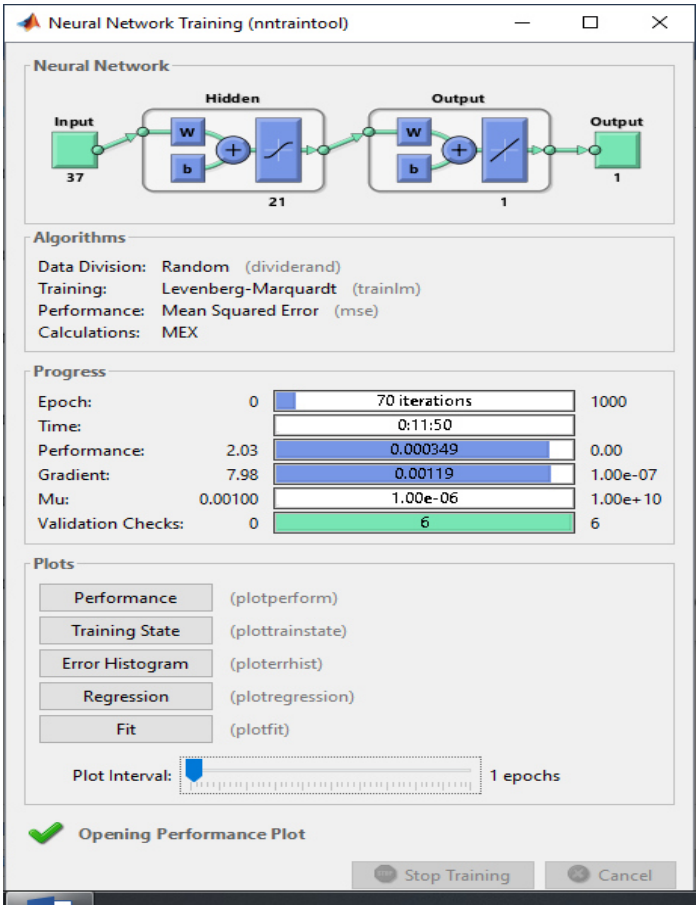

Рисунок 2.8 – Результат обучения нейросети с 21 нейроном

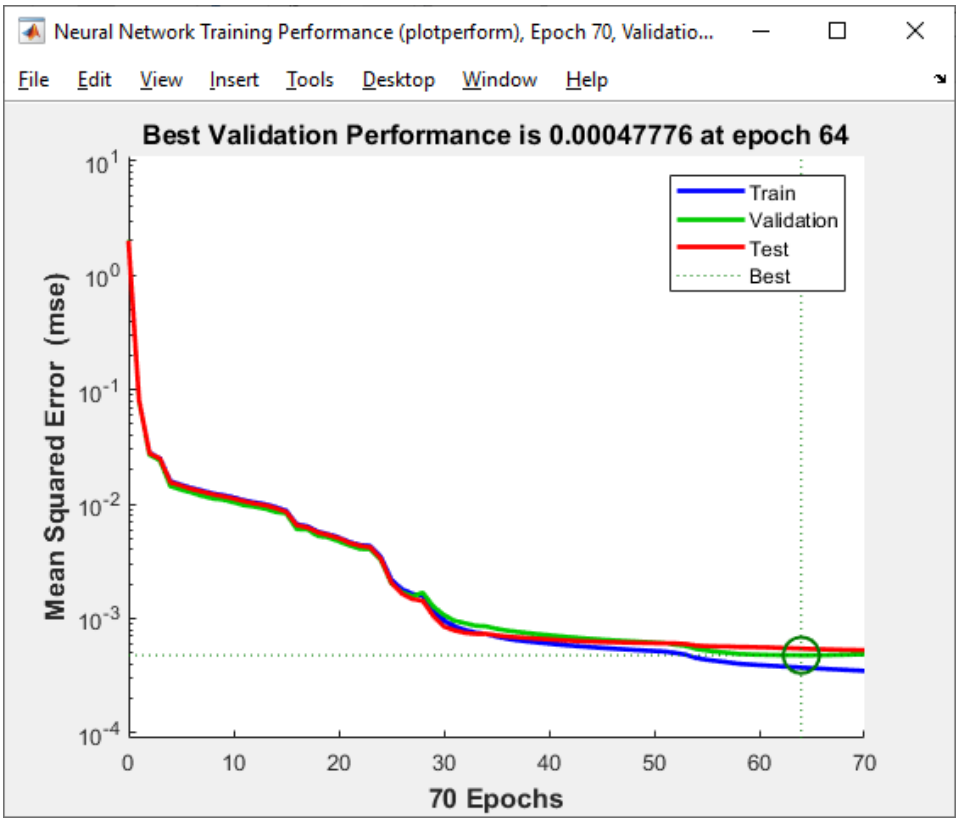

Рисунок 2.9 – График результата нейросети с 21 нейроном

Для определения наилучшего результата буду менять число нейронов и сравнивать полученные результаты. С каждым последующим обучением число нейронов в скрытом слое нейросети будет сокращаться на 1. При этом параметры самого обучения меняться не будут, оставаясь одинаковыми: 70% записей на обучение, 15% записей на валидацию и 15% на тестирование.

Определяю результаты с 20 нейронами.

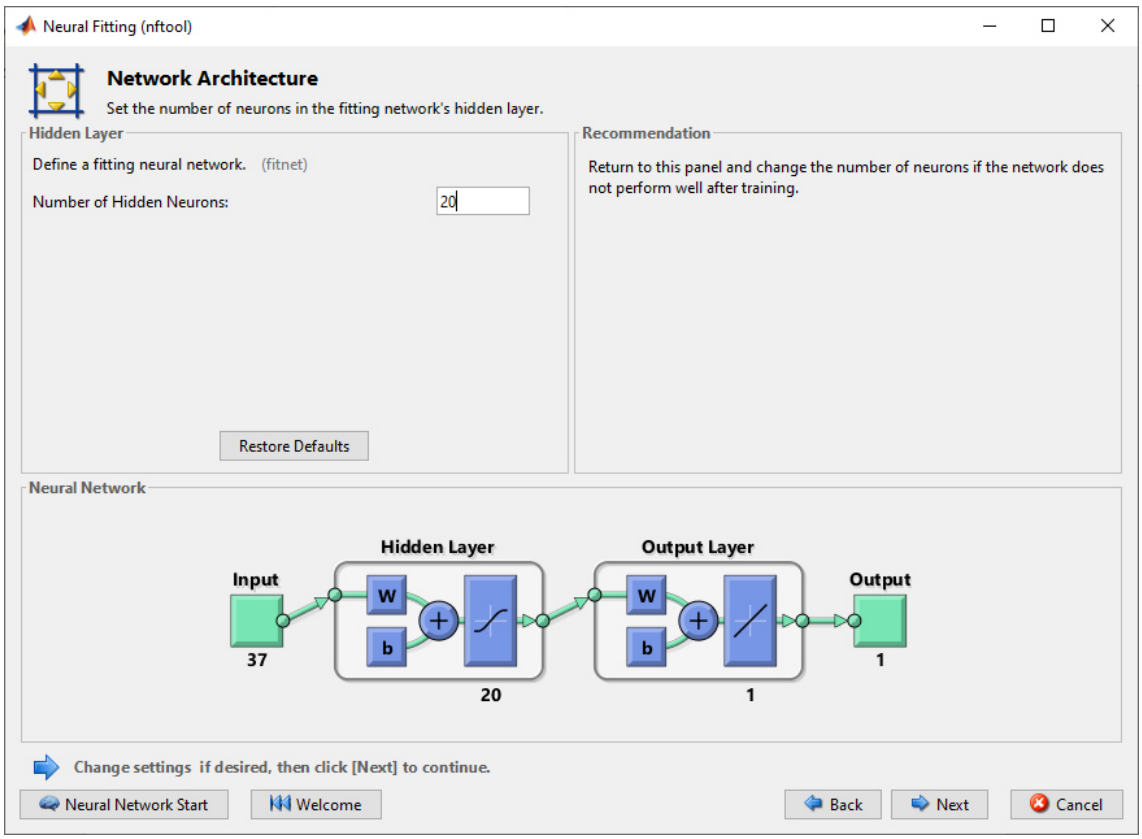

Рисунок 2.10 – Выбор 20 нейронов

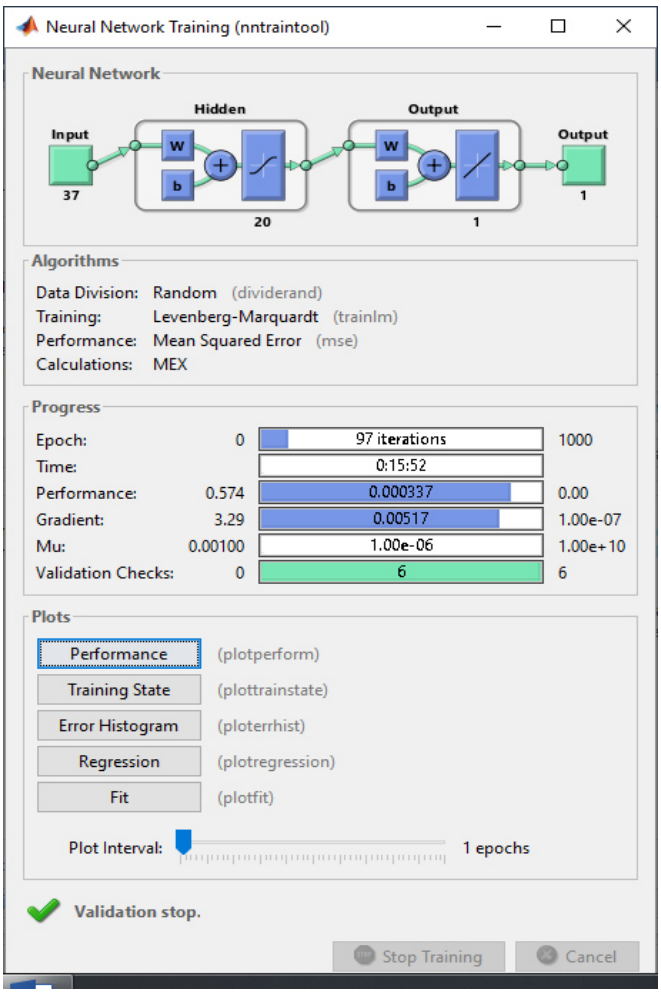

Рисунок 2.11 – Результат обучения нейросети с 20 нейронами

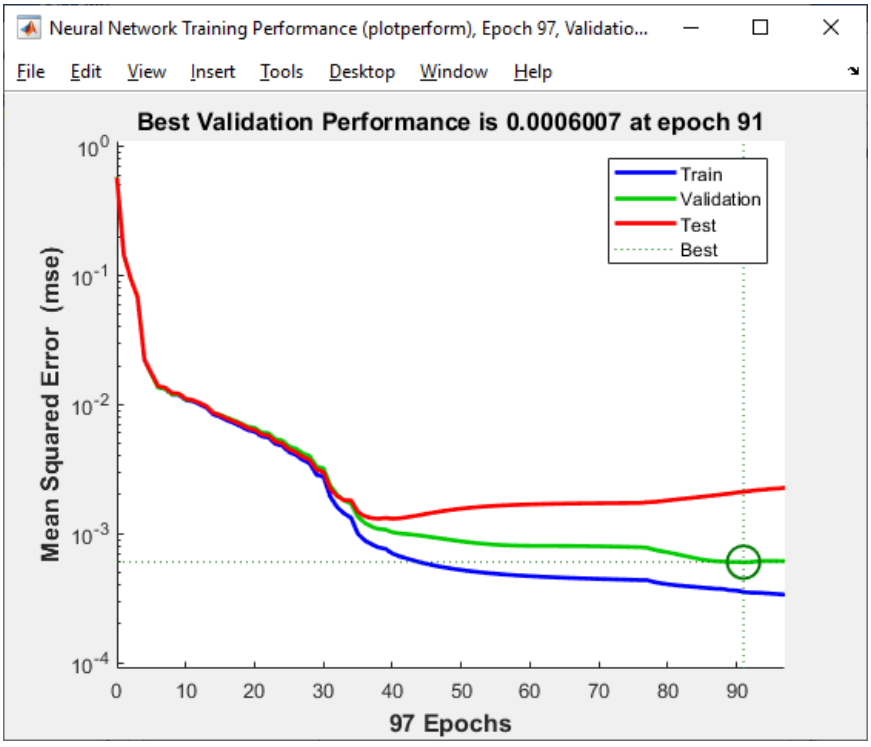

Рисунок 2.12 – График результата нейросети с 20 нейронами

Обучаю сеть с 19 нейронами.

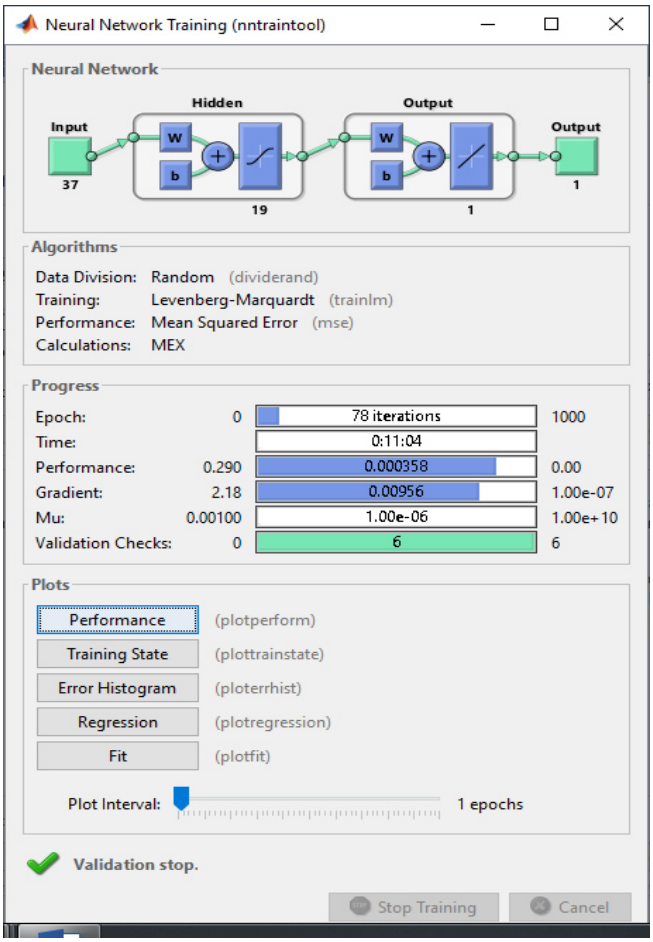

Рисунок 2.13 – Результат обучения нейросети с 19 нейронами

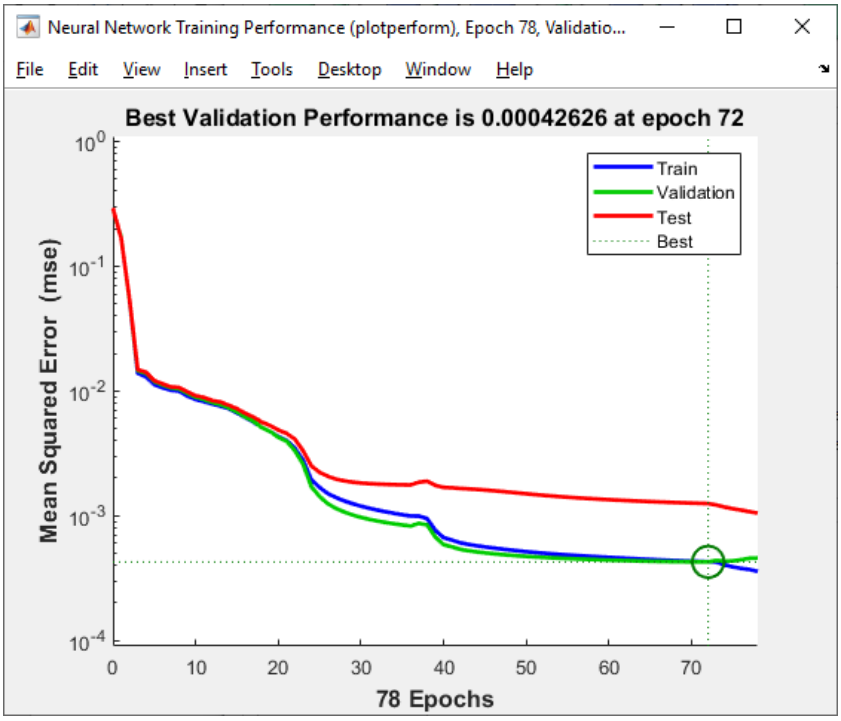

Рисунок 2.14 – График результата нейросети с 19 нейронами

Обучение с 18 нейронами

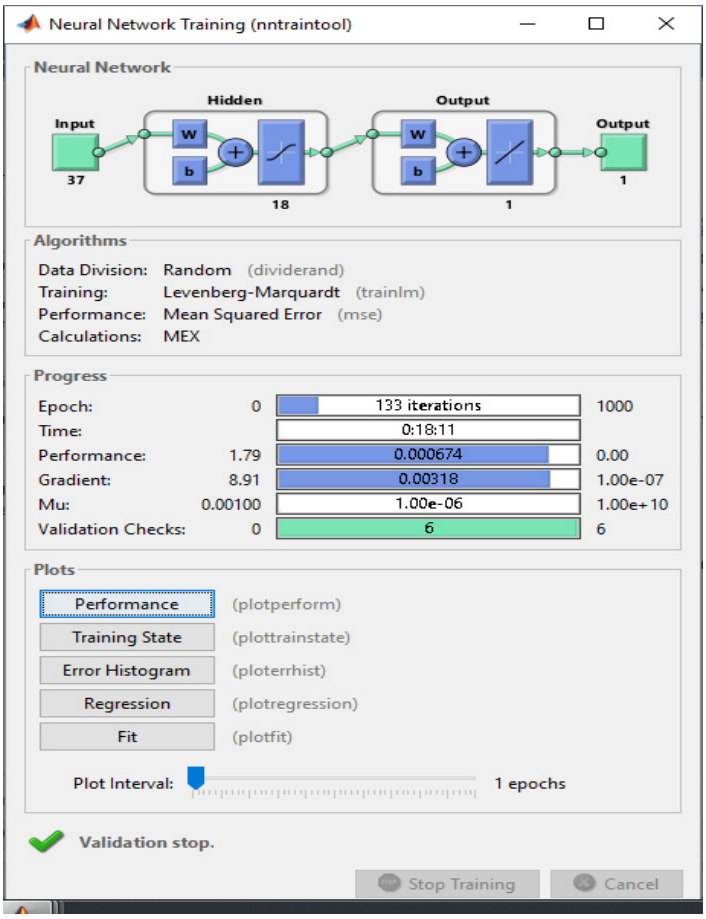

Рисунок 2.15 – Результат обучения с 18 нейронами

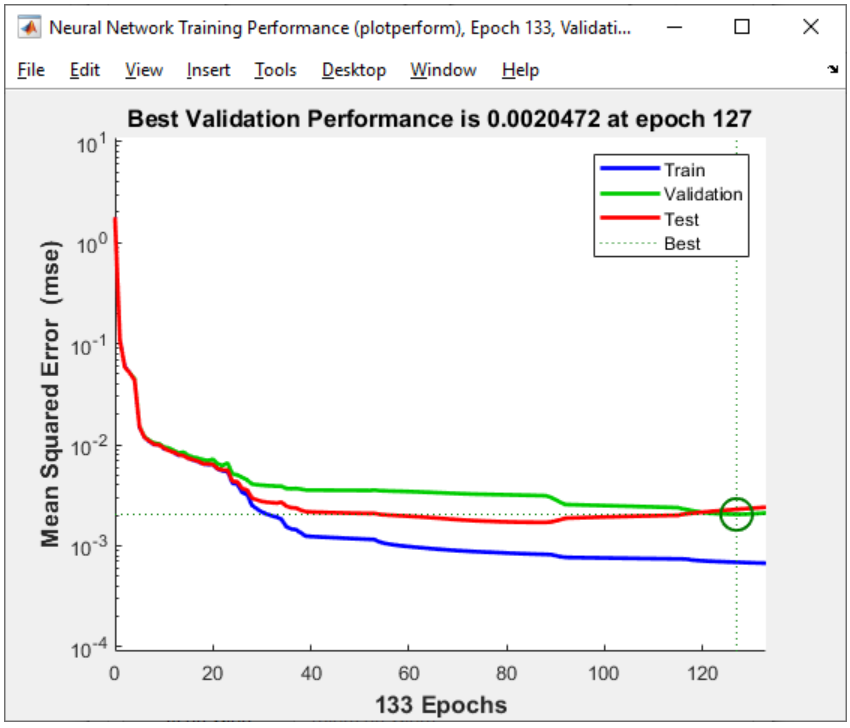

Рисунок 2.16 – График результата обучения нейросети с 18 нейронами

Обучение с 17 нейронами

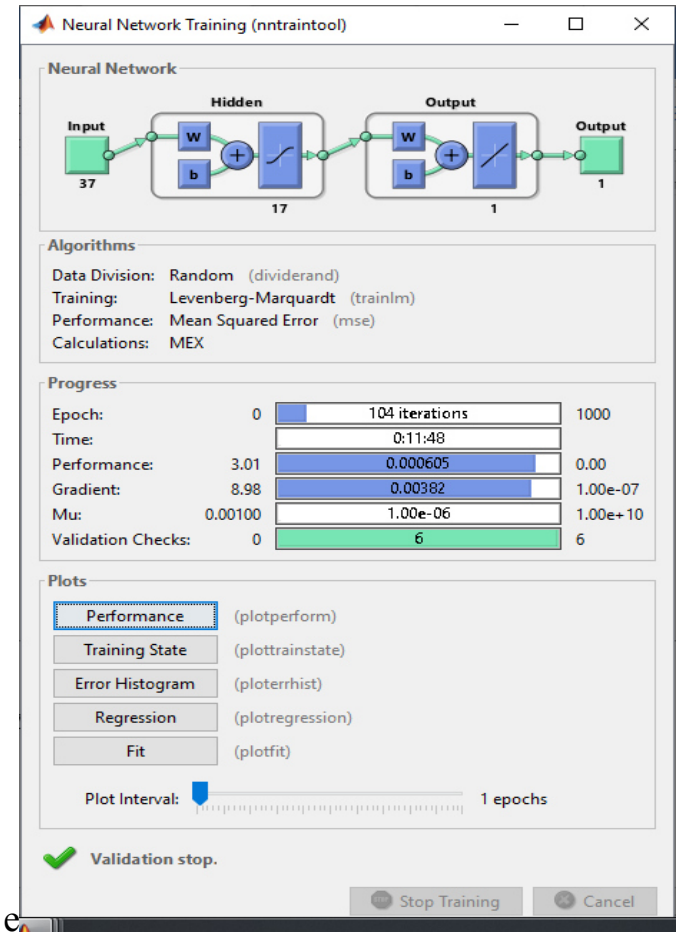

Рисунок 2.17 – Результат обучения нейросети с 17 нейронами

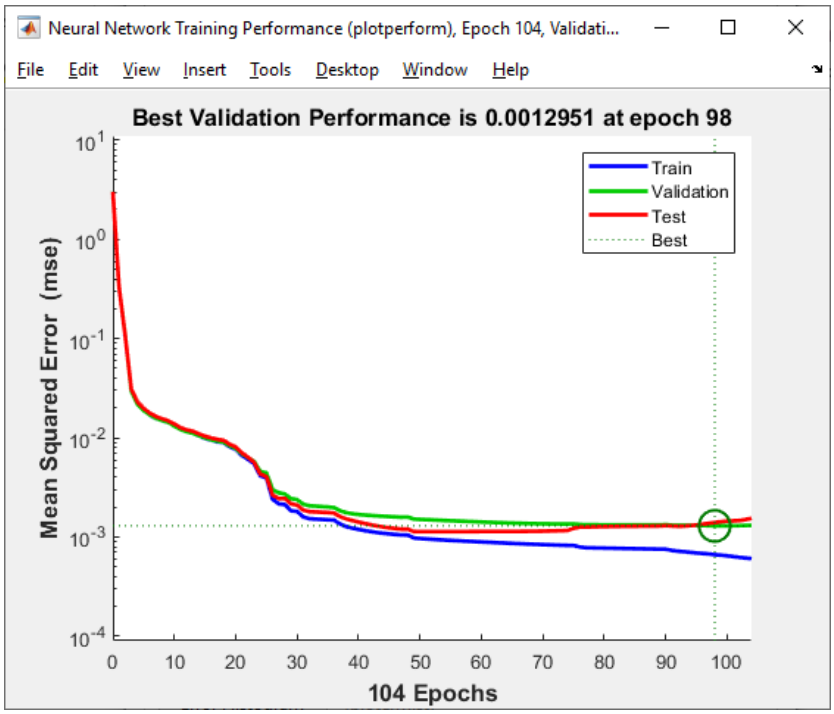

Рисунок 2.18 – График результата обучения нейросети с 17 нейронами

Обучение с 16 нейронами

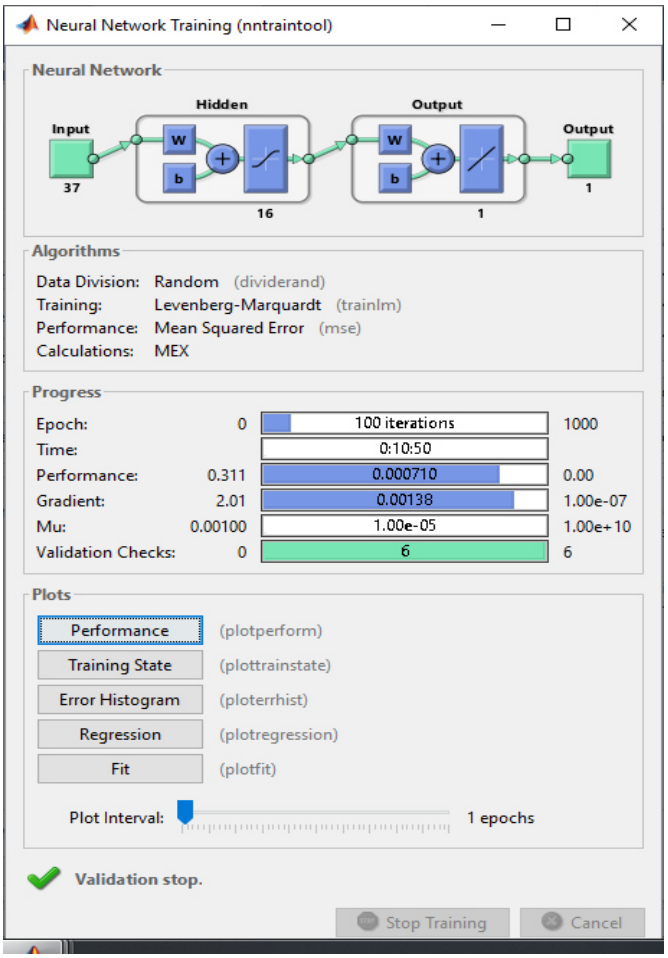

Рисунок 2.19 – Результат обучения нейросети с 16 нейронами

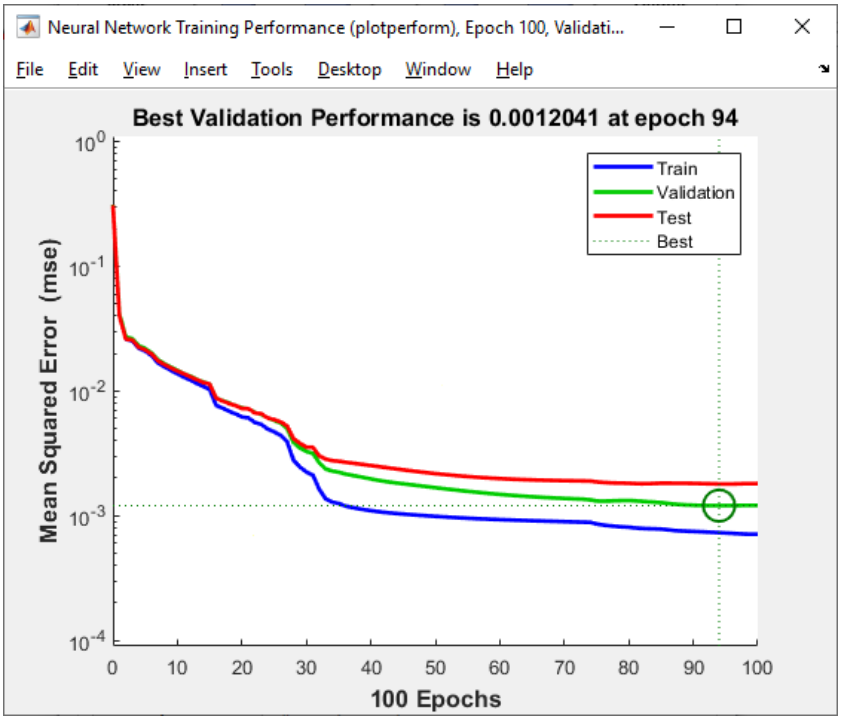

Рисунок 2.20 – График результата обучения нейросети с 16 нейронами

Нейросеть с 15 нейронами

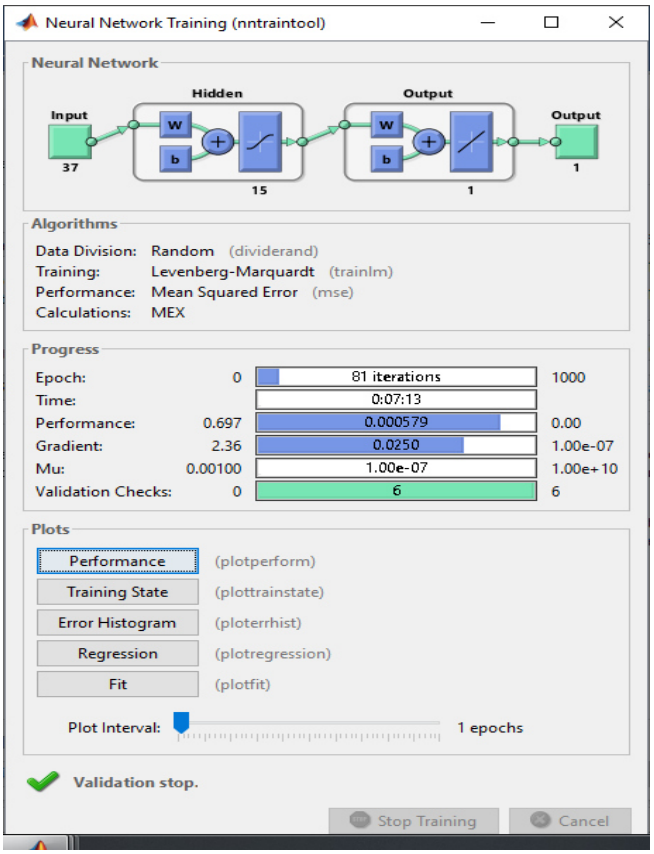

Рисунок 2.21 – Результат обучения нейросети с 15 нейронами

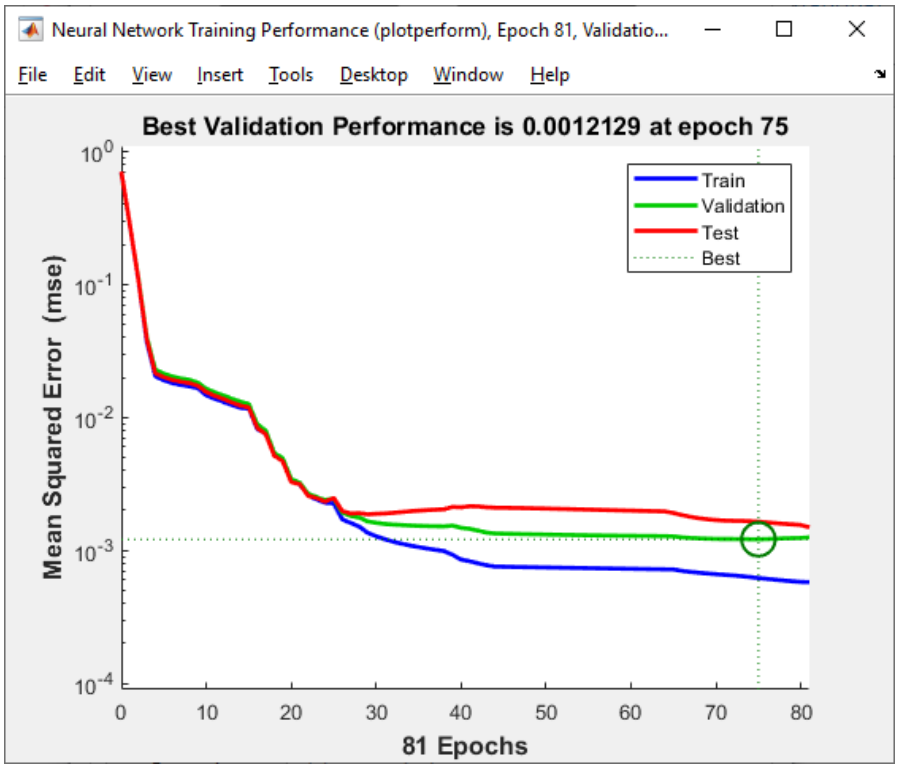

Рисунок 2.22 – График результата обучения нейросети с 15 нейронами

Обучение нейросети с 14 нейронами

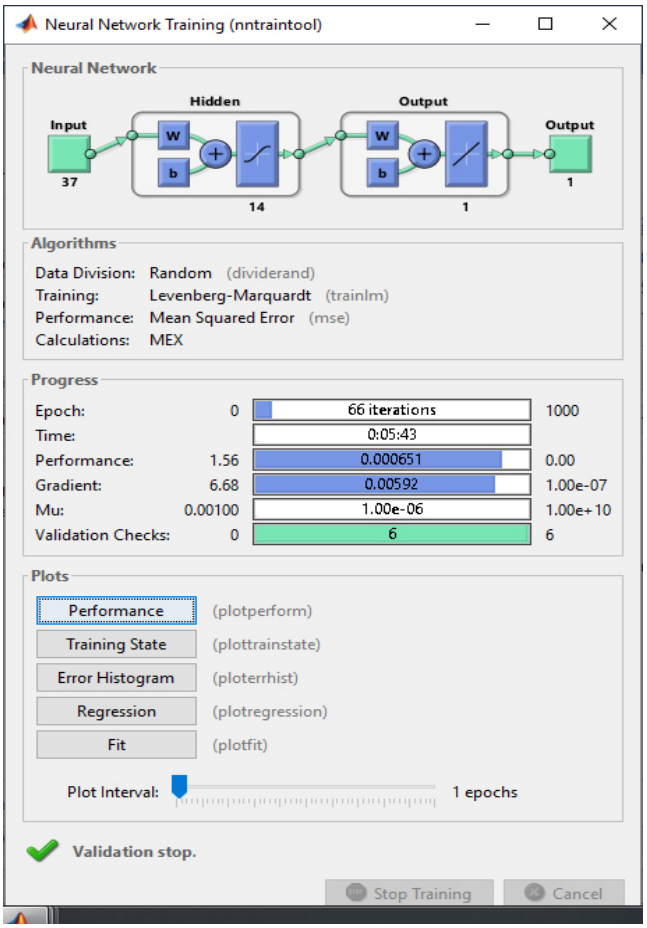

Рисунок 2.23 – Результат обучения нейросети с 14 нейронами

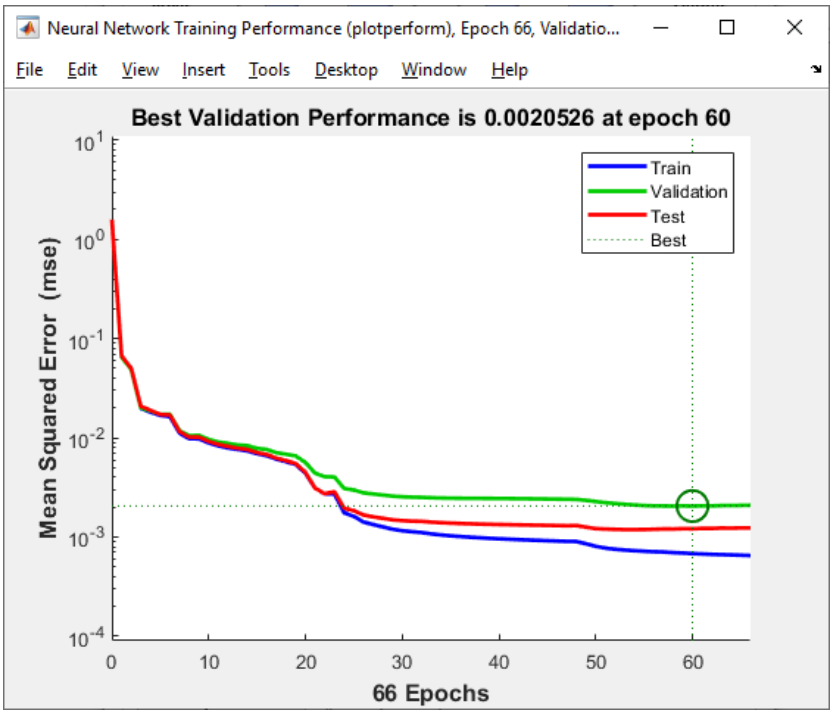

Рисунок 2.24 – График результата обучения нейросети с 14 нейронами

Обучение нейросети с 13 нейронами

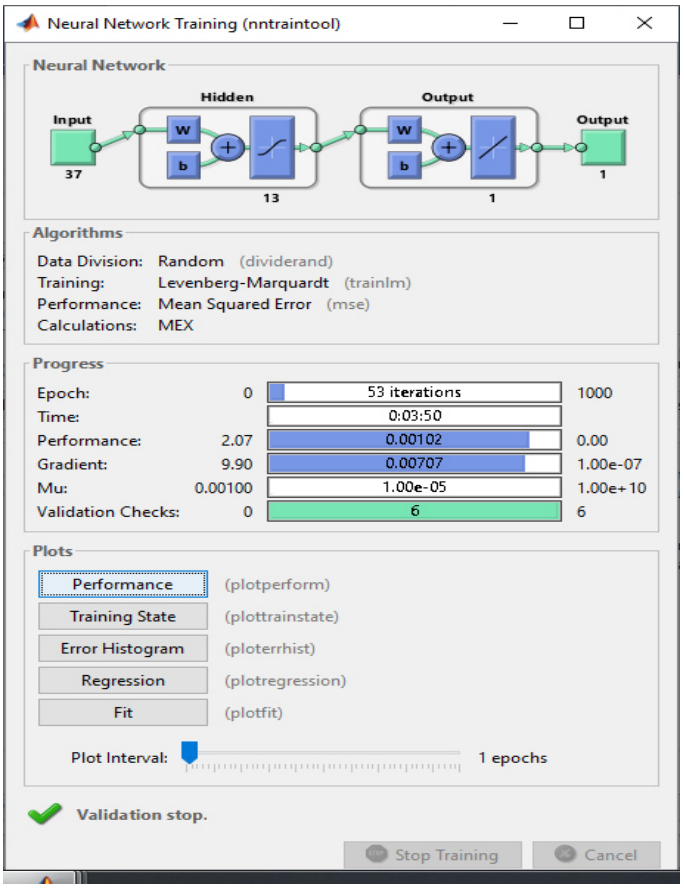

Рисунок 2.25 – Результат обучения нейросети с 13 нейронами

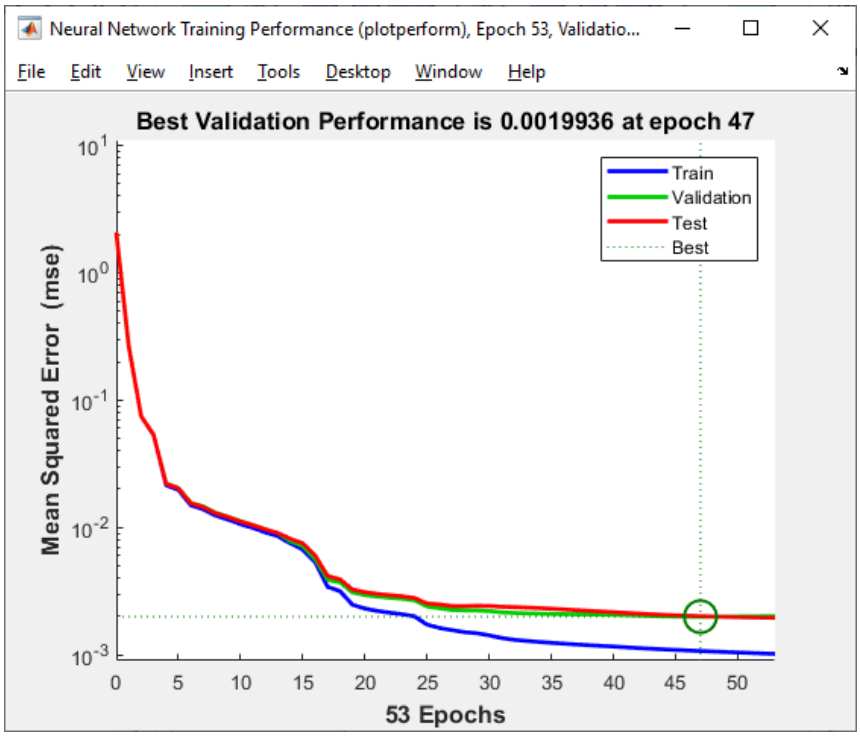

Рисунок 2.26 – График результата обучения нейросети с 13 нейронами

Обучение нейросети с 12 нейронами

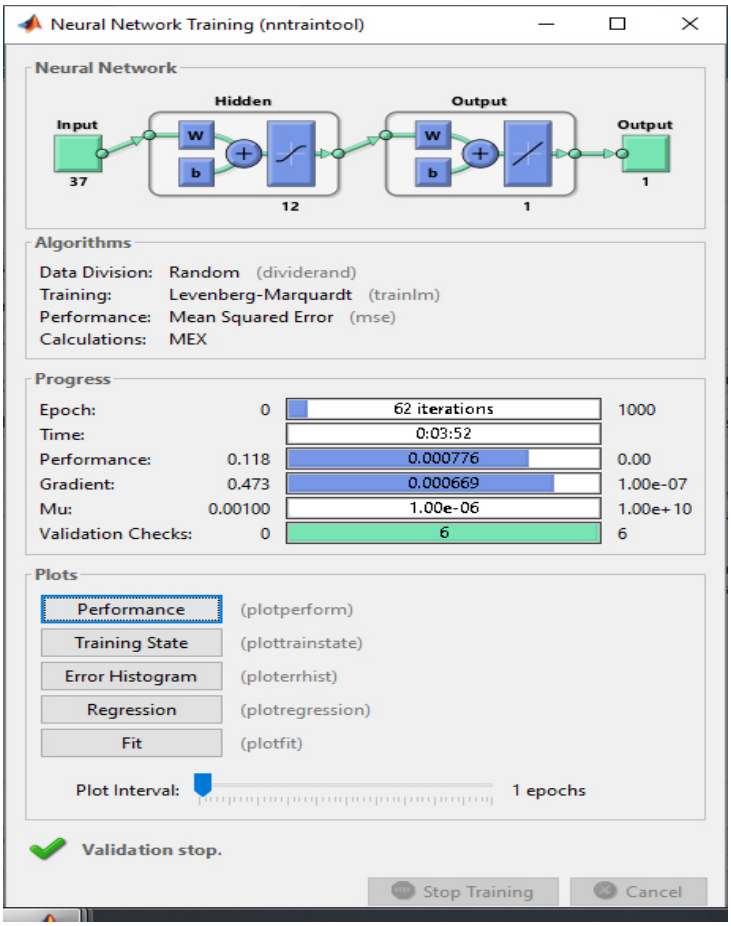

Рисунок 2.27 – Результат обучения нейросети с 12 нейронами

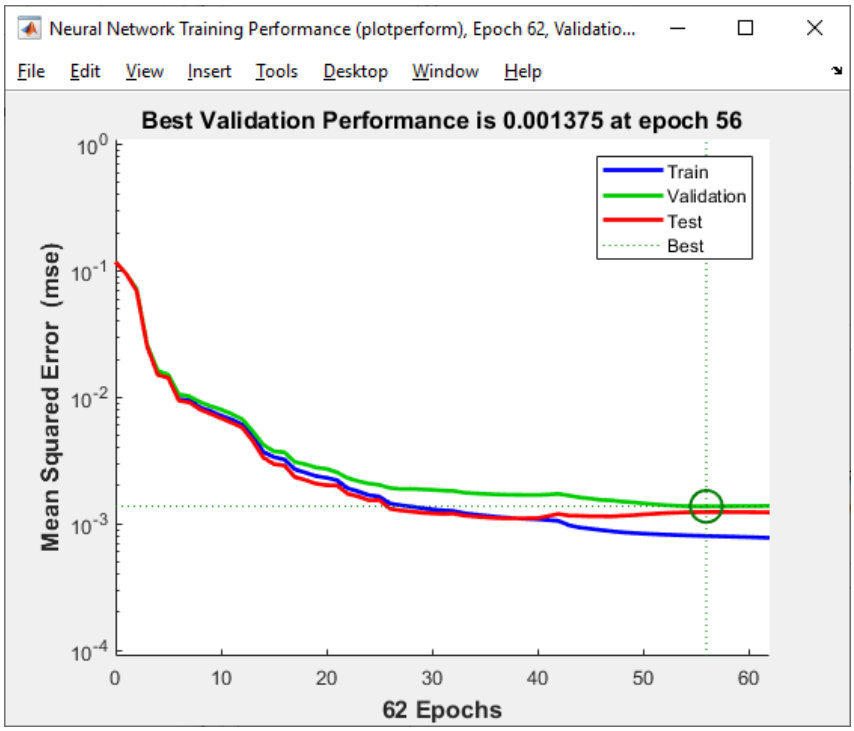

Рисунок 2.28 – График результата обучения нейросети с 12 нейронами

Обучение нейросети с 11 нейронами

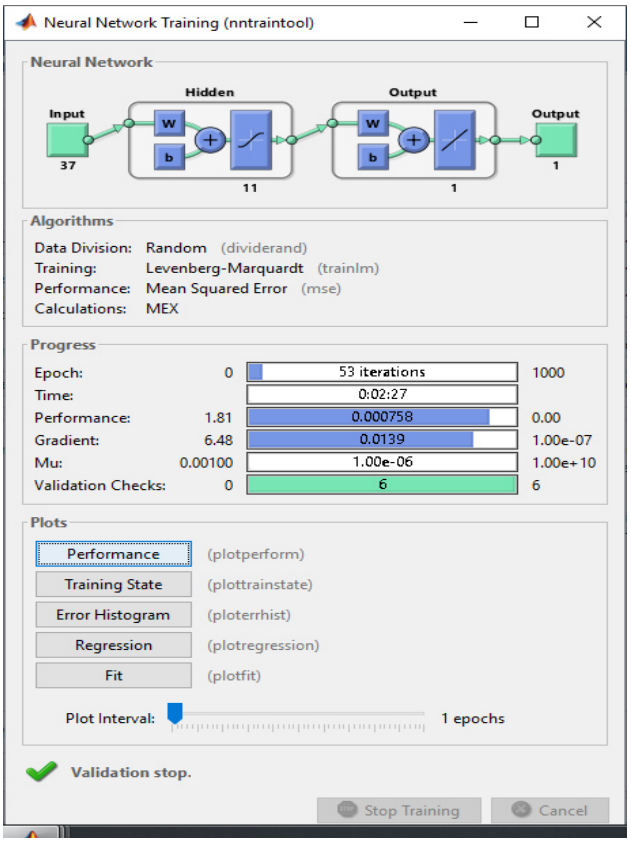

Рисунок 2.29 – Результат обучения нейросети с 11 нейронами

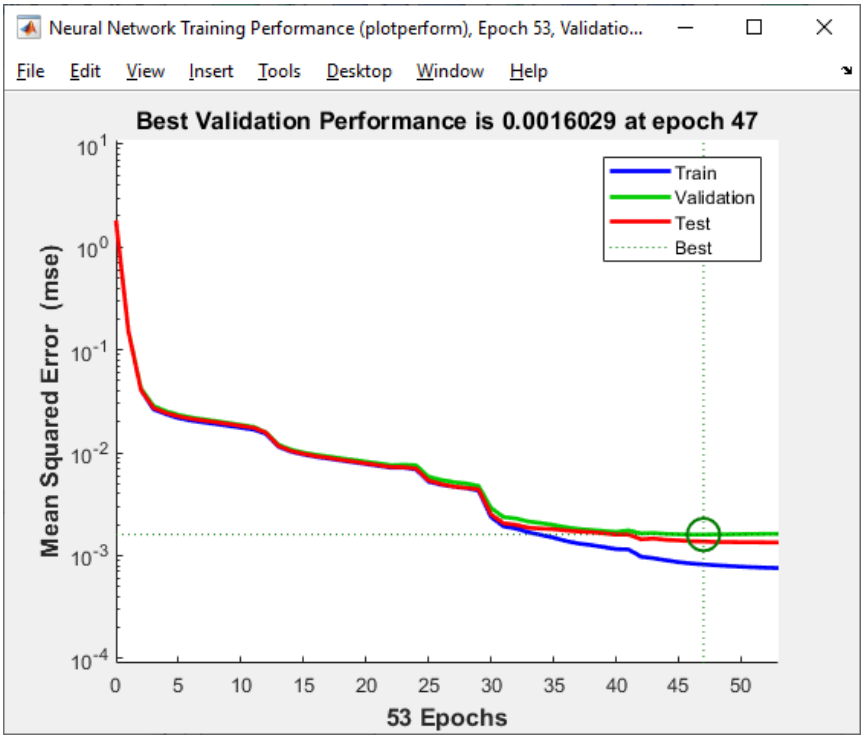

Рисунок 2.30 – График результата обучения нейросети с 11 нейронами

Обучение нейросети с 10 нейронами.

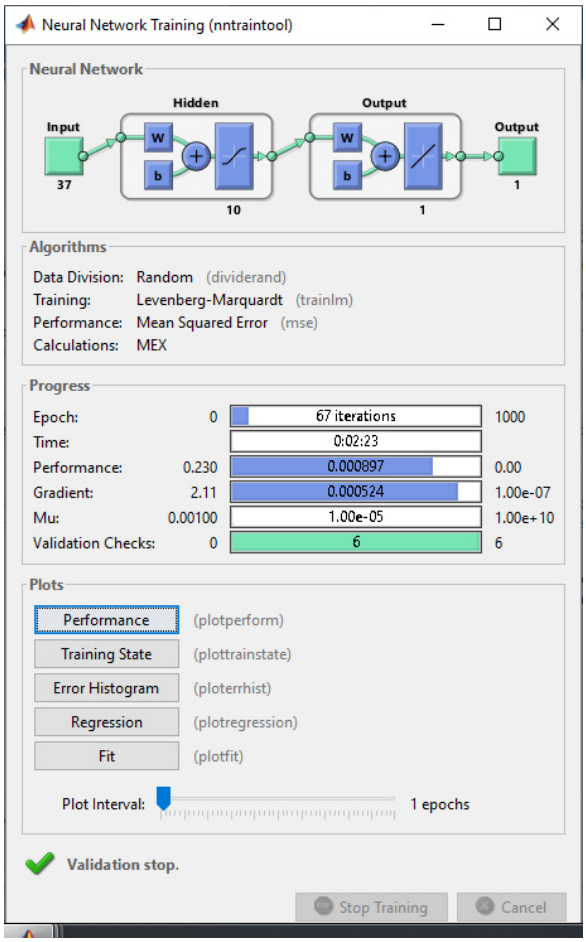

Рисунок 2.31 – Результат обучения нейросети с 10 нейронами

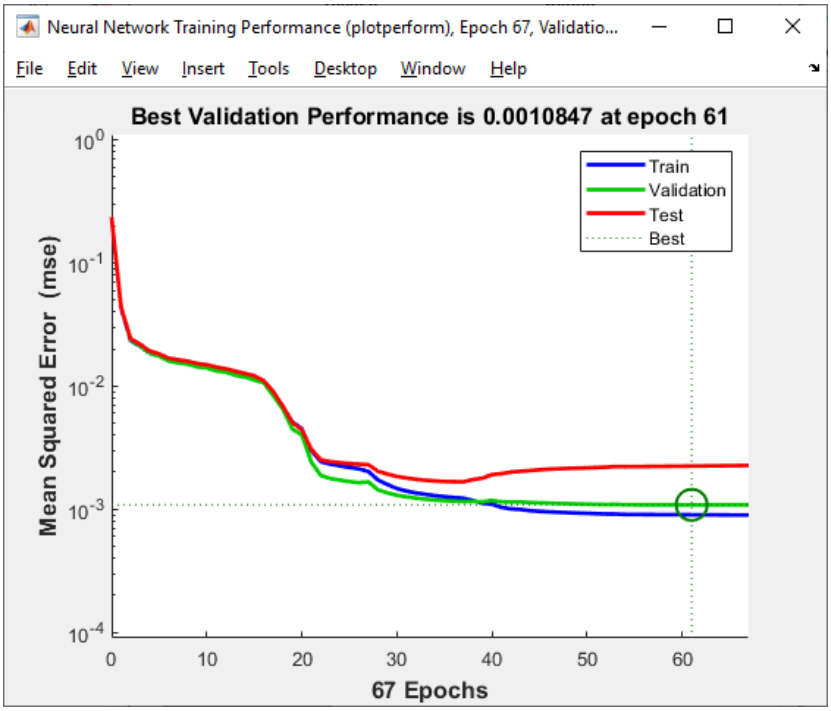

Рисунок 2.32 – График результата обучения нейросети с 10 нейронами

В результате были проведено обучение нейросети 12 раз.

Таблица 2.1 – Сокращение числа нейронов на скрытом слое

Количество нейронов на скрытом слое	Количество итераций (эпох)	Время, затраченное на обучение, мм:сс
21	70	11:50
20	97	15:52
19	78	11:04
18	133	18:11
17	104	11:48
16	100	10:50
15	81	7:13
14	66	5:43
13	53	3:50
12	62	3:52
11	53	2:27
10	67	2:23

Теперь проверяю нейросеть, для проверки ее способности определять атаки. Задаю ей 3 вектора соединений – 2 обычных и 1 атаку

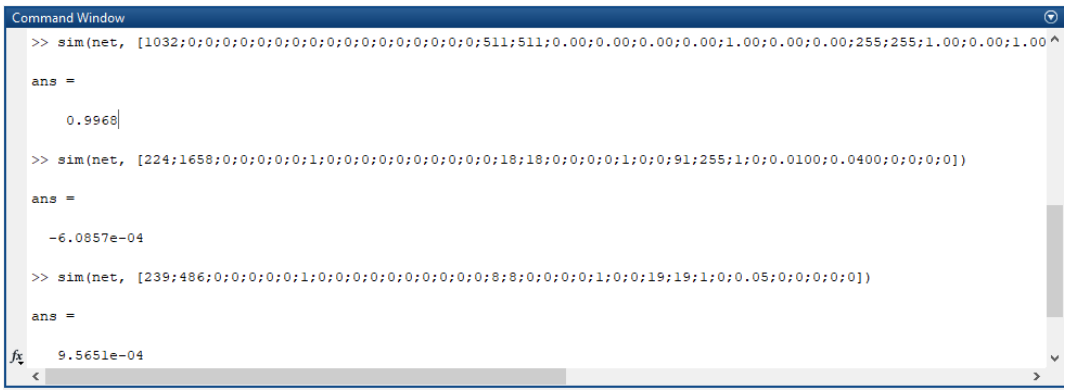

Рисунок 2.33 – Результат выполнения нейросети

Как видно из рисунка, из 3 соединений нейросеть смогла определить, сетевые атаки. Соединение 1 является атакой, так как полученное для него значение 0,9968 близко к 1, а нейросеть была обучена принимать за 0 нормальное соеденине в то время как 2 и 3 – нормальными. Значение 6.0857e-04 для 2 соединения означает число $6{,}0857*10^{-4}$, а значение 9.5651e-04 для 3 соединения означает число $9{,}5651*10^{-4}$, и они оба близки к 0.

3. Технико-экономическое обоснование

Дипломная работа на тему «Применение нейросетевых технологий при обнаружении вторжений». Цель данной дипломной работы заключается в создании нейронной сети в программной среде Matlab, ее обучении – распознавании сетевых атак и использовании в дальнейшем для их обнаружения.

В данном дипломном проекте по разработке нейросетевого метода защиты информации будет участвовать группа специалистов, которая включает в себя: руководитель проекта, программист, инженер по информационной безопасности, тестировщик. В обязанности руководителя проекта входит соблюдение и разработка рабочих графиков, их контроль и оптимизация. В обязанности программиста входит создание и обучение нейронной сети в программной среде Matlab. В обязанности инженера по информационной безопасности входит помощь программисту с обучением нейросети видам сетевых атак. В обязанности тестировщика-пентестера входит тестирование полученной нейросети, а именно имитация сетевой атаки. Технико-экономическое обоснование содержит следующие пункты:

– определение сложности разработки программного продукта;
– расчет затрат на разработку ПП;
– определение ценности готового продукта;
– оценка результатов работы программного продукта.

3.1. Трудоемкость разработки ПП

Создание продукта включает в себя не только обучение, конфигурацию нейросети, но и детальный анализ проекта, и поиск решений для достижения поставленных целей проекта. В таблице 1 показана возможная поэтапная разработка сайта .

Таблица 3.1 – Этапы разработки ПП

Этапы разработки ПО	Вид работы	Трудоемкость, чел. час.
Этап 1	Предпроектные исследования	15
Этап 2	Разработка технического задания	15
Этап 3	Создание и конфигурация нейронной сети	30
Этап 4	Обучение нейронной сети	30
Этап 5	Тестирование	25
Этап 6	Анализ полученных результатов	35
Этап 7	Повторное обучение	25

Продолжение таблицы 3.1

Этап 9	Окончательный анализ	20
Этап 10	Сдача нейросети в эксплуатацию	30
Итого: трудоемкость выполнения программного продукта		250

Продолжительность рабочего дня равна 8 часам. В результате для реализации программного обеспечения необходим 250/8=31 рабочий день.

3.2. Расчет затрат на разработку ПП

Определение затрат на разработку ПП определяется путем составления соответствующей сметы, которая включает в себя следующие статьи:

– материальные затраты;
– затраты на оплату труда;
– социальный налог;
– амортизация основных фондов;
– прочие затраты.

К материальным затратам относятся затраты на материалы, энергию и другие затраты необходимые для разработки ПП. Расчет материальных затрат происходит по форме, предоставленной в таблице 3.2.

Таблица 3.2 – Затраты на материальные ресурсы

Наименование материала	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Бумага (1000 листов)	Упаковка	2	1 100,00	2 200,00
Тонер-картридж для принтера	Штука	2	5 000,00	10 000,00
Магнитно маркерная доска 70х40см	Штука	1	7 200,00	7 200,00
Итого:				19 400,00

Для реализации программного продукта необходимы материалы на сумму 19 400 тенге.

Общую сумму, необходимую на материальные средства ($З_м$) можно рассчитать по следующей формуле:

$$З_м = \sum \boldsymbol{P_i} * Ц_{\boldsymbol{i}}, \quad (3.1)$$

где $\boldsymbol{P_i}$ - расход i-го вида материального ресурса, натуральные единицы;

$Ц_{\boldsymbol{i}}$ - цена за единицу i-го вида материального ресурса, тг;

i - вид материального ресурса;

n - количество видов материальных ресурсов.

Для разработки программного обеспечения будет использоваться ноутбук Acer Aspire 5750G. Также будет использован маршрутизатор для раздачи Wi-Fiи получения доступа в Интернет и принтер. Помимо этого, необходим набор бумаги и тонер-картридж для принтера и магнитно-маркерная доска.

Расчет затрат на необходимое оборудование и программное обеспечение производится по форме, приведенной в таблице 3.3.

Таблица 3.3 – Расчет затрат на оборудование и ПП, необходимое для проекта

Наименование материала	Марка	Ед. измерения	Количество	Цена за ед. в тенге	Сумма в тенге
Ноутбук	AcerAspire 5750G	Штук	1	165 000,00	165 000,00
Принтер	HPDeskjet 2520hc	Штук	1	38 670,00	38 670,00
Маршрутизатор	TP-Link TL-WR741ND	Штук	1	9 600,00	9 600,00
Итого:					213 270,00

$$З_м = 213\ 270{,}00 \text{ тг.}$$

Для реализации программного продукта необходима аппаратура на сумму 213 270,00 тенге.

Общая сумма, необходимая на материальные расходы будет равна:

$$З_М = 19\ 400{,}00 + 213\ 270{,}00 = 232\ 670{,}00 \text{ тг.}$$

3.3. Расчет затрат на электроэнергию

Так как для разработки ППиспользуется электрооборудование, то необходимо рассчитать затраты на электроэнергию.

Расчет электроэнергии, которая необходима для оборудования определяется по следующей формуле:

$$З_{эл.эн.обор.} = \sum \boldsymbol{M} * \boldsymbol{K} * \boldsymbol{S} * \boldsymbol{T}, \qquad (3.2)$$

где М – потребляемая мощность, Вт;

К – коэффициент использования ($К_{исц}$ = 0,7..0,9);

Т – время работы;

S – тариф (1кВт/ч = 23,85 тг).

Итоги по расчетам стоимости затрачиваемой электроэнергии представлены в таблице 3.4.

Таблица 3.4 – Затраты на электроэнергию

Наименование приборов	Паспортная мощность, кВт	Коэффициент мощности	Время работы оборудования, ч	Цена электроэнергии, тг/кВт*ч	Сумма, тг.
Ноутбук	0,54	0,8	160	23,85	1 648,51
Маршрутизатор	0,1	0,9	160	23,85	343,44
Принтер	0,4	0,9	16	23,85	137,38
Освещение	0,3	0,7	160	23,85	801,36
Итого:					2 930,69

$З_{эл.эн.обор.}$ = 2 930,69 (тенге)

3.4. Расчет затрат на оплату труда

Для разработки программного продукта, как указывалось ранее, необходимо четыре работника:

– руководитель;

– программист;

– инженер информационной безопасности;

– тестировщик-пентестер;

Сумму расходов на оплату труда можно рассчитать по следующей формуле:

$$З_{тр} = \sum ЧС_{i} * T_{i} \quad (3.3)$$

где $ЧС_{i}$ - часовая ставка i-го работника, тг;

T_{i} - трудоемкость разработки модели, чел.×ч; i - категория работника;

n - количество работников, занятых разработкой ПП.

Часовую ставку сотрудника можно рассчитать по следующей формуле:

$$ЧС_{i} = \frac{ЗП_{i}}{ФРВ_{i}} \quad (3.4)$$

где $ЗП_{i}$ - месячная заработная плата i-го работника, тг;

$ФРВ_{i}$- месячный фонд рабочего времени i-го работника, час.

Месячная заработная плата руководителя равняется 250 000 тенге, месячная заработная плата программиста Matlab равняется 185 000 тенге, месячная заработная плата инженера информационной безопасности

равняется 215 000 тенге, месячная заработная плата тестировщика-пентестераравняется 129 000 тенге. Рассчитаем часовую ставку каждого работника согласно формуле (3.4):

$$ЧС_{руководитель} = \frac{\mathbf{250\,000}}{\mathbf{22 * 8}} = \mathbf{1\,420,45}тг/ч$$

$$ЧС_{\boldsymbol{Matlab}-прогр.} = \frac{\mathbf{185\,000}}{\mathbf{22 * 8}} = \mathbf{1\,051,14}\ тг/ч$$

$$ЧС_{инженер\ инф.безоп.} = \frac{\mathbf{215\,000}}{\mathbf{22 * 8}} = \mathbf{1\,221,59}тг/ч$$

$$ЧС_{тестировщик-пентестер} = \frac{\mathbf{129\,000}}{\mathbf{22 * 8}} = \mathbf{732,95}тг/ч$$

Расчеты затрат по оплате труда показаны в таблице 5.

Таблица 3.5 – Расчет заработной платы

Категория работника	Квалификация	Трудоемкость разработкиПП, час.	Часовая ставка, тг/ч	Сумма, тг.
Руководитель	Инженер-проектировщик	120	1 420,45	170 454,00
Программист	Инженер-программист	90	1 051,14	94 602,60
Инженер информационной безопасности	Специалист по информационной безопасности	90	1 221,59	109 943,10
Тестировщик-пентестер	Специалист по информационной безопасности	100	732,95	73 295,00
Итого:				448 294,10

3.5. Расчет затрат по социальному налогу

Согласно Налоговому кодексу Республики Казахстан социальный налог составляет 9,5% от фонта оплаты труда, поскольку в данном случае нет необходимости добавлять 1,5% на медицинскую страховку. Социальный налог можно рассчитать по следующей формуле:

$$С_н = (ФОТ - ПО) * \mathbf{0,095} \quad (3.5)$$

где ПО - отчисления в пенсионный фонд, они составляют 10% от ФОТ.

$$ПО = \mathbf{448\,294,10 * 0,1 = 44\,829,41}\ тенге$$

$$С_н = (\mathbf{448\,294,10 - 44\,829,41}) * \mathbf{0,095 = 38\,329,15}\text{ тенге}$$

3.6. Амортизация основных фондов и прочиезатраты

Нормы амортизации ОФ необходимо определить в соответствии с налоговым кодексом РК. Амортизацию ОФ можно определить по следующей формуле:

$$\boldsymbol{A}_г = \frac{С_{об} * Н_а}{\mathbf{100}} \quad (3.6)$$

где, $С_{об}$ – стоимость оборудования;

$Н_а$ – норма амортизации (норма амортизация = 25);

Формула (6) позволяет рассчитать нужную сумму для амортизационных отчислений за год для ноутбука:

$$\boldsymbol{A}_г = \frac{\mathbf{165\,000 * 25}}{\mathbf{100}} = \mathbf{41\,250}\text{ тенге}$$

Теперь необходимо рассчитать норму амортизации за период разработки:

$$\boldsymbol{A}_г = \frac{\mathbf{41\,250 * 31}}{\mathbf{365}} = \mathbf{3\,503,42}\text{ тенге}$$

Где 31 – это общее количество дней разработки. Подобным образом необходимо рассчитать норму амортизации для всего оборудования. Результаты расчетов приведены в таблице 6.

Таблица 3.6 – Амортизация ОФ

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Сумма амортизации за год, тг	Сумма амортизации за время разработки, тг
Ноутбук	165 000,00	25	41 250,00	3 503,42
Принтер	38 670,00	25	9 667,50	821,07
Маршрутизатор	9 600,00	25	2 400,00	203,84
Итого:			53 317,50	4 528,33

3.7. Смета расходов на разработку ПП.

На основе всех представленных расчетов необходимо оформить смету расходов на разработку ПП согласно форме, которая приведена в таблице 7.

Учтем прочие расходы, к которым относится оплата за интернет (1 790,00 тг в месяц).

$$З_{пр} = 1\,790\text{ тг.}$$

Таблица 4.7 – Смета затрат на разработку ПП

Статьи затрат	Сумма, тг	Процентное соотношение, %
Затраты на материальные расходы	232 670,00	32
Затраты на оплату труда	448 294,10	61
Социальные налоги	38 329,15	5
Затраты на электроэнергию	2 930,69	0,65
Амортизация основных фондов	4 528,33	0,8
Прочие расходы	1 790,00	0,55
Итого по смете:	728 542,27	100

Рисунок 4.1 – Диаграмма затрат

3.8. Определение возможной (договорной) цены ПО

Стоимость программного продукта определяется на основе качества разработанного продукта, сроков его разработки и производительности продукта. Стоимость $Ц_Д$ можно рассчитать по следующей формуле:

$$Ц_Д = З_{нир}\left(\mathbf{1} + \frac{\boldsymbol{P}}{\mathbf{100}}\right), \quad (3.7)$$

где $З_{нир}$ – затраты на разработку программного продукта,тг;

Р – средний уровень рентабельности ПО, (%). Данный параметр принят равным 25%.

$$Ц_д = \mathbf{728\,542,27 + 728\,542,27 * 0,25 = 728\,542,27 + 182\,135,57} = \mathbf{910\,677,84}\text{ тенге}$$

Далее необходимо определить стоимость реализации с учетом НДС, ставка НДС устанавливается законодательством РК. На 2019 года ставка НДС составляет 12%. Стоимость реализации, учитывая НДС можно рассчитать по следующей формуле:

$$Ц_р = Ц_д + Ц_д * НДС, \quad (3.8)$$

$$Ц_р = \mathbf{910\,677,84 + 910\,677,84 * 0,12 = 910\,677,84 + 109\,281,34} = \mathbf{1\,019\,959,18}\text{ тг.}$$

В данной части дипломного проекта содержится экономические расчеты, которые показываютзатраты необходимые при разработке нейросетевого инструмента защиты информации. Расчеты включают в себя:

– определение трудоемкости разработки программного продукта;

– расчет затрат на разработку ПП;

– определение ценности готового продукта;

– оценка результатов работы программного продукта.

По выполненным расчетам можно сказать, что себестоимость программного продукта равна 728 542,27 тг., расчеты так же показали, что возможная прибыль равна 182 135,57 тг., а цена реализации с учетом НДС равна 1 019 959,18 тг.

4. Глава БЖД

4.1 Анализ условий труда

Дипломный проект по теме «Применение нейро-сетевых технологий при обнаружении вторжений», что подразумевает собой программное средство, для эксплуатации которого необходим рабочий компьютер. Имеется помещение размерами 21х10 м и окном площадью S=60 м^2. В помещении размещены компьютеры, доступ к которым имеют сотрудники информационной безопасности, начальство и охрана. Также в помещении имеются принтеры и роутер. Все устройства являются новыми и производят низкий уровень шума, кроме этого в самом помещении есть шумоподавление. Данный раздел посвящен расчету естественного и искусственного освещения.

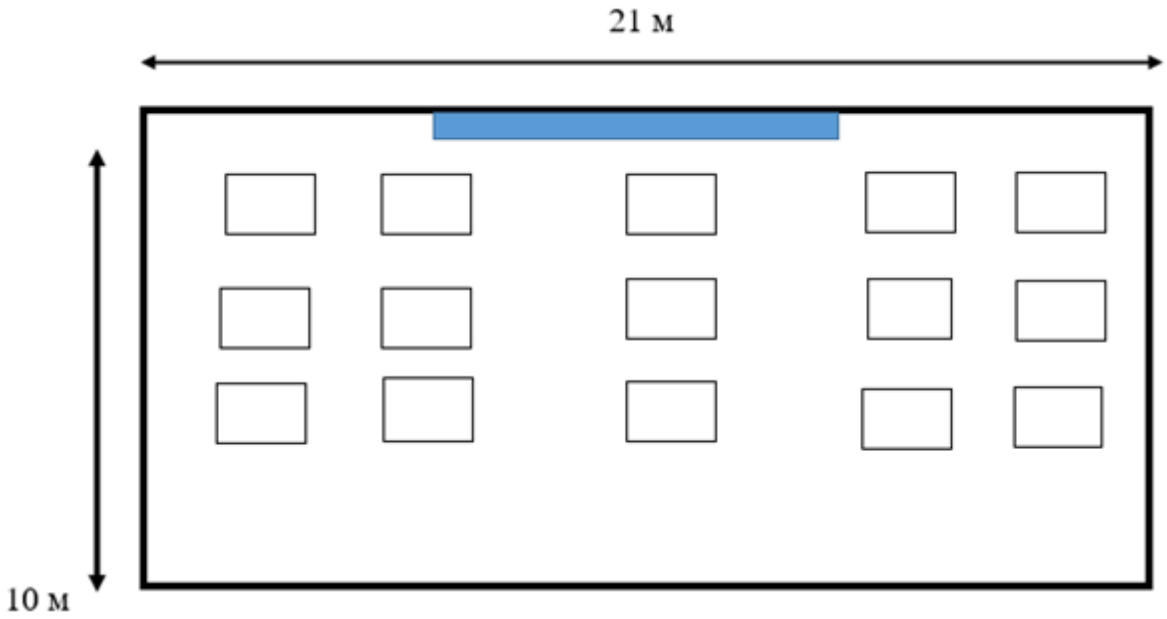

Рисунок 4.1 – Размерность помещения

Таблица 4.1 – Исходные данные

Тип помещения	Параметры помещения				Разряд зрит. работ	$\rho_{пот}$	$\rho_{стен}$	$\rho_{пол}$	$h_{нок}$ м	Световой пояс	Н зд	Расст. до рядом стоящего здания, Р
	L, м	B, м	H, м	$h_{ок}$, м								
Офисное	21	15	6	4	III,б	50	50	10	1	Алматы	18	10

4.1. Расчет естественного освещения

Изначально в помещении были рассмотрены 15 ламп и окно 60 м^2

Площадь боковых проемов при боковом освещении определяется из следующей формулы:

$$100 \cdot \frac{S_0}{S_n} = \frac{e_N \cdot К_з \cdot \eta_0}{\tau_0 \cdot r_1} \cdot К_{зд}, \quad (4.1)$$

где S_0- площадь световых проемов при боковом освещении, м^2;
S_n– площадь пола помещения, м^2;
e_N – нормируемое значение КЕО;
$К_З$ –коэффициент запаса;
η_0– световая характеристика окон;
τ_0– общий коэффициент светопропускания;
r_1– коэффициент, учитывающий повышение КЕО при боковом освещении, благодаря свету, отраженному от поверхности помещения и подстилающего слоя, примыкающего к заданию;
$К_{зд}$– коэффициент, учитывающий затемнение окон противостоящими зданиями.
Определим площадь пола помещения:

$$S_n = L \cdot B \quad (4.2)$$

$$S_n = 21 \cdot 15 = 315 \text{ м}^2$$

Нормируемое значение КЕО, e_N, для заданий, располагаемых в различных районах определять по формуле:

$$e_N = e_H \cdot m_N, \quad (4.3)$$

где m_N – коэффициент светового климата;
e_H – значение КЕО.
Учитывая заданный световой пояс (г.Алматы) адм. район 9, приняв ориентацию световых проемов З, В определим:

$$m_N = 0.8$$

Учитывая III б разряд зрительных работ, найдем:

$$e_H = 1.2$$

Следовательно:

$$e_N = 1.2 \cdot 0.8 = 0{,}96$$

Учитывая тип помещения, найдем коэффициент запаса:

$$К_з = 1.2$$

при ЕО вертикально.

Для определения световой характеристики, $\boldsymbol{\eta_0}$, необходимо рассчитать отношение длины помещения к его глубине $\frac{\mathbf{L}}{\mathbf{l}}$, отношение ширины помещения к расчетной высоте $\frac{\mathbf{l}}{\mathbf{h_{расч}}}$.

$$l = B - 1 \tag{4.4}$$

$$l = 15 - 1 = 14 \text{ м}$$

$$\frac{L}{l} = \frac{21}{14} = 1.5$$

Найдем $h_{расч}$:

$$h_{расч} = h_{0к} + h_{н.ок.} - h_{р.п.} \tag{4.5}$$

$$h_{расч} = 4 + 1 - 0.8 = 4.2 \text{ м}$$

$$\frac{l}{h_{pac}} = \frac{14}{4.2} = 3.3 \approx 3$$

$$\frac{l}{B} = \frac{14}{15} = 0.93 \approx 1$$

Учитывая найденные отношения примем световую характеристику, $\eta_0 = 15$.

Общий коэффициент светопропускания, τ_0, рассчитывают по формуле:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4, \tag{4.6}$$

где τ_1 – коэффициент светопропускания материала.

Так как в качестве светопропускающего материала используется стекло листовое двойное, то:

$$\tau_1 = 0.8$$

τ_2 – коэффициент, учитывающий потери света в переплетах светопроема. Определяется с учетом использования стальных двойных глухих переплетов:

$$\tau_2 = 0.8$$

τ_3 – коэффициент, учитывающий потери света несущих конструкциях, при боковом освещении:

$$\tau_3 = 1$$

τ_4 – коэффициент, учитывающий потери света в солнцезащитных устройствах. Выбираем регулируемые жалюзи и шторы (межстекольные внутренние, наружные)

$$\tau_4 = 1$$

Следовательно:

$$\tau_0 = \tau_1 \cdot \tau_2 \cdot \tau_3 \cdot \tau_4 = 0.8 \cdot 0.8 \cdot 1 \cdot 1 = 0.64$$

$$\rho_{ср} = \frac{\rho_{пот} + \rho_{стен} + \rho_{пол}}{3} \quad (4.7)$$

$$\rho_{ср} = \frac{50 + 50 + 10}{3} = 0{,}36 \approx 0{,}3$$

$$r_1 = 1{,}7$$

Учитывая $Н_{зд}$=18 и Р=10 м (расстояние до рядом стоящего здания), найдем коэффициент, учитывающий затемнение окон противостоящими зданиями, $К_{зд}$:

$$\frac{Р}{Н_{зд}} = \frac{10}{18} = 0.55 => К_{зд} = 1.7$$

Зная значение всех параметров, рассчитываем площадь боковых проемов при естественном освещении по следующей формуле:

$$S_0 = \frac{S_п \cdot e_N \cdot К_з \cdot \eta_0}{100 \cdot \tau_0 \cdot r_1} \cdot К_{зд} \quad (4.8)$$

$$S_0 = \frac{315 \cdot 0{,}96 \cdot 1.2 \cdot 15 \cdot 1{,}7}{100 \cdot 0.64 \cdot 1{,}7} = 85{,}05 \text{ м}^2$$

Таким образом, данные расчеты естественного освещения не удовлетворяет рассчитанному нормативному значения.

4.2 Расчет искусственного освещения

Для расчета искусственного освещения используют один из трех методов: по коэффициенту использования светового потока, точечный и метод удельной мощности.

При расчете общего равномерного освещения основным является метод использования светового потока, создаваемого источником света, и с учетом отражения от стен, потолка, пола.

Тип помещения	Параметры помещения				Разряд зрит. работ	$\rho_{пот}$	$\rho_{стен}$	$\rho_{пол}$	W, Вт	Ф, лм
	L,м	B, м	H, м	$h_{ок}$, м						
Офисное	21	15	6	4	III,б	50	50	10	36	5000

Расчет освещения начинают с выбора типа светильника, который принимается в зависимости от условий среды и класса помещений по взрыво- пожароопасности.

Таблица 4.3 – Исходные данные с учетом мощности и светового потока

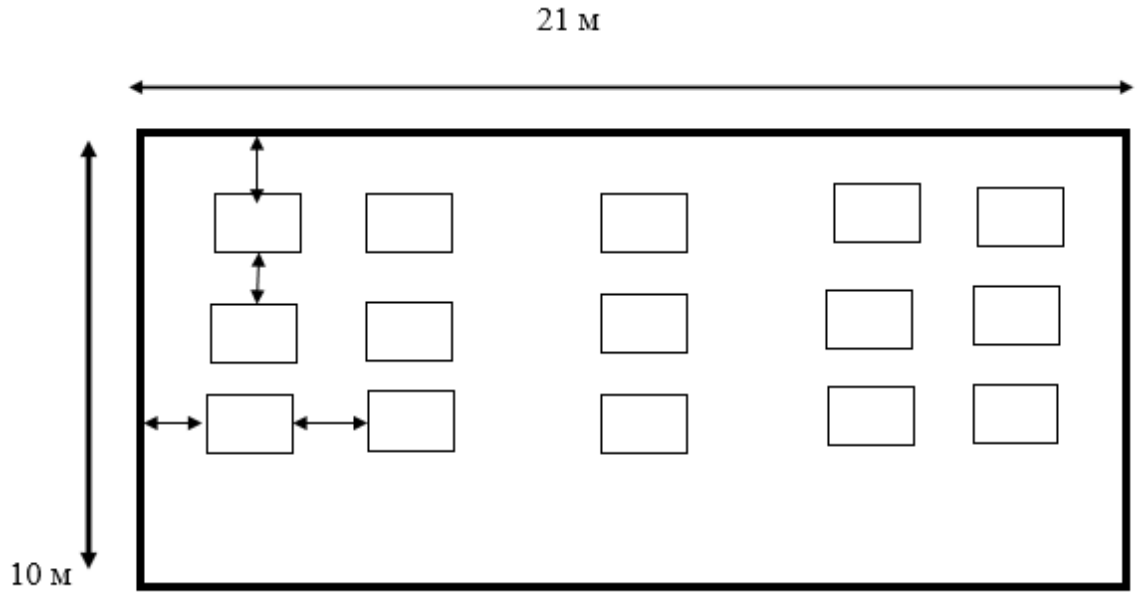

Рисунок 4.2 – первоначальное расположение светильников

Метод коэффициента использования светового потока.

В первую очередь нужно рассчитать заданное номинальное значение оно должно быть больше 300

$$E_\tau = \frac{N n ф \cdot \mu}{K \cdot SV} = \frac{15*2*5000*0{,}55}{1.5*315*1.1} = 158 \quad (4.9)$$

Сначала нужно рассчитать заданное номинальное значение оно должно быть больше 300

Получилась у нас 158<300, что не удовлетворяет условному значению

Разряд зрительной работы III, б, поэтому нормируемая освещенность по таблице Ен= 300 лк (при системе общего освещения).

Определение расчетной высоты подвеса:

$$h_{\text{расч}} = H_{\text{помещения}} - \text{H}_{\text{свеса}} - \text{H}_{\text{р.п.}}, \qquad (4.10)$$

где $\text{H}_{\text{свеса}} = 0{,}5$ – высота свеса ламп, м;
$\text{H}_{\text{р.п.}} = 0{,}8$ – расстояние рабочей поверхности над полом, м;
$H_{\text{помещения}} = 6$ – высота помещения, м.

$$\text{h}_{\text{расч}}=6\text{-}0{,}5\text{-}0{,}8=4{,}7 \text{ м};$$

В практике расчетов значения коэффициентов η находятся из таблиц, связывающих геометрические параметры помещения (индекс помещения) с их оптическими характеристиками.

Индекс помещения определяется по формуле:

$$i = \frac{\text{A} * \text{B}}{h_{\text{расч}} * (\text{A} + B)} = \frac{21 * 15}{4{,}7(21 + 15)} = \frac{315}{169{,}2} = 1{,}86 \qquad (4.11)$$

где А- длина помещения, м;
В- ширина помещения, м;
$h_{\text{расч}}$ - расчетная высота, м.
Для светильника типа TLPL228.2x36 находим η = 0,55.
Таким образом количество светильников равно:

$$N = \frac{\text{E}_{\tau} \cdot \text{K}_{\text{З}} \cdot \text{S} \cdot \text{V}}{n\text{ф} \cdot \mu} = \frac{300 \cdot 1.5 \cdot 315 \cdot 1.1}{2 * 5000 * 0{,}55} = 28 \qquad (4.12)$$

где $\text{E}_{\text{н}} = 300$ лк - заданное номинальное освещение;
$\text{S} = 315\text{м}^2$ – площадь помещения;
Z = 1,1- коэффициент неравномерности освещения;
n- количество ламп в светильнике;
Ф = 5000 лм.

Расчет освещенности точечным методом

Выше мы определили расчетную высоту подвеса $\text{h}_{\text{расч}} = 4{,}7$ м;
Найдем расстояние между светильниками, учитывая λ=0.6÷1.5.

$$L_A=\lambda$$

$$h_p=H_п-h_{свеса}-h_{р.пов.}$$

Принимаем $h_{свеса}$ = 0,5 м $h_{р.пов.}$= 0,8

$$\cdot h_p=1{,}216\cdot 4{,}7=4{,}5 \text{ м}$$

$$L_B= L_A - (0{,}6\div 1{,}5)=4{,}5 - 1{,}5=3 \text{ м}$$

$$l_a=l_b=(L_a/1{,}5)=3/1{,}5=2 \text{ м}$$

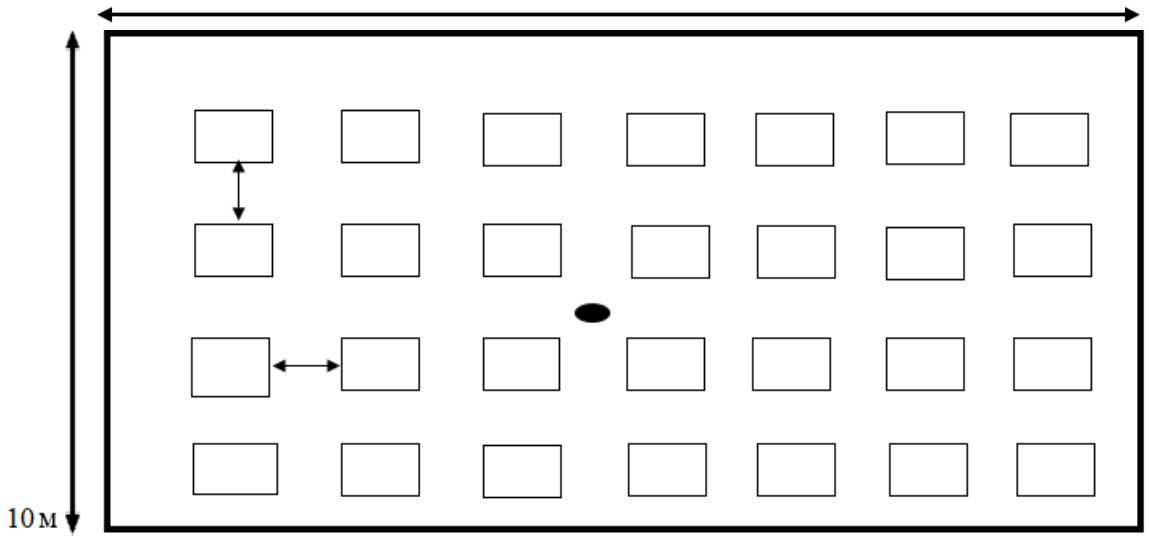

Рисунок 4.3 – использование 28 лапм

Для расчета намечаем контрольную точку А. Необходимо найти $d_{3,18}$; $d_{1,5,16,20}$; $d_{8,13}$; $d_{2,4,17,19}$; $d_{6,10,11,15}$; $d_{7,9,12,14}$; – проекции расстояния на потолок между точкой А и соответствующим светильником:

$$d_{1,5,16,20}=\sqrt{6^2+9^2}=10{,}82 \text{ м};$$

$$d_{2,4,17,19}=\sqrt{4{,}5^2+6^2}=7{,}5 \text{ м};$$

$$d_{6,10,11,15}=\sqrt{2+9^2}=9{,}22 \text{ м}$$

$$d_{7,9,12,14}=\sqrt{4{,}5^2+2^2}= 4{,}9 \text{м}$$

$$d_{3,18}=6 \text{ м};$$

$$d_{8,13}=2 \text{ м};$$

Далее определяем угол между высотой потолка и соответствующим отрезком d:

$$\text{tg}\alpha_1 = \frac{d_{1,5,16,20}}{h\text{расч}} = \frac{10,82}{4,7} = 2,9 \rightarrow \alpha_1 = 71°;$$

$$\cos^3 \alpha_1 = 0,0345$$

$$\text{tg}\alpha_2 = \frac{d_{2,4,17,19}}{h_{\text{расч}}} = \frac{7,5}{4,7} = 2 \rightarrow \alpha_2 = 64°;$$

$$\cos^3 \alpha_2 = 0,084$$

$$\text{tg}\alpha_3 = \frac{d_{6,10,11,15}}{h\text{расч}} = \frac{9,22}{4,7} = 2,5 \rightarrow \alpha_1 = 68°;$$

$$\cos^3 \alpha_3 = 0,053$$

$$\text{tg}\alpha_4 = \frac{d_{7,9,12,14}}{h\text{расч}} = \frac{4,9}{4,7} = 1,3 \rightarrow \alpha_1 = 53°;$$

$$\cos^3 \alpha_4 = 0,218$$

$$\text{tg}\alpha_5 = \frac{d_{3,18}}{h\text{расч}} = \frac{6}{4,7} = 1,62 \rightarrow \alpha_1 = 58°;$$

$$\cos^3 \alpha_5 = 0,149$$

$$\text{tg}\alpha_6 = \frac{d_{8,13}}{h\text{расч}} = \frac{2}{4,7} = 0,54 \rightarrow \alpha_1 = 29°;$$

$$\cos^3 \alpha_6 = 0,669$$

Выбираем тип светильника ПВЛМ (с 2 лампами ЛБР)

Таблица 4.4 – Светотехнические характеристики светильника

Тип светиль-ника	Освещеность I_α, кд при угле α									
	0	15	25	35	45	55	65	75	85	90
ПВЛМ	175	165	148	130	110	70	60	30	20	1

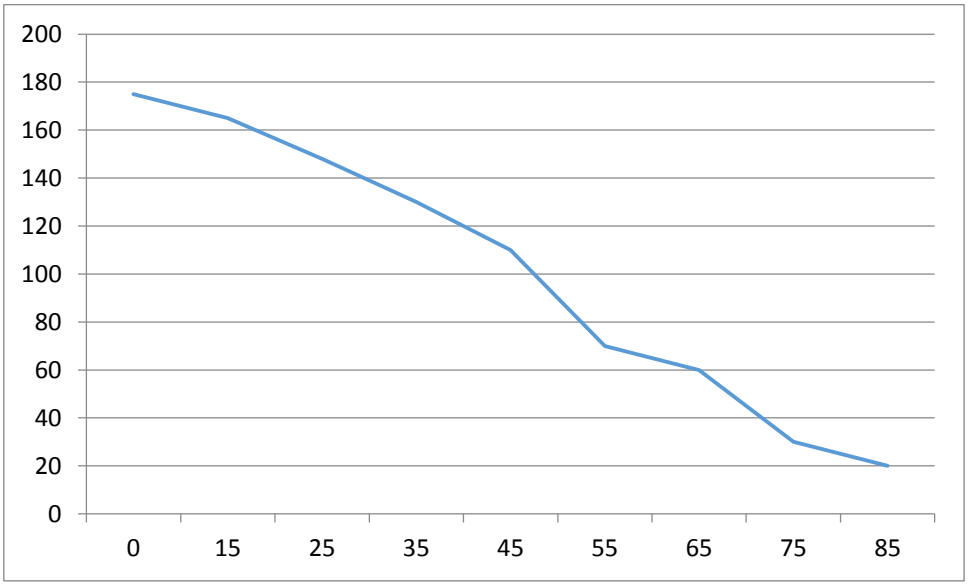

Рисунок 4.4 – Зависимость α=f(I_α)

По этому углу находим силу света от каждого источника:

$I_{\alpha1(1,5,16,20)}$=45 кд

$I_{\alpha2(2,4,17,19)}$=61 кд

$I_{\alpha3(6,10,11,15)}$=55 кд

$I_{\alpha4(7,9,12,14)}$=77 кд

$I_{\alpha5(3,18)}$=70 кд

$I_{\alpha6(8,13)}$=140 кд

Освещенность помещения относительно контрольной точки от каждого источника:

$$e_{АГ} = \frac{n \cdot I_\alpha \cos^3 \alpha}{h_p}$$

$$e_{АГ1} = \frac{4 \cdot 45 \cdot 0{,}0345}{4{,}7^2} = 0{,}45\text{лк};$$

$$e_{АГ2} = \frac{4 \cdot 61 \cdot 0{,}084}{4{,}7^2} = 1{,}5\text{лк};$$

$$e_{АГ3} = \frac{4 \cdot 55 \cdot 0{,}053}{4{,}7^2} = 0{,}85\text{лк};$$

$$e_{АГ4} = \frac{4 \cdot 77 \cdot 0{,}218}{4{,}7^2} = 5\text{лл};$$

$$e_{АГ5} = \frac{2 \cdot 70 \cdot 0{,}149}{4{,}7^2} = 1{,}6\text{лк};$$

$$e_{АГ6} = \frac{2 \cdot 140 \cdot 0.669}{4{,}7^2} = 13{,}9\text{лк};$$

$$\sum_{i=1}^{15} e_{АГi} = e_{АГ1} + e_{АГ2} + e_{АГ3} + e_{АГ4} + e_{АГ5} + e_{АГ6} = 0{,}45+1{,}5+0{,}85+5+1{,}6+13{,}9=23{,}3\text{лк}$$

Суммарная освещенность:

$$E = \frac{\mu \cdot \Phi_л \cdot n}{1000 \cdot K_з} \cdot \sum_{i=1}^{15} e_{АГi}$$

где μ – коэффициент, учитывающий действие «удаленных» светильников (1,1 ÷ 1,25). Световой поток выбранной лампы ЛБР(ЛХБР80) 4160 лм.

$$E_{АГ} = \frac{1.2 \cdot 4160 \cdot 40}{1000 \cdot 1.5} \cdot 23{,}3 = 464{,}521\text{лк}$$

E_{min}=300 лк берем из таблицы 3.12 (см. список литературы первая МУ)

$E_{АГ} \geq E_{min}$(т.к. освещенность незначительно больше нормированного освещения нужно увеличить количество светильников до 28 шт).

В данной главе был проведен расчет естественного освещения. При проверки естественного освещения помещений необходимо определить площадь световых проемов, обеспечивающих нормированное значение КЕО. В помещении для обеспечения нормированного значения КЕО, e_N= 0.96, при разряде IIIб окна 60 м^2 не обеспечивает нормирования значение зрительных работ требуется площадь световых проемов равная 85,05 м^2.

Далее был проведен расчет искусственного освещения. Расчет освещенности точечным методом показал, что заданного числа светильников было меньше необходимого для обеспечения достаточной освещенности помещения. Для обеспечения необходимой освещенности помещения необходимо увеличить количество светильников до 28, чтобы обеспечить нормальное освещение для человеческого глаза.

Заключение

В ходе выполнения данной дипломной работы были рассмотрены нейросети, их виды, строение нейросети. Были изучены виды сетевых атак. Сконфигурирована и обучена искусственная нейронная сеть в программной среде Matlab, которая состояла из 2 слоев. Нейросеть была обучена распозновать сетевые атаки, а именно DoS атаки. Нейросеть определяла сетевую атаку, приписывая значение этому соединение равное 1, а обычному – 0. Сперва была обучена нейросеть с 21 нейроном, после чего количество нейронов уменьшалось для сравнения результатов обучения. В итоге нейросеть прошла тестирование и смогла определить атаку.

Список литературы

1 TheKatherin Обзор методов Data Mining. URL: http://intellect-tver.ru/?p=165

2 Теуво Кохонен “Самоорганизующиеся карты” Издательство БИНОМ. Лаборатория знаний: 2014 С.: 659 ISBN: 978-5-99-63-1348-8

3 Боршевников А. Е. Сетевые атаки. Виды. Способы борьбы [Текст] // Современные тенденции технических наук: материалы Междунар. науч. конф. (г. Уфа, октябрь 2011 г.). — Уфа: Лето, 2011. — С. 8-13. — URL: https://moluch.ru/conf/tech/archive/5/1115/ (дата обращения: 10.02.2019).

4 Статья. Алгоритм Левенберга-Марквардта. URL: http://www.machinelearning.ru/wiki/index.php?title=Алгоритм_Левенберга-Марквардта (дата обращения10.02.2019).

5 «Искусственный интеллект» 4’2005 618 УДК 004. 6 Е. В. Малащук, Д. В. Бабин, С. М. Вороной, М. Г. Кочеткова

6Дюк В. А. Санкт-Петербургский институт информатики и автоматизации РАН: DataMining – интеллектуальный анализ данных \ URL: http://www.inftech.webservis.ru/it/database/datamining/ar2.html

7 Электронное издание «Википедия». URL: https://ru.wikipedia.org/wiki/ Искусственная нейронная сеть (дата обращения 20.02.2019)

8 Зенин А. В. Исследование возможностей использования нейронных сетей // Молодой ученый. — 2017. — №16. — С. 130-140. URL https://moluch.ru/archive/150/42394/ (дата обращения: 12.03.2019).

9 Ахметов Б.С., Горбаченко В.И. А95 Лабораторный практикум по курсу «Нейронные сети». – Алматы: ТОО «Издательство LEM», 2015. – 152 с.

10 Галушкин А. И. Нейронные сети: Основы теории. — М.: Горячая Линия — Телеком, 2012. — 496 с.

11Тархов Д. А. Нейросетевые модели и алгоритмы. Справочник. — М.: Радиотехника, 2014. — 352 с.

12 Рашитов Э. Э., Стоякова К. Л., Ибраев Р. Р. Модель математической нейронной сети // Молодой ученый. — 2017. — № 15 (149). — С. 77–80.

13 Documentation // Java Neural Network Framework Neuroph. URL: http://neuroph.sourceforge.net/documentation.html (дата обращения: 25.03.2019).

14 14 Грачева А.В.Понятие интеллектуальной информационной системы — URL: https://mylektsii.ru/ (дата обращения: 23.04.2019).