Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и телекоммуникаций
Кафедра информационной безопасности автоматизированных систем

Утверждена распоряжением по институту
От «02» марта 2020 г. № 013-р/12.00

Допущена к защите
«18» июня 2020 г.
Зав. кафедрой информационной безопасности автоматизированных систем
канд. техн. наук, профессор
А. Ф. Чипига

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ АТАК НА СЕТЕВЫЕ РЕСУРСЫ С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

Рецензенты:
Ерин Евгений Борисович
руководитель обособленного структурного подразделения ООО «Информационные системы и аутсорсинг» в городе Ставрополь

Выполнил:
Нуйкин Кирилл Алексеевич
Студент 5 курса, группы ИБС-с-о-15-2 специальности 10.05.03 «Информационная безопасность автоматизированных систем» специализация «Защищенные автоматизированные системы управления» очной формы обучения

Нормоконтролер:
Гиш Татьяна Александровна
доцент кафедры информационной безопасности автоматизированных систем

Научный руководитель:
Соломонов Дмитрий Владимирович
доцент кафедры информационной безопасности автоматизированных систем

Дата защиты:
« _____» _____ 2020 г.

Оценка: _____

Ставрополь, 2020 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и телекоммуникаций
Кафедра информационной безопасности автоматизированных систем

Утверждена распоряжением по институту
От «02» марта 2020 г. № 013-р/12.00

Допущена к защите
«18» июня 2020 г.
Зав. кафедрой информационной безопасности автоматизированных систем
канд. техн. наук, профессор
А. Ф. Чипига
(подпись)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ (ДИПЛОМНОМУ ПРОЕКТУ) НА ТЕМУ:

РАЗРАБОТКА СИСТЕМЫ ОБНАРУЖЕНИЯ АТАК НА СЕТЕВЫЕ РЕСУРСЫ С ПРИМЕНЕНИЕМ НЕЙРОСЕТЕВЫХ ТЕХНОЛОГИЙ

Автор дипломного проекта 17.06.2020 г. Нуйкин Кирилл Алексеевич
подпись

Специальность 10.05.03 Информационная безопасность автоматизированных систем

Специализация Защищенные автоматизированные системы управления

Группа ИБС-с-о-15-2

Руководитель проекта 17.06.2020 г. Д. В. Соломонов
подпись

Консультантыы по разделам:
Безопасность и экологичность проекта Д. В. Соломонов

Организационно-экономический раздел Д. В. Соломонов

Нормоконтролер Т. А. Гиш

Ставрополь, 2020 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и телекоммуникаций
Кафедра информационной безопасности автоматизированных систем
Специальность 10.05.03 Информационная безопасность автоматизированных систем
Специализация Защищенные автоматизированные системы управления

«УТВЕРЖДАЮ»
Зав. кафедрой информационной безопасности автоматизированных систем, канд. техн. наук, профессор
Чипига А. Ф.
«13» декабря 2019 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
(ДИПЛОМНЫЙ ПРОЕКТ)

Студент Нуйкин Кирилл Алексеевич группа ИБС-с-о-15-2
1. Тема Разработка системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий
Утверждена распоряжением по институту № 013-р/12.00 от «02» марта 2020 г.
2. Срок представления проекта к защите «18» июня 2020 г.
3. Исходные данные для проектирования набор данных CICIDS2017

4. Содержание пояснительной записки:
4.1 Анализ области использования нейронной сети для системы обнаружения атак на сетевые ресурсы.
4.2 Разработка системы обнаружения атак на сетевые ресурсы с применением с применением нейросетевых технологий.
4.3 Разработка и тестирование модели нейронной сети.
4.4 Безопасность и экологичность проекта
4.5 Организационно-экономический раздел
4.6 Другие разделы проекта
5. Перечень графического материала

Дата выдачи задания « 13 » декабря 2019 г.
Руководитель проекта Д. В. Соломонов
подпись

Консультанты по разделам:
безопасность и экологичность проекта Д. В. Соломонов
подпись

организационно-экономический раздел Д. В. Соломонов
подпись

Задание принял к исполнению «13» декабря 2019 г. К. А. Нуйкин
подпись

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и телекоммуникаций
Кафедра информационной безопасности автоматизированных систем
Специальность 10.05.03 Информационная безопасность автоматизированных систем
Специализация Защищенные автоматизированные системы управления

КАЛЕНДАРНЫЙ ПЛАН

Фамилия, имя, отчество Нуйкин Кирилл Алексеевич
Тема ВКР Разработка системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий

Руководитель доцент Соломонов Д. В.
Консультанты безопасность и экологичность работы доцент Соломонов Д. В.
организационно-экономический раздел доцент Соломонов Д. В.

№	Наименование этапов выпускной квалификационной работы	Срок выполнения работы	Примечание
1.	Подбор и анализ научно-технической литературы по теме исследования	13.12.19 – 31.01.20	
2.	Анализ области использования нейронной сети для системы обнаружения атак на сетевые ресурсы	01.02.20 – 16.02.20	
3.	Разработка системы обнаружения атак на сетевые ресурсы с применением с применением нейросетевых технологий	17.02.20 – 08.03.20	
4.	Разработка и тестирование модели нейронной сети	09.03.20 – 30.03.20	
5.	Оценка безопасности и экологичности работы, оценка технико-экономической эффективности работы	31.04.20 – 19.05.20	
6.	Оформление пояснительной записки	20.05.20 – 05.06. 20	
7.	Подготовка доклада и презентационного материала	06.06.20 – 12.06.20	
8.	Представление выпускной квалификационной работы на кафедру, предварительная защита, брошюровка пояснительной записки, получение допуска к защите.	12.06.20 – 18.06.20	

Научный руководитель _____ Соломонов Д. В.
подпись

Зав. кафедрой _____ Чипига А. Ф.
подпись

«13» декабря 2019 г.

Содержание

Введение

В настоящее время важной задачей в области обеспечения информационной безопасности является обнаружение атака на сетевые ресурсы. Успешно реализованная атака на сетевые ресурсы может привести как к потере информации, так и к несанкционированному доступу к информации, но и что не мало важно информация может быть незаметно искажена. Это актуализирует необходимость разработки и использования эффективных методов и средств обнаружения сетевых атак для защиты информации в компьютерных системах и сетях.

Целью дипломного проекта является разработка системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий.

Задачи дипломного проекта:

1. Рассмотрение существующих методов анализа и классификации трафика.
2. Исследование технологий и программного обеспечения используемых для разработки компонентов системы.
3. Проектирование и разработка сборщика данных.
4. Проектирование разработка анализирующего модуля.
5. Проектирование и разработка веб-приложения для просмотра подозрительной активности.
6. Проектирование и разработка двух моделей нейронных сетей для обнаружения разных атак.

Объектом исследования дипломного проекта является составленная и обученная нейронная сеть. Предметом исследования выступает трафик сетевой трафик.

Выпускная квалификационная работа состоит из введения, 5 глав, заключения, списка используемой литературы и приложений.

В первой главе произведена классификация сетевых атак, обозначены возможные методы воздействия на систему и последствия их воздействий. Обозначены основные методы обнаружения сетевых атак, выделены их плюсы и минусы. Так же проведено исследование технологий и программного обеспечения, используемых для разработки системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий.

Во второй главе произведено определение компонентов системы в ходе которого было уставлено какие компоненты необходимо разработать. Произведена разработка сборщика данных. Произведена разработка анализирующего модуля. Произведена разработка веб-приложения для просмотра данных, помеченных как аномальные.

В третьей главе произведена разработка двух моделей нейронных сетей, так же произведено общее тестирование системы.

В четвертой главе описаны вредные и опасные производственные факторы, действующие на пользователя ЭВМ.

В пятой главе проанализированы технико-экономические показатели, проведен расчет оценочной стоимости проекта.

В приложениях показаны исходный код программы сборщика данных, программы анализатора, веб-приложения на языке Python.

1. Анализ области использования нейронной сети для системы обнаружения атак на сетевые ресурсы

1.1. Классификация сетевых атак

Сетевая атака – действия с применением программных и (или) технических средств и с использованием сетевого протокола, направленные на реализацию угроз несанкционированного доступа к информации, воздействия на нее или на ресурсы автоматизированной информационной системы [1].

Глобально сетевые атаки можно разделить последующим признакам:

1. По характеру воздействия. Характер воздействия может быть активным и пассивным. Под активным воздействием понимается воздействие, которое оказывает влияние на работу системы. Это влияние характеризуется нарушением работоспособности системы либо изменением ее конфигурации. Особенностью активного воздействия является способность его обнаружение, так как в результате атаки происходят изменения в системе. Под пассивным воздействие, понимается воздействие, которое обычно не оказывает влияние на систему. Но может привести к нарушению политики безопасности. Такое воздействие практически невозможно обнаружить.

2. По цели воздействия. Целями воздействия является нарушение целостности, конфиденциальности данных, либо нарушение доступности системы. При нарушении целостности данных искажается информация, что делает невозможным дальнейшее ее использование в работе. Данное информационное разрушающее воздействие представляет собой пример активного воздействия. При нарушении конфиденциальности данных происходит разглашение информации. Нарушение конфиденциальности является примером пассивного воздействия. При нарушении доступности системы, система переходит полностью или частично в статус «отказ в обслуживании», что приводит к остановки внутренних процессов предприятия, либо организации.

3. По наличию обратной связи с атакуемым объектом. Атаки бывают с обратной связью и без обратной связи. Атаки с обратной связью позволяют атакующему на основе ответов от системы реагировать на изменения, происходящие в системе. Атаки без обратной связи не требуют реакции атакующего на изменения происходящие в системе.

4. По расположению нарушителя относительно атакуемого объекта. По расположению нарушителя относительно атакуемого объекта существуют межсегментные и внутрисегментные атаки. Межсегментные атаки значительно сложнее осуществить, но они являются наиболее опасными, так как атакующий и объект для атаки могут находиться на расстояние многих тысяч километров, что может значительно воспрепятствовать мерам по отражению атаки.

5. По уровню модели OSI, на котором осуществляется воздействие. Сетевые атаки могут проведены на всех уровнях модели OSI.

1.2. Методы обнаружения сетевых атак

Обнаружение сетевых атак – это процесс распознавания и реагирования на подозрительную деятельность, направленную на сетевые или вычислительные ресурсы организации [2, 3]. От используемых методов анализа, которые применяются для обнаружения сетевых атак сильно зависит эффективность обнаружения атак. В настоящее время можно выделить три основных метода обнаружения сетевых атак, это статистический анализ, экспертные системы и нейронные сети.

Каждый из методов имеет определенные недостатки, поэтому во многих системах используется комбинированный подход к обнаружению атак. Такой подход позволяет избавиться от недостатков разных систем и достичь наибольшей защиты сетевых ресурсов.

1. Статистический анализ. При использовании данного метода защиты определяются профили для каждого субъекта системы, любое отклонение от профиля субъекта расценивается как несанкционированная деятельность.

Статистические методы универсальные, так как не требуют знаний о возможных атаках на систему. Однако при их использовании могут возникать некоторые трудности, связанные, например, с тем, что их можно «обучить» воспринимать несанкционированные действия как нормальные. Этот подход имеет два основных преимущества: использование зарекомендовавшего себя аппарата математической статистики и адаптация к поведению субъекта [4].

2. Экспертные системы. Этот метод является достаточно распространенным. При его применении информация об атаках формулируется в виде правил или в форме сигнатур. Сигнатура является шаблоном, которая характеризует определенную атаку. Если срабатывается одно из правил, либо происходящие действия в системе совпадают с одной из сигнатур, то такие действия определяются как несанкционированные. Такой метод характеризуется практические полным отсутствием ложных тревог. Но при этом чтобы система защиты оставалась актуально необходимо постоянно обновлять базы данных атак. Недостатком данного метода то, что такие системы неустойчивы к модификациям атак и не могут автоматически адаптироваться.

Нейронные сети. Изменяющаяся природа сетевых атак требует разработки гибкой адаптивной системы защиты, которая способна анализировать большое количество сетевого трафика с постоянно меняющимися условиям сетевой активности. Альтернативой систем, основанных на правилах, являются нейронные сети [5]. В отличие от экспертных систем [6], которые могут дать пользователю определенный ответ, соответствуют или нет рассматриваемые характеристики атаки характеристикам, заложенным в базе знаний, нейронная сеть проводит анализ информации и предоставляет возможность оценить, что анализируемые данные согласуются с характеристиками, которые она научена распознавать [7]. Основными плюсами в использования нейронных сетей в задачах обнаружения сетевых атак можно выделить [8]:

– гибкость нейронной сети (сеть способна анализировать данные сетевого трафика в условиях их искажения или неполноты);

– быстрая скорость обработки информации (поскольку защита вычислительных ресурсов требует оперативной идентификации атак, скорость обработки в нейронной сети может быть достаточной для реагирования в реальном режиме времени на проводимые атаки до того, как в системе появятся непоправимые повреждения);

– возможность прогнозирования (выходные данные нейронной сети могут интерпретироваться в форме вероятности, что предоставляет возможность прогнозирования дальнейшего развития атаки; нейросетевая система обнаружения вторжений может определять вероятность того, что отдельное событие либо серия событий указывают на атаку, и провести защитные мероприятия прежде, чем атака будет успешно реализована);

– способность анализировать характеристики умышленных атак и идентифицировать элементы, которые не похожи на те, что наблюдались в сети при ее обучении (нейронная сеть может быть обучена распознавать известные подозрительные события с высокой степенью точности, а также может использовать эти знания для идентификации атак, которые неточно соответствуют характеристикам предыдущих вторжений).

1.3. Исследование технологий и программного обеспечения, используемых для разработки системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий

Нейронные сети – это одно из направлений исследований в области искусственного интеллекта, основанное на попытках воспроизвести нервную систему человека, а именно: способность нервной системы обучаться и исправлять ошибки, что должно позволить смоделировать, хотя и достаточно грубо, работу человеческого мозга [9].

К задачам, решаемым нейронными сетями, можно отнести:

– распознавание образов и классификация. На вход нейронной сети подается изображение, она определяет объект, находящийся на этом изображении. Во время обучения нейронной сети на вход подается вектор значений и признаков образа с указанием принадлежности к определенному классу;

– задачи кластеризации. Задачи кластеризации, известные так же, как классификация образов «без учителя», не имеют обучающей выборки с метками классов. Их принцип работы основан на выявлении закономерностей и подобия между образами и размещении этих образов в один кластер или категорию;

– аппроксимация функций. Задача аппроксимации, заключается в нахождении оценки неизвестной функции, по ее значениям. Точность аппроксимации зависит от выбора структуры нейронной сети;

– оптимизация. Способы оптимизации с использованием нейронных сетей, подразумевают нахождение такого решения, которое удовлетворяет системе условий и минимизирует целевую функцию. В качестве примера можно рассмотреть алгоритм, реализующий некую последовательность действий, который можно полностью заменить функционированием нейронной сети, используя правила, по которым он составлен.

Нейронная сеть состоит из нейронов. Каждый нейрон является элементарной структурной единицей нейронной сети. На рисунке 1.1 изображена структурная схема нейрона.

Процесс обучения сети сводится к изменению весовых коэффициентов w_n. NET в данном случае и есть результат вычислений нейрона. Результаты вычисления передаются на выход через функцию активации.

Функция активации нейрона – это функция, которая вычисляет выходной сигнал нейрона [10]. На вход этой функции подается сумма всех произведений сигналов и весов этих сигналов.

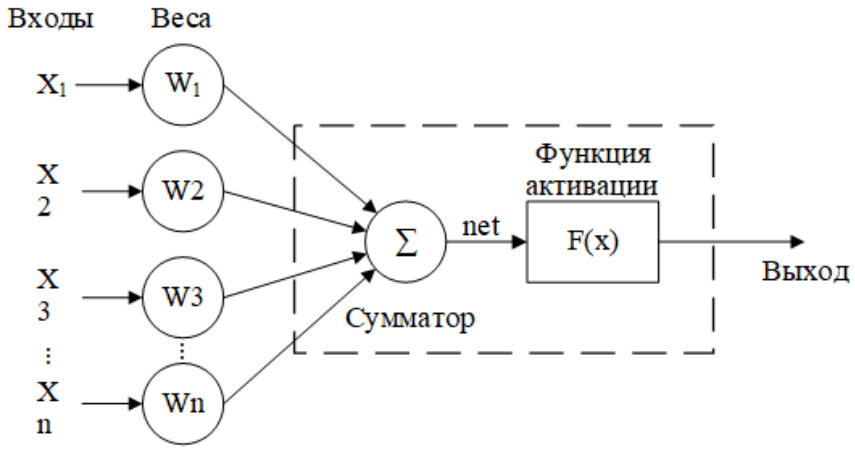


Рисунок 1.1 – Структурная схема нейрона

Можно выделить следующие функции активации:

1. Единичный скачок или жесткая пороговая функция. Функция описывается следующей формулой:

$$\text{Out} = \begin{cases} 0, \text{NET} < \theta \\ 0, \text{NET} \geq \theta \end{cases} \quad (1)$$

Пока средневзвешенная сумма меньше определенного значения, функция активации возвращает ноль, а когда становится больше – единицу.

2. Сигмоидальная функция или сигмоид. Формула описывающая сигмоид:

$$\text{Out} = \frac{1}{1 + e^{-NET}} \quad (2)$$

Часто применяется в многослойных нейронных и других сетях с непрерывными сигналами. Гладкость и непрерывность функции – важные положительные качества.

3. Гиперболический тангенс. Функция описывается следующими формулами:

$$\text{Out} = \text{th}(\text{NET}) \quad (3)$$

или

$$Out = \frac{e^{NET} - e^{-NET}}{e^{NET} + e^{-NET}} \quad (4)$$

Также часто применяется в сетях с непрерывными сигналами. Ее особенность в том, что она может возвращать отрицательные значения результата.

Архитектурно в нейросетях можно выделить несколько основных типов:

1. Многослойные сети. В многосвязных (или многослойных) сетях нейроны объединяются в слои. Слой содержит совокупность нейронов с едиными входными сигналами. Число нейронов в каждом слое может быть любым и никак заранее не связано с количеством нейронов в других слоях. В общем случае сеть состоит из Q слоев, пронумерованных слева направо. Внешние входные сигналы подаются на входы нейронов первого слоя (входной слой часто нумеруют как нулевой), а выходами сети являются выходные сигналы последнего слоя. Вход нейронной сети можно рассматривать как выход «нулевого слоя» вырожденных нейронов, которые служат лишь в качестве распределительных точек, суммирования и преобразования сигналов здесь не производится. Кроме входного и выходного слоев в многослойной нейронной сети есть один или несколько промежуточных (скрытых) слоев. Связи от выходов нейронов некоторого слоя q к входам нейронов следующего слоя (q+1). Структурная схема многослойной сети представлена на рисунке 1.2.

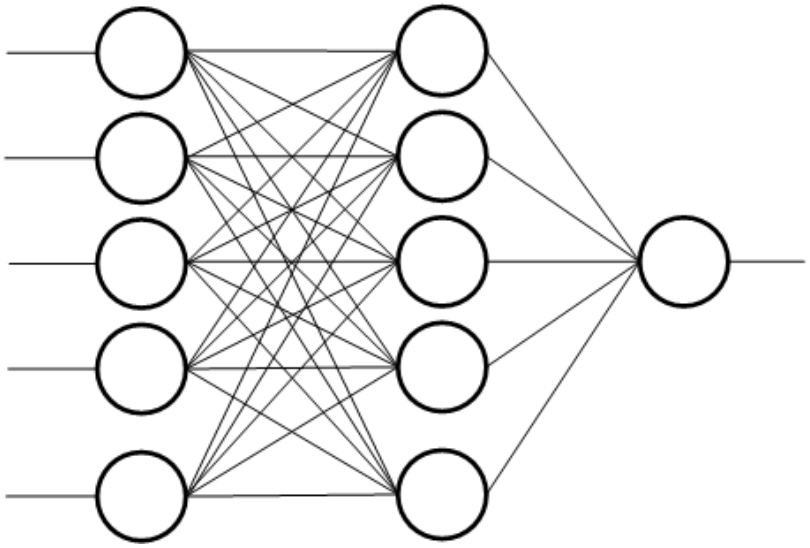


Рисунок 1.2 – Структурная схема многослойной сети

2. Полносвязные сети. Полносвязные сети представляют собой ИНС, каждый нейрон которой передает свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные сигналы подаются всем нейронам. Выходными сигналами сети могут быть все или некоторые выходные сигналы нейронов после нескольких тактов функционирования сети. Структурная схема полносвязной сети представлена на рисунке 1.3.

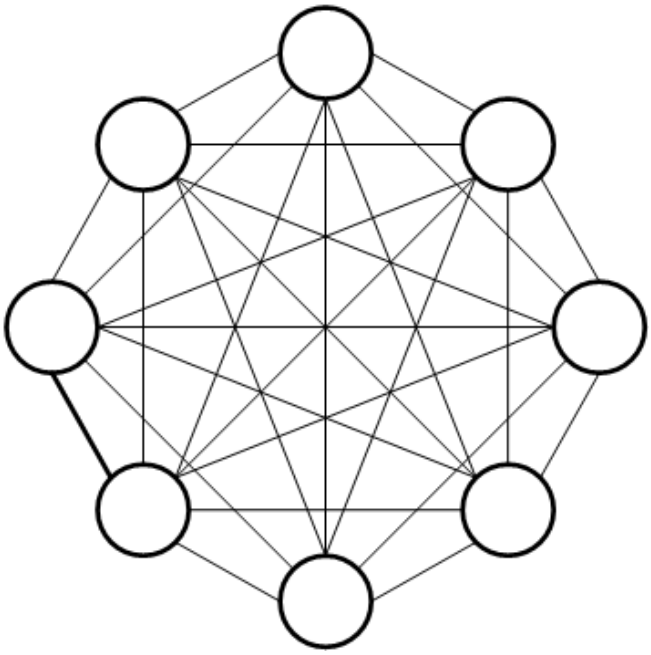


Рисунок 1.3 – Структурная схема полносвязной сети

3. Слабосвязные сети. Слабосвязные сети (нейронные сети с локальными связями) представляют собой слоистые сети с небольшим количеством связей. Структурная схема слабосвязных сетей представлена на рисунке 1.4.

Самым важным свойством нейронных сетей является их способность обучаться на основе данных окружающей среды и в результате обучения повышать свою производительность. Повышение производительности происходит со временем в соответствии с определенными правилами. Обучение нейронной сети происходит посредством интерактивного процесса корректировки синаптических весов и порогов. В идеальном случае нейронная

сеть получает знания об окружающей среде на каждой итерации процесса обучения [11].

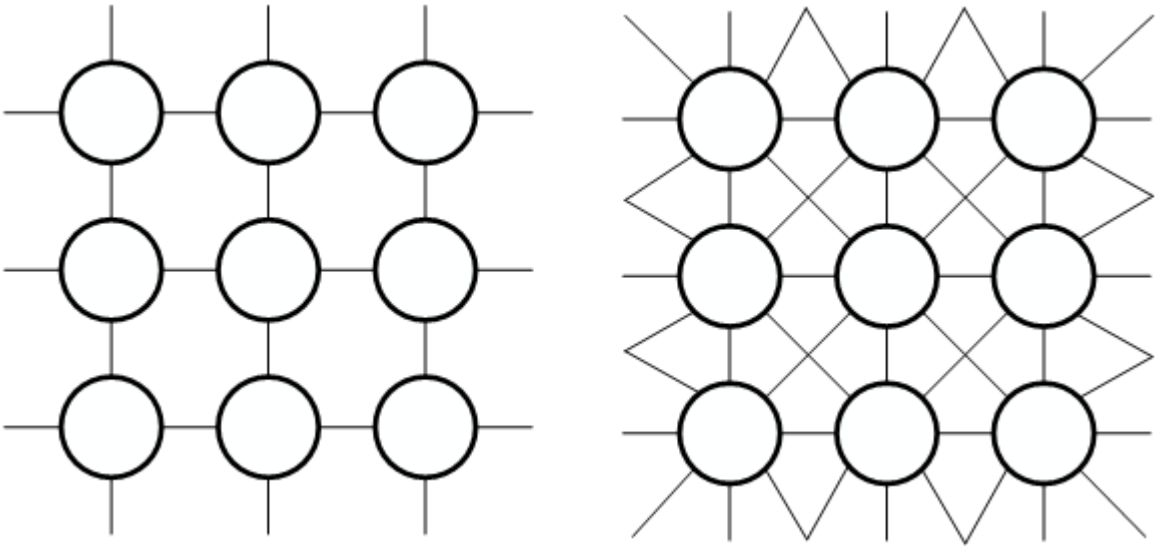


Рисунок 1.4 – Структурная схема слабосвязных сетей

Существуют два концептуальных подхода к обучению нейронных сетей: обучение с учителем и обучение без учителя.

Обучение нейронной сети с учителем предполагает, что для каждого входного вектора из обучающего множества существует требуемое значение выходного вектора. Эти вектора образуют обучающую пару и веса в сети меняются до тех пор, пока приемлемый уровень отклонения между векторами не будет достигнут. Алгоритм обучения с учителем представлен на рисунке 1.5.

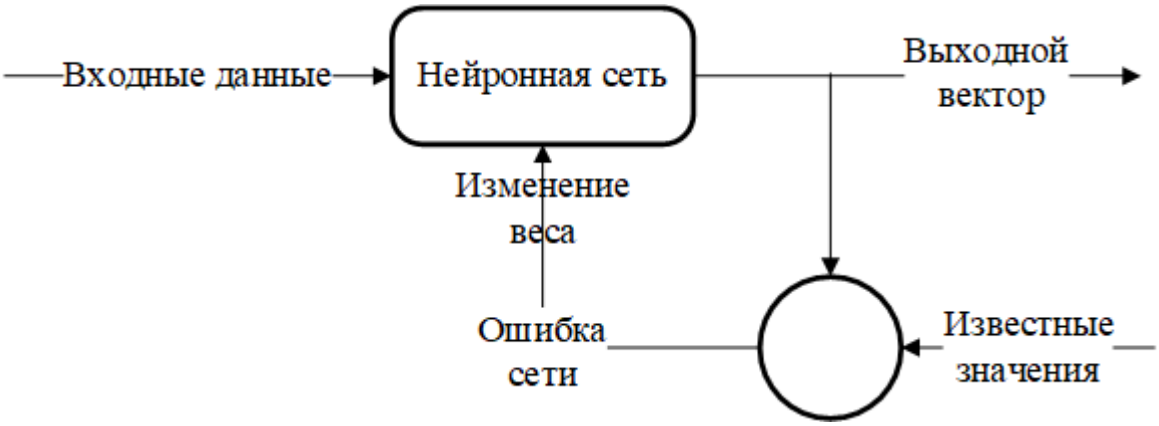


Рисунок 1.5 – Алгоритм обучения нейронной сети с учителем

Обучение нейронной сети без учителя предполагает то, что обучающее множество состоит только из входных векторов. Алгоритм обучения сети подстраивает веса так, чтобы получались согласованные выходные векторы.

При этом соблюдается следующая последовательность событий [12]:

1) в нейронную сеть поступают внешние сигналы (входящие параметры);

2) свободные параметры сети меняются;

3) после изменений нейронная сеть отвечает на входящие сигналы уже другим образом.

Алгоритм обучения без учителя представлен на рисунке 1.6.

Рисунок 1.6 – Алгоритм обучения нейронной сети без учителя

Для обучения модели нейронной сети для системы обнаружения атак на сетевые ресурсы будет использоваться подход с учителем, обучение будет производиться на наборе данных CICIDS2017.

Набор данных CICIDS2017 содержит доброкачественные и самые современные распространенные атаки, которые напоминают реальные данные (PCAPs). Он также включает в себя результаты анализа сетевого трафика с использованием CICFlowMeter с маркированными потоками на основе временной метки, IP-адреса источника и назначения, портов источника и назначения, протоколов и атак (CSV-файлы). Описание параметров, содержащихся в датасете приведено в таблице 1.1.

Таблица 1.1 – Описание параметров датасета CICIDS2017

№ п/п	Название параметра	Описание
1	2	3
1	Destination Port	Порт назначения

Продолжение таблицы 1.1

№ п/п	Название параметра	Описание
1	2	3
2	Flow Duration	Продолжительность потока
3	Total Fwd Packets	Всего пакетов в прямом направлении
4	Total Backward Packets	Всего пакетов в обратном направлении
5	Total Length of Fwd Packets	Общий размер пакетов в прямом направлении
6	Total Length of Bwd Packets	Общий размер пакетов в обратном направлении
7	Fwd Packet Length Max	Максимальный размер пакета в прямом направлении
8	Fwd Packet Length Min	Минимальный размер пакета в прямом направлении
9	Fwd Packet Length Mean	Средний размер пакета в прямом направлении
10	Fwd Packet Length Std	Размер стандартного отклонения пакета в прямом направлении
11	Bwd Packet Length Max	Максимальный размер пакета в обратном направлении
12	Bwd Packet Length Min	Минимальный размер пакета в обратном направлении
13	Bwd Packet Length Mean	Средний размер пакета в обратном направлении
14	Bwd Packet Length Std	Размер стандартного отклонения пакета в обратном направлении
15	Flow Bytes/s	скорость потока в байтах, то есть количество пакетов, передаваемых в секунду
16	Flow Packets/s	скорость потока пакетов, то есть количество пакетов, переданных в секунду
17	Flow IAT Mean	Среднее время между двумя потоками
18	Flow IAT Std	Стандартное отклонение времени двух потоков
19	Flow IAT Max	Максимальное время между двумя потоками
20	Flow IAT Min	Минимальное время между двумя потоками
21	Fwd IAT Total	Общее время между двумя пакетами, отправленными в прямом направлении
22	Fwd IAT Mean	Среднее время между двумя пакетами, отправленными в прямом направлении
23	Fwd IAT Std	Время стандартного отклонения между двумя пакетами, отправленными в прямом направлении
24	Fwd IAT Max	Максимальное время между двумя пакетами, отправленными в прямом направлении
25	Fwd IAT Min	Минимальное время между двумя пакетами, отправленными в прямом направлении
26	Bwd IAT Total	Общее время между двумя пакетами, отправленными в обратном направлении
27	Bwd IAT Mean	Среднее время между двумя пакетами, отправленными в обратном направлении
28	Bwd IAT Std	Время стандартного отклонения между двумя пакетами, отправленными в обратном направлении
29	Bwd IAT Max	Максимальное время между двумя пакетами, отправленными в обратном направлении
30	Bwd IAT Min	Минимальное время между двумя пакетами,

Продолжение таблицы 1.1

№ п/п	Название параметра	Описание
1	2	3
		отправленными в обратном направлени
31	Fwd PSH Flags	Количество раз, когда флаг PSH был установлен в пакетах, движущихся в прямом направлении (0 для UDP)
32	Bwd PSH Flags	Количество раз, когда флаг PSH был установлен в пакетах, движущихся в обратном направлении (0 для UDP)
33	Fwd URG Flags	Количество раз, когда флаг URG был установлен в пакетах, проходящих в прямом направлении (0 для UDP)
34	Bwd URG Flags	Сколько раз был установлен флаг URG в пакетах, движущихся в обратном направлении (0 для UDP)
35	Fwd Header Length	Всего байт, используемых для заголовков в прямом направлении
36	Bwd Header Length	Всего байт, используемых для заголовков в прямом направлении
37	Fwd Packets/s	Количество прямых пакетов в секунду
38	Bwd Packets/s	Количество обратных пакетов в секунду
39	Min Packet Length	Минимальная длина пакета
40	Max Packet Length	Максимальная длина пакета
41	Packet Length Mean	Средняя длина пакета
42	Packet Length Std	Стандартное отклонение длины пакета
43	Packet Length Variance	Минимальное время прибытия пакета
44	FIN Flag Count	Количество пакетов с FIN
45	SYN Flag Count	Количество пакетов с SYN
46	RST Flag Count	Количество пакетов с RST
47	PSH Flag Count	Количество пакетов с PUSH
48	ACK Flag Count	Количество пакетов с ACK
49	URG Flag Count	Количество пакетов с URG
50	CWE Flag Count	Количество пакетов с CWE
51	ECE Flag Count	Количество пакетов с ЕЭК
52	Down/Up Ratio	Коэффициент загрузки и выгрузки
53	Average Packet Size	Средний размер пакета
54	Avg Fwd Segment Size	Средний размер сегмента в прямом направлении
55	Avg Bwd Segment Size	Средний размер сегмента в обратном направлении
56	Fwd Header Length	Размер заголовка в прямом направлении
57	Fwd Avg Bytes/Bulk	Средний объем байтов в прямом направлении
58	Fwd Avg Packets/Bulk	Среднее количество пакетов в прямом направлении
59	Fwd Avg Bulk Rate	Среднее количество насыпного курса в прямом направлении
60	Bwd Avg Bytes/Bulk	Средний объем байтов в обратном направлении
61	Bwd Avg Packets/Bulk	Среднее количество пакетов в обратном направлении
62	Bwd Avg Bulk Rate	Среднее количество пакетов в обратном

Продолжение таблицы 1.1

№ п/п	Название параметра	Описание
1	2	3
		направлении
63	Subflow Fwd Packets	Среднее количество пакетов в подпотоке в прямом направлении
64	Subflow Fwd Bytes	Среднее количество байтов в подпотоке в прямом направлении
65	Subflow Bwd Packets	Среднее количество пакетов в подпотоке в обратном направлении
66	Subflow Bwd Bytes	Среднее количество байтов в подпотоке в обратном направлении
67	Init_Win_bytes_forward	Количество байтов, отправленных в начальном окне в прямом направлении
68	Init_Win_bytes_backward	Количество байтов, отправленных в начальном окне в обратном направлении
69	act_data_pkt_fwd	пакетов с не менее 1 байта полезной нагрузки данных TCP в прямом направлении
70	min_seg_size_forward	Минимальный размер сегмента наблюдается в прямом направлении
71	Active Mean	Среднее время, когда поток был активен, прежде чем стал свободным
72	Active Std	Стандартное отклонение времени, в течение которого поток был активен до простоя
73	Active Max	Максимальное время, в течение которого поток был активен до простоя
74	Active Min	Минимальное время активности потока до простоя
75	Idle Mean	Среднее время, когда поток простаивал, прежде чем стать активным
76	Idle Std	Стандартное отклонение времени, в течение которого поток простаивал до того, как стал активным
77	Idle Max	Максимальное время простоя потока до его активации
78	Idle Min	Минимальное время, в течение которого поток простаивал, прежде чем стать активным
79	Label	Маркировка пакета

В качестве программы для захвата трафика выбран CICFlowMeter. CICFlowMeter – генерирует двунаправленные потоки (Biflow), где первый пакет определяет прямое (источник к месту назначения) и обратное (место назначения к источнику) направления, отсюда получаются 84 статистические характеристики, такие как длительность, количество пакетов, количество байтов, длина пакетов, и т. д. также рассчитываются отдельно в прямом и

обратном направлении. Выходные данные приложения представляют собой формат файла CSV с шестью столбцами, помеченными для каждого потока, а именно FlowID, SourceIP, DestinationIP, SourcePort, DestinationPort и Protocol с более чем 80 функциями сетевого трафика.

Основным языком программирования для всех создаваемых приложения в дипломном проекте выбран Python. Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода [13]. Выбор Python в качестве языка программирования не случаен. Этот язык имеет низкий порог вхождения, возможность, без особых проблем, запуститься на любой операционной системе (ОС) и самое главное на этом языке написано множество библиотек для построения нейронных сетей.

Python Asyncio – asyncio предоставляет инфраструктуру для написания однопоточного параллельного кода с использованием сопрограмм, мультиплексирования доступа ввода-вывода через сокеты и другие ресурсы, запуска сетевых клиентов и серверов и других связанных примитивов. Вот более подробный список содержимого пакета:

– подключаемый цикл событий с различными системными реализациями;

– абстракции транспорта и протокола (аналогичные тем, что есть в Twisted);

– конкретная поддержка TCP, UDP, SSL, каналов подпроцесса, отложенных вызовов и других (некоторые могут зависеть от системы);

– класс Future, который имитирует класс в модуле concurrent.futures, но адаптирован для использования с циклом событий;

– сопрограммы и задачи, основанные на выходе из (PEP 380), чтобы помочь написать параллельный код в последовательном порядке;

– поддержка отмены фьючерсов и сопрограмм;

– примитивы синхронизации для использования между сопрограммами в одном потоке, имитируя их в поточном модуле;

– интерфейс для передачи работы в пул потоков, когда вам абсолютно необходимо использовать библиотеку, которая блокирует вызовы ввода / вывода.

Для реализации веб серверов и клиента будет использоваться подключаемая библиотека aiohttp.

Aiohttp – это HTTP Web сервер и клиент для asyncio (PEP-3156).

Так же для реализации веб приложения будет необходима библиотека Aiohttp Jinja2.

Aiohttp Jinja2 – это современный и удобный для разработчиков язык шаблонов для Python, созданный по образцу шаблонов Django. Он быстрый, широко используемый и безопасный благодаря дополнительной среде выполнения шаблонов в песочнице. Это текстовый шаблонизатор, поэтому он может быть использован для создания любого вида разметки, а также исходного кода. Лицензирован под BSD лицензией. Шаблонизатор Jinja позволяет настраивать:

– теги;

– фильтры;

– тесты;

– глобальные переменные.

Также, в отличие от шаблонизатора Django, Jinja позволяет конструктору шаблонов вызывать функции с аргументами на объектах. Jinja, как и Smarty, также поставляется с простой в использовании системой фильтров, похожей на конвейер Unix.

В качестве хранилища данных будет использоваться база данных PostgreSQL. PostgreSQL – свободная объектно-реляционная система управления базами данных (СУБД)[14].

Для работы с PostgreSQL будет использована подключаемая библиотека Asyncpg. Asyncpg – это библиотека интерфейса базы данных, разработанная специально для PostgreSQL и Python / asyncio. Asyncpg – это эффективная, чистая реализация двоичного протокола сервера PostgreSQL для использования с платформой Python asyncio. Одной из особенностей является то,что asyncpg изначально реализует серверный протокол PostgreSQL и предоставляет его возможности напрямую, а не скрывает их за общим фасадом, таким как DB-API.

Для работы с дасетом будут использоваться библиотеки pandas и NumPy.

Pandas – программная библиотека на языке Python для обработки и анализа данных. Работа pandas с данными строится поверх библиотеки NumPy, являющейся инструментом более низкого уровня. Предоставляет специальные структуры данных и операции для манипулирования числовыми таблицами и временными рядами [15].

NumPy – это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами [16].

В качестве библиотек используемых для обучения нейронных сетей будут выступать TensorFlow и Keras.

TensorFlow – открытая программная библиотека для машинного обучения, разработанная компанией Google для решения задач построения и тренировки нейронной сети с целью автоматического нахождения и классификации образов, достигая качества человеческого восприятия [17]. Данная библиотека позволяет без особых проблем создавать нейронные сети различной сложности, подготавливать данные для их обучения и тестирования. Так же TensorFlow предоставляет инструменты для визуализации процессов происходящих во время обучения. Визуализация является достаточно важной частью для понимания качества обучения нейронной сети. Но работа с

TensorFlow требует достаточно глубоких познаний в нейронных сетях для облегчения этой задачи существует библиотека Keras.

Keras – открытая нейросетевая библиотека, написанная на языке Python. Она представляет собой надстройку над фреймворками Deeplearning, TensorFlow и Theano[18]. Keras является высокоуровневым API для нейронных сетей, может работать поверх TensorFlow, CNTK или Theano. Он был разработан с целью обеспечения быстрого экспериментирования. Способность переходить от идеи к результату с наименьшими временными затратами.

1.4. Выводы

В главе произведена классификация сетевых атак, обозначены возможные методы воздействия на систему и последствия их воздействий. Обозначены основные методы обнаружения сетевых атак, выделены их плюсы и минусы.

Так же проведено исследование технологий и программного обеспечения, используемых для разработки системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий. Дано определение нейронным сетям и определены задачи, которые они способны решать. Произведен разбор базовой единицы нейронной сети – нейрона. Рассмотрены различные структуры нейронных сетей, такие как многослойные, полносвязные и слабосвязные сети.

Так же затронуты подходы к обучению и установлено, что для реализации модели нейронной сети будет использоваться обучение с учителем.

Произведен разбор датасета CICIDS2017, который будет использоваться для обучения и тестирования модели нейронной сети. Приведены описания каждого из параметров.

Обозначены основные инструменты, используемые для разработки системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий. Этими инструментами будут являться высокоуровневый язык программирования Python и подключаемые библиотеки Aiohttp, Aiohttp Jinja2, Asyncpg, Pandas, NumPy, TensorFlow и Keras.

2. Разработка системы обнаружения атак на сетевые ресурсы с применением с применением нейросетевых технологий

2.1. Определение компонентов системы

Так как для сбора трафика будет использоваться CICFlowMeter, а его единственным возможным выводом данных является постоянно дозаписываемый csv файл, то существует необходимость в создании интерфейса для его чтения и выдачи новых значений в удобном виде. Этим интерфейсом будет служить веб-сервер, который в фоне будет считывать csv файл, переформатировать считанные данные в формат json и выдавать их в виде веб-страницы.

Далее данные с веб-страницы будут считываться веб-клиентом и передаваться на анализ в нейронную сеть. По результатам анализа они будут помечены как нормальный, либо аномальный трафик. Аномальный трафик будет записан в базу данных.

Данные из базы данных будут формироваться и выводиться в виде небольших отчетов в специальном веб-приложении.

Такая структура системы обнаружения атак позволит создавать неограниченное количество анализирующих модулей, что позволит обрабатывать трафик в реальном времени. Анализирующие модули могут быть расположены на нескольких серверах организации. Так же при выходе из строя одного или нескольких анализирующих модулей система продолжит свою работу.

2.2. Разработка сборщика данных

Для начала разработки необходимо определиться с форматом входных данных. Для этого запустим программу CICFlowMeter и произведем захват трафика. Интерфейс программы представлен на рисунке 2.1

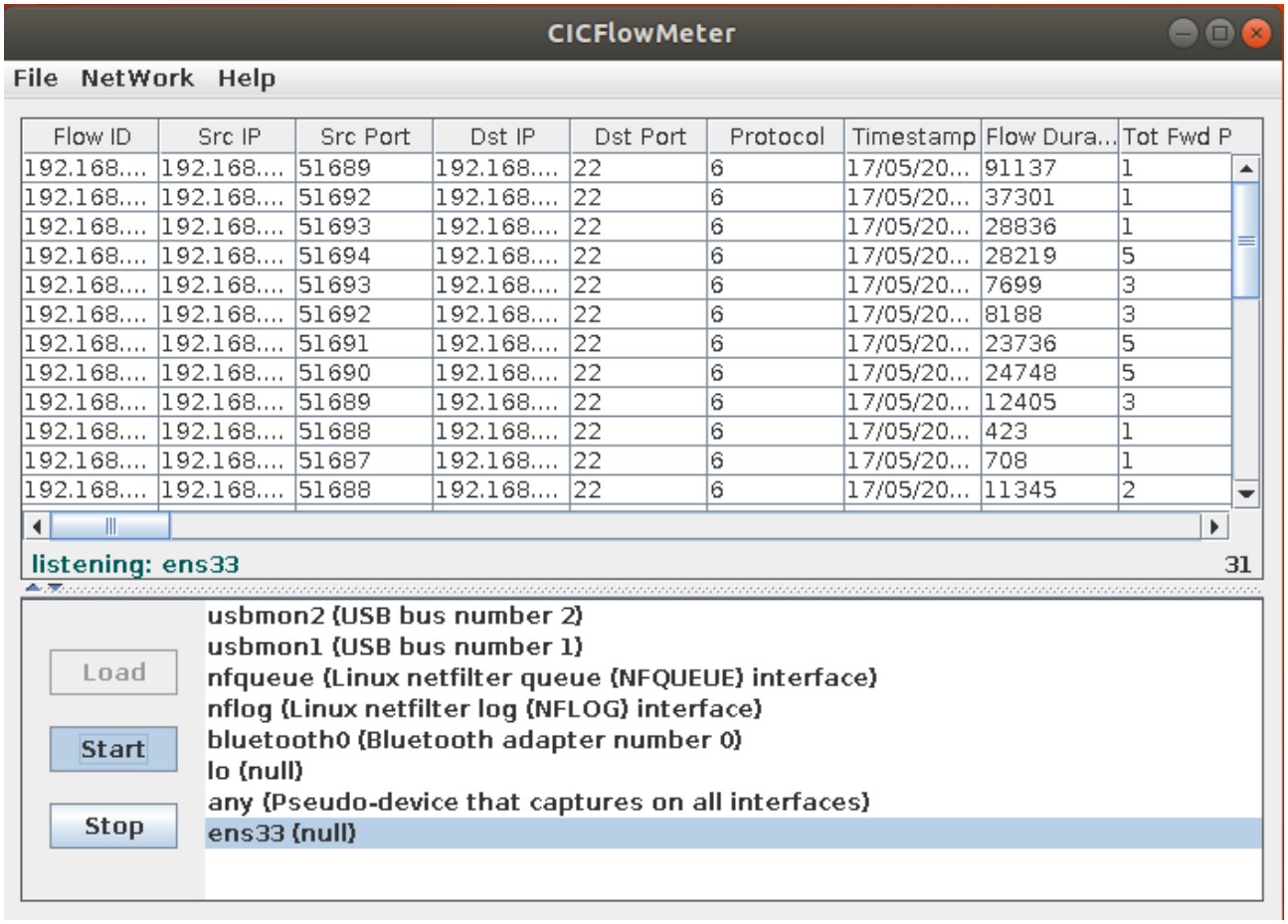


Рисунок 2.1 – Интерфейс программы CICFlowMeter

После начала захвата трафика программа создала файл 2020-05-17_Flow.csv. Пример полей в файле представлен в таблице 2.1.

Таблица 2.1 – Пример данных предоставляемых программой CICFlowMeter

№ п/п	Название параметра	Пример данных
1	2	3
1	Flow ID	152.199.19.161-192.168.1.2-443-51472-6
2	Src IP	192.168.1.2
3	Src Port	51472
4	Dst IP	152.199.19.161
5	Dst Port	443
6	Protocol	6
7	Timestamp	17/05/2020 02:47:24 AM
8	Flow Duration	299482
9	Tot Fwd Pkts	1
10	Tot Bwd Pkts	1

Продолжение таблицы 2.1

1	2	3
11	TotLen Fwd Pkts	0
12	TotLen Bwd Pkts	0
13	Fwd Pkt Len Max	0
14	Fwd Pkt Len Min	0
15	Fwd Pkt Len Mean	0
16	Fwd Pkt Len Std	0
17	Bwd Pkt Len Max	0
18	Bwd Pkt Len Min	0
19	Bwd Pkt Len Mean	0
20	Bwd Pkt Len Std	0
21	Flow Byts/s	0
22	Flow Pkts/s	6,678198
23	Flow IAT Mean	299482
24	Flow IAT Std	0
25	Flow IAT Max	299482
26	Flow IAT Min	299482
27	Fwd IAT Tot	0
28	Fwd IAT Mean	0
29	Fwd IAT Std	0
30	Fwd IAT Max	0
31	Fwd IAT Min	0
32	Bwd IAT Tot	0
33	Bwd IAT Mean	0
34	Bwd IAT Std	0
35	Bwd IAT Max	0
36	Bwd IAT Min	0
37	Fwd PSH Flags	0
38	Bwd PSH Flags	0
39	Fwd URG Flags	0
40	Bwd URG Flags	0
41	Fwd Header Len	20
42	Bwd Header Len	20
43	Fwd Pkts/s	3,339099
44	Bwd Pkts/s	3,339099
45	Pkt Len Min	0
46	Pkt Len Max	0
47	Pkt Len Mean	0
48	Pkt Len Std	0
49	Pkt Len Var	0
50	FIN Flag Cnt	1
51	SYN Flag Cnt	0
52	RST Flag Cnt	0
53	PSH Flag Cnt	0
54	ACK Flag Cnt	1
55	URG Flag Cnt	0
56	CWE Flag Count	0
57	ECE Flag Cnt	0
58	Down/Up Ratio	1

Продолжение таблицы 2.1

1	2	3
59	Pkt Size Avg	0
60	Fwd Seg Size Avg	0
61	Bwd Seg Size Avg	0
62	Fwd Byts/b Avg	0
63	Fwd Pkts/b Avg	0
64	Fwd Blk Rate Avg	0
65	Bwd Byts/b Avg	0
66	Bwd Pkts/b Avg	0
67	Bwd Blk Rate Avg	0
68	Subflow Fwd Pkts	1
69	Subflow Fwd Byts	0
70	Subflow Bwd Pkts	1
71	Subflow Bwd Byts	0
72	Init Fwd Win Byts	-1
73	Init Bwd Win Byts	1024
74	Fwd Act Data Pkts	0
75	Fwd Seg Size Min	0
76	Active Mean	0
77	Active Std	0
78	Active Max	0
79	Active Min	0
80	Idle Mean	0
81	Idle Std	0
82	Idle Max	0
83	Idle Min	0
84	Label	No Label

После того как формат входных данных определен приступим к разработке приложения. Для его реализации выбран язык python, так же будут задействована библиотека aiohttp.

Логика работы программы представлена на диаграммах активности, на рисунках 2.2-2.4.

Процесс веб-сервера и процесс чтения и преобразования данных из файла запускаются параллельно и выполняются до тех пор, пока не будет получен сигнал выхода из программы. После запуска программа очистит все файлы в директории будет ожидать новый, который должен создать CICFlowMeter. Далее будет происходить построчное чтение файла, форматирование и выдача данных в виде веб страницы.

Рисунок 2.2 – Диаграмма активности программы сборщика данных

Рисунок 2.3 – Диаграмма активности сборщика данных, веб-сервер

Рисунок 2.4 – Диаграмма активности программы сборщика данных, модуль чтения и преобразования данных из файла

Выходные данные программы представлены на рисунке 2.5.

Рисунок 2.5 – Выходные данные программы в формате json

Полный листинг программы определен в Приложении А.

2.3. Разработка анализирующего модуля

Для реализации анализирующего модуля так же выбран язык python с использованием библиотек aiohttp, asyncpg, pandas, keras и numpy. Логика работы программы представлена на диаграмме активности, на рисунке 2.6. Процесс анализа запускается в бесконечном цикле и работает до тех пор, пока не будет получен сигнал выхода из программы. Данные, которые будут получены от сборщика будут являться избыточными для анализа их нейронной сетью, но так как нейронные сети, настроенные под разные атаки используют разные поля из этих данных от этой избыточности нельзя отказаться.

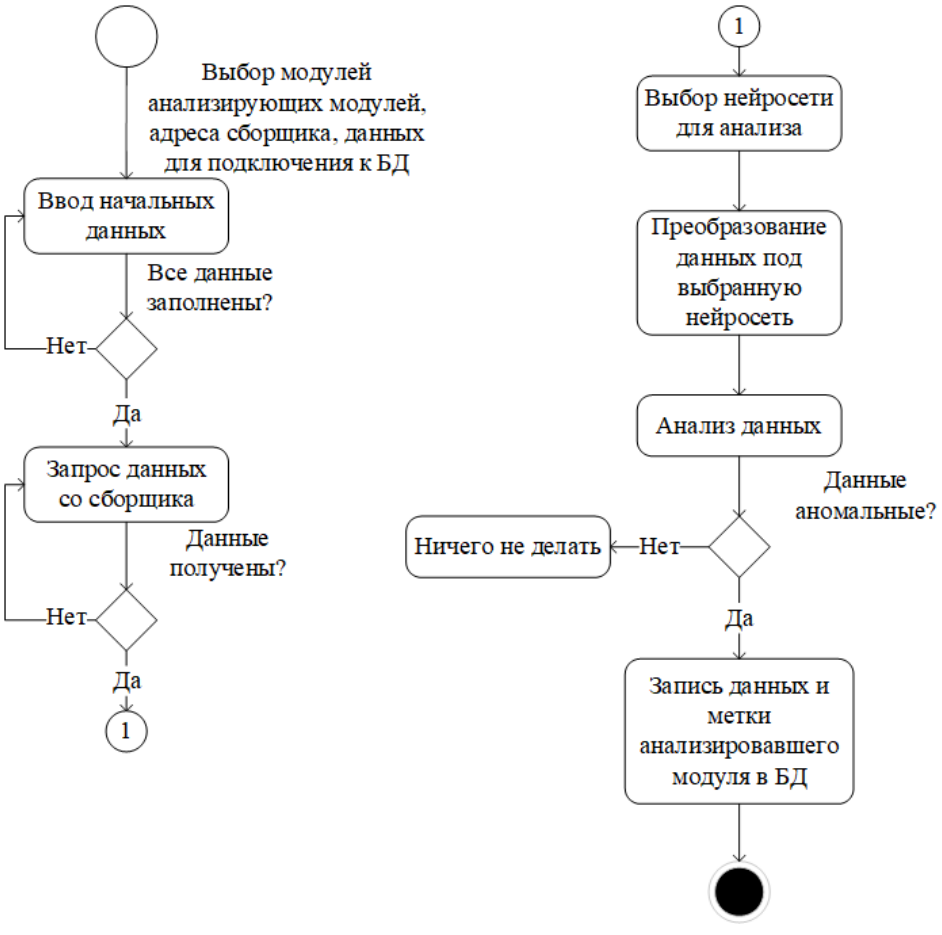


Рисунок 2.6 – Диаграмма активности программы анализатора

Стоит отметить, что содержимое файла nn_analyse.py может меняться в зависимости от модели нейронной сети. Так как на вход нейронной сети могут

подаваться разные данные. В данной программной реализации возможно подключение нескольких нейронных сетей для анализа данных.

Так же была создана база данных, ее даталогическая модель представлена на рисунке 2.7.

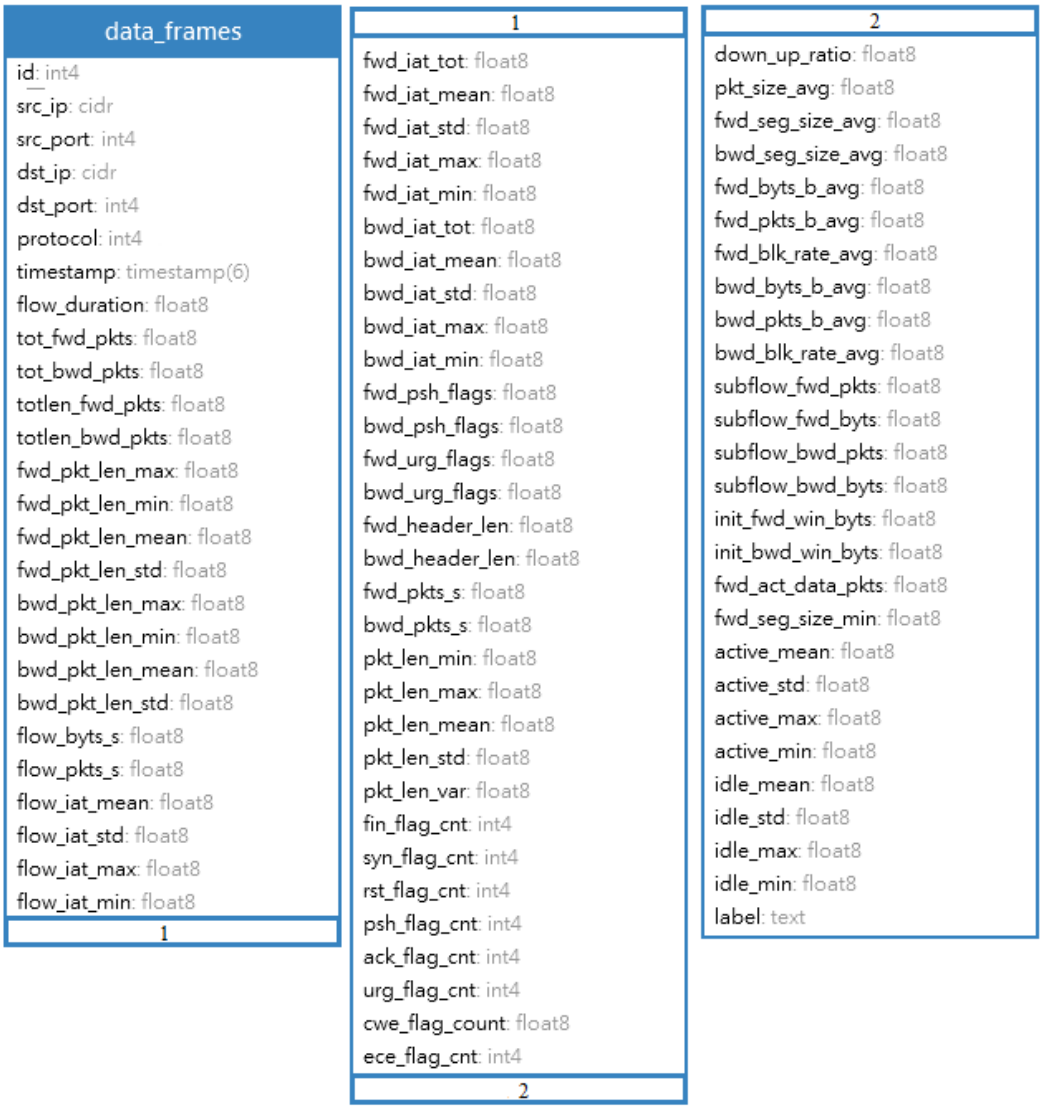


Рисунок 2.7 – Даталогическая модель базы данных

Полный листинг программы определен в Приложении Б.

2.4. Разработка веб-приложения для просмотра подозрительной активности

Веб-приложение не будет иметь модели авторизации. Так как управлене доступом будет обеспечено на уровне веб-сервера, который ограничит список

ip адресов с которых может быть получен доступ к веб-приложению. В качестве хранилища данных будет использован PostgreSQL сервер. Модель базы данных представлена на рисунке 2.7.

В соответсвии с потребностями были разработаны следующие экраны:

– главный экран со списком событий;

– экран с подброной информацией о событии.

Главная страница приложения представлена на рисунке 2.8. На ней отображены произошедшие инциденты, их номер, имя обнаружевшего модуля, ip:порт адрес источника, ip:порт адрес назначения и дата фиксации инцидента. Все инциденты отсортированы по дате. Так же есть возможность посмотреть более подробную информацию об инциденте, нажав на кнопку «Подробнее».

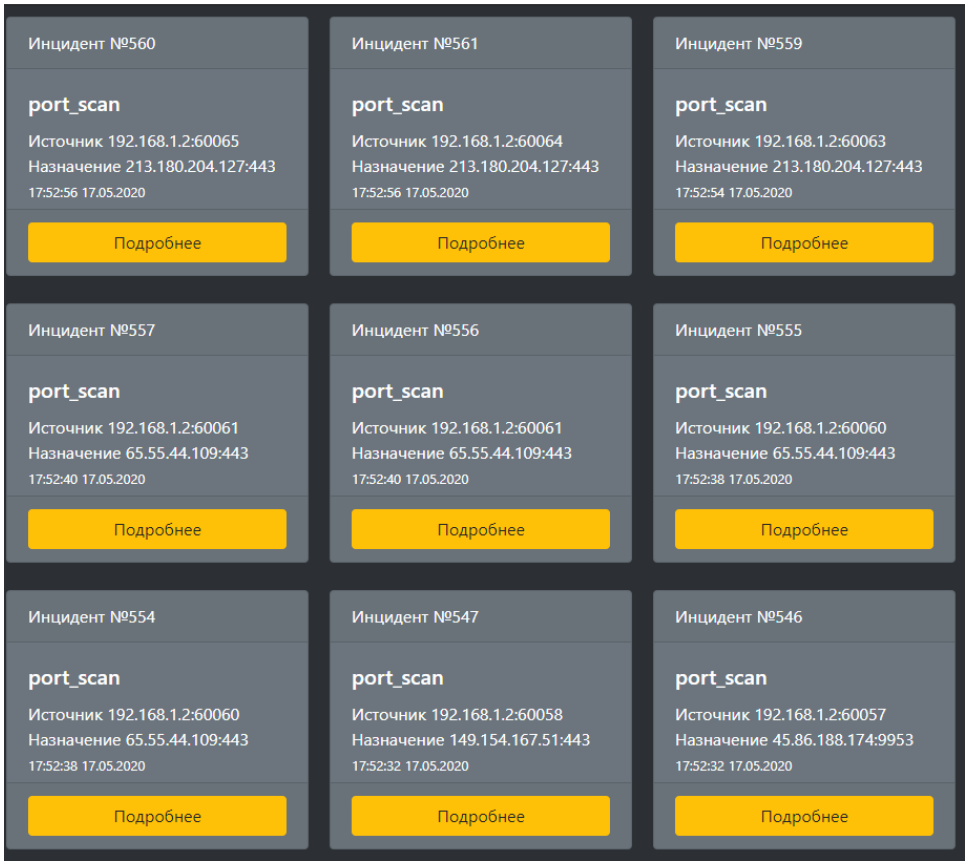
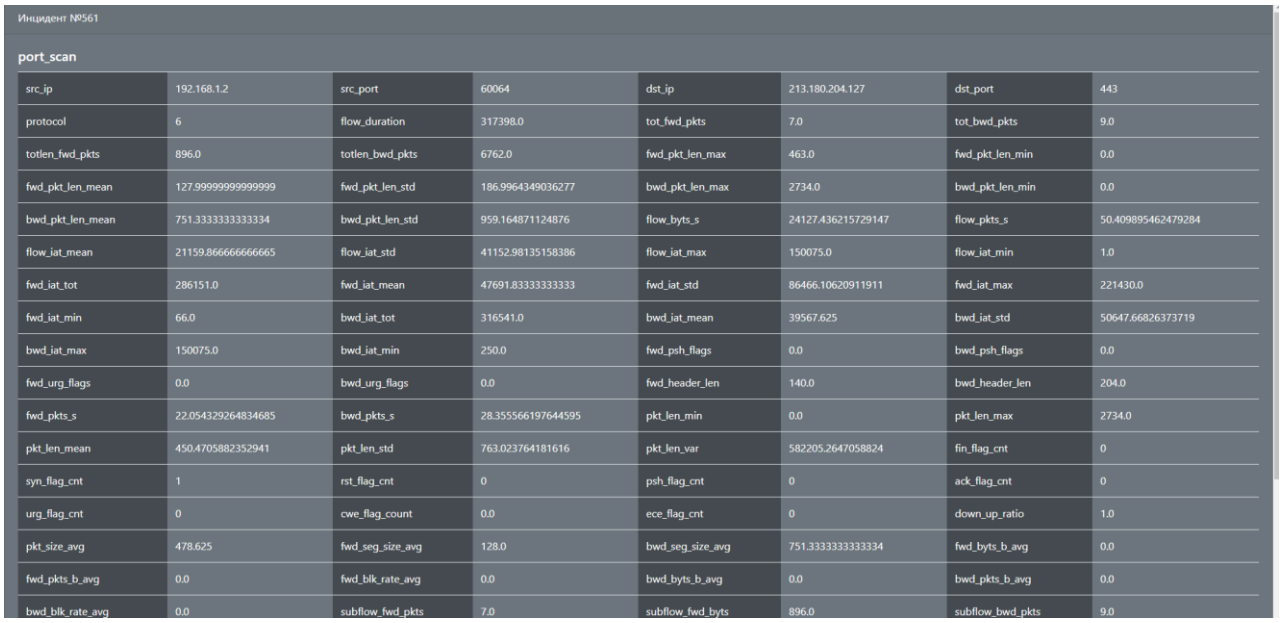


Рисунок 2.8 – Главная страница веб-приложения

Нажав на кнопку «подробнее» пользователь попадает на страницу с подробным описанием инцидента. Страница с подробным описанием

изображена на рисунке 2.9. Здесь в виде таблицы отражен полный список данных по данному инциденту.



Инцидент №561

port_scan

src_ip	192.168.1.2	src_port	60064	dst_ip	213.180.204.127	dst_port	443
protocol	6	flow_duration	317398.0	tot_fwd_pkts	7.0	tot_bwd_pkts	9.0
totlen_fwd_pkts	896.0	totlen_bwd_pkts	6762.0	fwd_pkt_len_max	463.0	fwd_pkt_len_min	0.0
fwd_pkt_len_mean	127.99999999999999	fwd_pkt_len_std	186.9964349036277	bwd_pkt_len_max	2734.0	bwd_pkt_len_min	0.0
bwd_pkt_len_mean	751.3333333333334	bwd_pkt_len_std	959.164871124876	flow_byts_s	24127.436215729147	flow_pkts_s	50.409895462479284
flow_iat_mean	21159.866666666665	flow_iat_std	41152.98135158386	flow_iat_max	150075.0	flow_iat_min	1.0
fwd_iat_tot	286151.0	fwd_iat_mean	47691.83333333333	fwd_iat_std	86466.10620911911	fwd_iat_max	221430.0
fwd_iat_min	66.0	bwd_iat_tot	316541.0	bwd_iat_mean	39567.625	bwd_iat_std	50647.66826373719
bwd_iat_max	150075.0	bwd_iat_min	250.0	fwd_psh_flags	0.0	bwd_psh_flags	0.0
fwd_urg_flags	0.0	bwd_urg_flags	0.0	fwd_header_len	140.0	bwd_header_len	204.0
fwd_pkts_s	22.054329264834685	bwd_pkts_s	28.355566197644595	pkt_len_min	0.0	pkt_len_max	2734.0
pkt_len_mean	450.4705882352941	pkt_len_std	763.023764181616	pkt_len_var	582205.2647058824	fin_flag_cnt	0
syn_flag_cnt	1	rst_flag_cnt	0	psh_flag_cnt	0	ack_flag_cnt	0
urg_flag_cnt	0	cwe_flag_count	0.0	ece_flag_cnt	0	down_up_ratio	1.0
pkt_size_avg	478.625	fwd_seg_size_avg	128.0	bwd_seg_size_avg	751.3333333333334	fwd_byts_b_avg	0.0
fwd_pkts_b_avg	0.0	fwd_blk_rate_avg	0.0	bwd_byts_b_avg	0.0	bwd_pkts_b_avg	0.0
bwd_blk_rate_avg	0.0	subflow_fwd_pkts	7.0	subflow_fwd_byts	896.0	subflow_bwd_pkts	9.0

Рисунок 2.9 – Подробное описание инцидента

Полный листинг программы определен в Приложении В.

2.5. Выводы

В главе произведено определение компонентов системы в ходе которого было уставлено какие компоненты необходимо разработать для системы обнаружения атак на сетевые ресурсы с применением нейросетевых технологий чтобы она могла работать со сборщиком трафика CICFlowMeter и при этом обеспечивала работу в реальном времени.

Произведена разработка сборщика данных. В ходе разработки было установлено какие данные предоставляет программа CICFlowMeter и в каком формате. Далее был разработан веб сервер на языке python, который считывает csv файл генерируемый программой CICFlowMeter, преобразовывает их в формат json и предоставляет их на специальной веб странице.

Произведена разработка анализирующего модуля. Анализирующий модуль собирает данные генерируемые сборщиком данных и «прогоняет» их

через модели нейронных сетей. Если данные помечаются как аномальные, они записываются в базу данных. В ходе разработки анализирующего модуля была разработана модель базы данных для хранения данных помеченных как аномальные.

Произведена разработка веб-приложения для просмотра данных, помеченных как аномальные. Приложение имеет всего 2 экрана, но при этом предоставляет полную информацию об аномальных данных.

3. Разработка и тестирование модели нейронной сети

3.1. Разработка модели для обнаружения DoS атак

Для начала разработки модели нужно определиться что представляет из себя модель нейронной сети. Модель нейронной сети описывает её архитектуру и конфигурацию, а также используемые алгоритмы обучения.

Архитектура нейронной сети определяет общие принципы её построения (плоскослоистая, полносвязная, слабосвязная, прямого распространения, рекуррентная и т.д.).

Конфигурация конкретизирует структуру сети в рамках заданной архитектуры: число нейронов, число входов и выходов сети, используемые активационные функции [19].

Исходя из этого необходимо выбрать архитектуру нейронной сети. Перед нейронной сетью стоит задача классификации трафика. Для задач классификации подходят следующие архитектуры нейронных сетей:

- перцептрон;
- сверточные нейронные сети;
- сети адаптивного резонанса;
- сеть радиально-базисных функций.

Так как данные, которые будут использоваться для обучения уже заранее размечены, то разумно будет выбирать между сверточными нейронными сетями и перцептроном, так как именно они подходят для обучения с учителем. Так как сверточные нейронные сети это специальная архитектура искусственных нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание образов[20], то для реализации модели нейронной сети для системы обнаружения атак на сетевые ресурсы будет выбрана архитектура перцептрона. Структурная схема перцептрона представлена на рисунке 3.1.

Далее приступим к определению конфигурации модели. Число входов и выходов нейронной сети будут зависеть от данных, используемых для обучения. Для дипломного проекта будет использована база данных атак CICIDS2017.

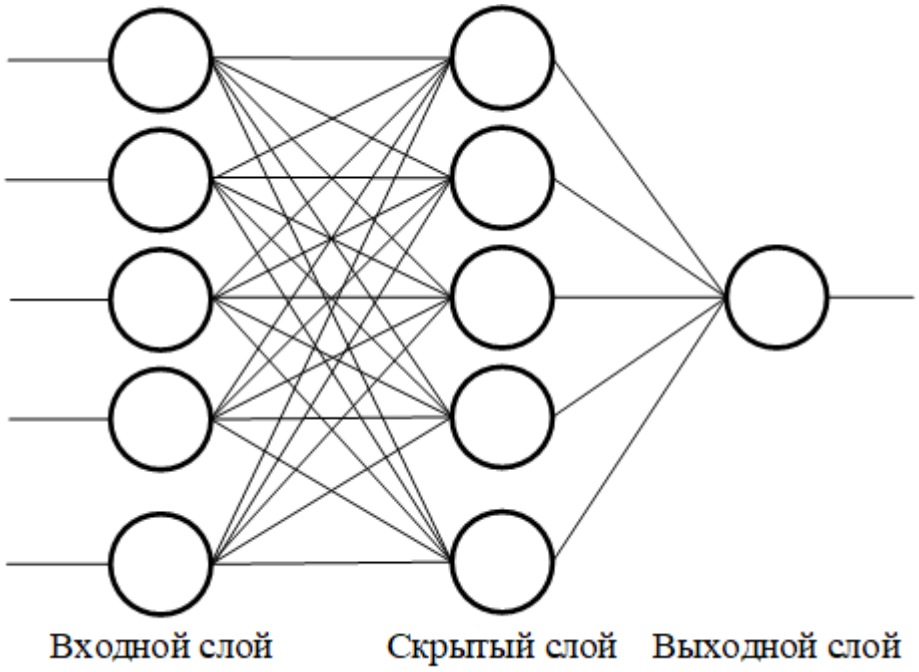


Рисунок 3.1 – Структурная схема перцептрона

Данные представляют собой csv файл. В котором содержится 77 параметров и итоговая оценка к этим параметрам. Является ли трафик аномальным или нормальным. Изначально данные имеют вид, представленный в таблице 3.1 в столбце «Пример данных до преобразования».

Такой формат для нейронной сети не подходит. Все данные должны иметь числовой формат. Поэтому они были приведены к виду, представленному в таблице 3.1 в столбце «Пример данных после преобразования

Таблица 3.1 – Параметры датасета до и после преобразования

№ п/п	Название параметра	Пример данных до преобразования	Пример данных после преобразования
1	2	3	4
1	Destination Port	54865	54865

Продолжение таблицы 3.1

№ п/п	Название параметра	Пример данных до преобразования	Пример данных после преобразования
1	2	3	4
2	Flow Duration	3	3
3	Total Fwd Packets	2	2
4	Total Backward Packets	0	0
5	Total Length of Fwd Packets	12	12
6	Total Length of Bwd Packets	0	0
7	Fwd Packet Length Max	6	6
8	Fwd Packet Length Min	6	6
9	Fwd Packet Length Mean	6	6
10	Fwd Packet Length Std	0	0
11	Bwd Packet Length Max	0	0
12	Bwd Packet Length Min	0	0
13	Bwd Packet Length Mean	0	0
14	Bwd Packet Length Std	0	0
15	Flow Bytes/s	4000000	4000000
16	Flow Packets/s	666666,7	666666,7
17	Flow IAT Mean	3	3
18	Flow IAT Std	0	0
19	Flow IAT Max	3	3
20	Flow IAT Min	3	3
21	Fwd IAT Total	3	3
22	Fwd IAT Mean	3	3
23	Fwd IAT Std	0	0
24	Fwd IAT Max	3	3
25	Fwd IAT Min	3	3
26	Bwd IAT Total	0	0
27	Bwd IAT Mean	0	0
28	Bwd IAT Std	0	0
29	Bwd IAT Max	0	0
30	Bwd IAT Min	0	0
31	Fwd PSH Flags	0	0
32	Bwd PSH Flags	0	0
33	Fwd URG Flags	0	0
34	Bwd URG Flags	0	0
35	Fwd Header Length	40	40
36	Bwd Header Length	0	0
37	Fwd Packets/s	666666,7	666666,7
38	Bwd Packets/s	0	0
39	Min Packet Length	6	6
40	Max Packet Length	6	6
41	Packet Length Mean	6	6
42	Packet Length Std	0	0
43	Packet Length Variance	0	0
44	FIN Flag Count	0	0
45	SYN Flag Count	0	0
46	RST Flag Count	0	0
47	PSH Flag Count	0	0

Продолжение таблицы 3.1

№ п/п	Название параметра	Пример данных до преобразования	Пример данных после преобразования
1	2	3	4
48	ACK Flag Count	1	1
49	URG Flag Count	0	0
50	CWE Flag Count	0	0
51	ECE Flag Count	0	0
52	Down/Up Ratio	0	0
53	Average Packet Size	9	9
54	Avg Fwd Segment Size	6	6
55	Avg Bwd Segment Size	0	0
56	Fwd Header Length	40	40
57	Fwd Avg Bytes/Bulk	0	0
58	Fwd Avg Packets/Bulk	0	0
59	Fwd Avg Bulk Rate	0	0
60	Bwd Avg Bytes/Bulk	0	0
61	Bwd Avg Packets/Bulk	0	0
62	Bwd Avg Bulk Rate	0	0
63	Subflow Fwd Packets	2	2
64	Subflow Fwd Bytes	12	12
65	Subflow Bwd Packets	0	0
66	Subflow Bwd Bytes	0	0
67	Init_Win_bytes_forward	33	33
68	Init_Win_bytes_backward	-1	-1
69	act_data_pkt_fwd	1	1
70	min_seg_size_forward	20	20
71	Active Mean	0	0
72	Active Std	0	0
73	Active Max	0	0
74	Active Min	0	0
75	Idle Mean	0	0
76	Idle Std	0	0
77	Idle Max	0	0
78	Idle Min	0	0
79	Label	BENIGN	0

Параметр оценки трафика получил номер, где 1 – аномальный трафик, 0 – нормальный. Все числа были приведены к вещественному типу с плавающей запятой.

Так для обучения необходимо 2 набора данных, тренировочный и тестовый, то из датасет будет поделен так, что 20% данных будет отдано под тесты, а 80% для обучения.

Для начала разработки модели необходимо определиться что такое DoS атака и чем она характеризуется. Датасет содержит 78 параметров их полное

использование является нецелесообразным, так как некоторые параметры во время атак остаются неизменными. А большое количество входных параметров существенно затруднит создание модели.

DoS (Denial of Service «отказ в обслуживании») – хакерская атака на вычислительную систему с целью довести её до отказа, то есть создание таких условий, при которых добросовестные пользователи системы не смогут получить доступ к предоставляемым системным ресурсам (серверам), либо этот доступ будет затруднён [21]. Исходя из характера воздействия DoS атаки можно отнести к активным атакам.

Для DoS атак характерно большое количество пакетов, которые атакуемому необходимо обработать, но из-за того, что вычислительные мощности ограничены возникают задержки, в последствии переходящие в отказ в обслуживании. Исходя из этого из этого из таблицы 1.1 были выбраны следующие параметры: Total Fwd Packets, Total Backward Packets, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std, Bwd Packet Length Min, Fwd Header Length, Bwd Header Length, Min Packet Length, Packet Length Mean, PSH Flag Count, ACK Flag Count, URG Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Subflow Fwd Packets, Subflow Bwd Packets, act_data_pkt_fwd, min_seg_size_forward, Label.

Таким образом модель будет иметь 20 нейронов на входе и 2 на выходе. Выходные нейроны будут определены как Normal и DoS, а их выходные значения будут показывать в какой степени нейронная сеть склоняется к тому или иному ответу.

В качестве активационной функции во всех слоях кроме выходного будет выступать функция relu, на выходном слое будет функция softmax. Выбор relu обусловлен тем, что она является наиболее популярной и наиболее используемой в создании нейронных сетей. Выбор функции softmax на выходном слое обусловлен видом выходных данных.

Для того чтобы избежать переобучения нейронной сети в библиотеке Keras предусмотрена функций ранней остановки. Ранняя остановка – это метод, который позволяет указать произвольно большое количество эпох обучения и прекратить обучение, как только производительность модели перестает улучшаться.

Так как количество эпох обучения будет контролироваться ранней остановкой, то остается подобрать количество скрытых слоев и количество нейронов в скрытых слоях.

Найдем оптимальное количество скрытых слоев. Количество нейронов в каждом скрытом слое будет ровняться 20. Оценка качества обучения будет проводиться по двум параметрам. Точность и потери, точность должна стремиться к 1, а потери к 0. Результаты подбора представлены в таблице 3.2.

Таблица 3.2 – Подбор оптимального количества скрытых слоев

№ п/п	Количество скрытых слоев	Потери	Точность
1	2	3	4
1	1	5,874060	0,634876
2	2	0,009109	0,998627
3	3	0,008066	0,998760
4	4	0,008952	0,998582
5	5	0,004545	0,998782
6	6	0,004774	0,998826
7	7	0,008644	0,998671
8	8	0,012150	0,998516
9	9	0,004999	0,998870
10	10	0,005463	0,998760
11	11	0,012202	0,998538
12	12	0,015628	0,997807
13	13	0,009046	0,998671
14	14	0,005136	0,998893
15	15	0,400654	0,793927
16	16	0,549039	0,634743
17	17	0,683352	0,569935
18	18	0,683342	0,569935
19	19	0,683556	0,569935
20	20	0,683431	0,569935
21	21	0,683355	0,569935
22	22	0,683346	0,569935
23	23	0,683348	0,569935

Продолжение таблицы 3.2

№ п/п	Количество скрытых слоев	Потери	Точность
1	2	3	4
24	24	0,683530	0,569935
25	25	0,683574	0,569935

Результаты приставлены на рисунке 3.2. По итогам тестирования оптимальным количеством скрытых слоев является 6, так как в таком случае получается наибольшая точность и наименьшие потери.

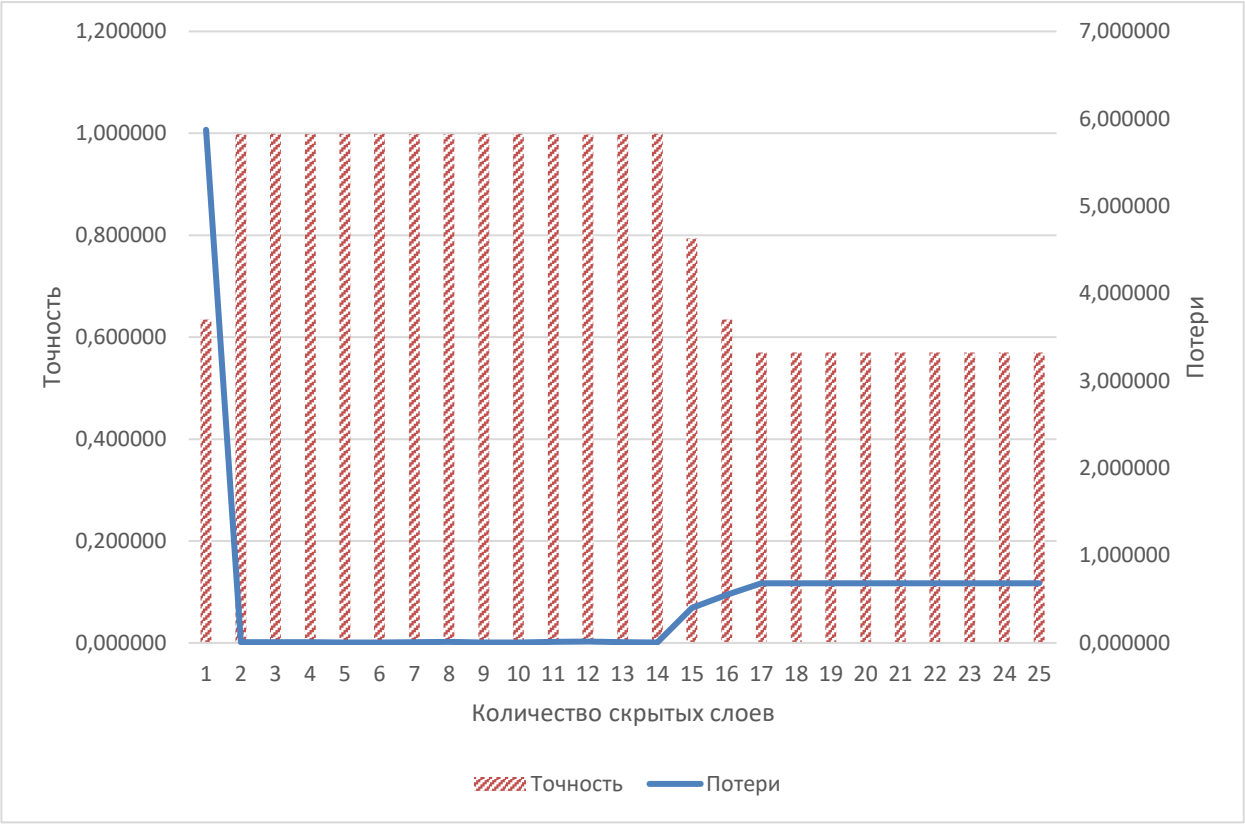


Рисунок 3.2 – Результаты тестирования оптимального количество слоев

Далее найдем оптимальное количество нейронов в скрытых слоях. Количество скрытых слоев будет равняться 6. Результаты подбора представлены в таблице 3.3.

Таблица 3.3 – Подбор оптимального количества нейронов в скрытых слоях

№ п/п	Количество нейронов в скрытых слоях	Потери	Точность
1	2	3	4
1	1	0,68335	0,56994
2	2	0,68346	0,56994
3	3	0,02298	0,99772
4	4	0,00729	0,99863
5	5	0,01075	0,99849
6	6	0,00679	0,99883
7	7	0,00722	0,99854
8	8	0,00806	0,99865
9	9	0,00736	0,99867
10	10	0,01158	0,99858
11	11	0,01019	0,99860
12	12	0,00678	0,99865
13	13	0,01083	0,99856
14	14	0,00699	0,99865
15	15	0,00533	0,99865
16	16	0,01168	0,99847
17	17	0,00521	0,99876
18	18	0,00540	0,99883
19	19	0,00709	0,99874
20	20	0,00946	0,99825
21	21	0,00693	0,99883
22	22	0,00572	0,99874
23	23	0,00922	0,99807
24	24	0,00543	0,99894
25	25	0,00979	0,99852
26	26	0,00684	0,99874
27	27	0,00492	0,99883
28	28	0,01066	0,99816
29	29	0,00844	0,99867
30	30	0,00946	0,99872
31	31	0,00526	0,99887
32	32	0,00453	0,99885
33	33	0,00762	0,99856
34	34	0,01094	0,99856
35	35	0,00687	0,99874
36	36	0,00903	0,99872
37	37	0,00975	0,99863
38	38	0,00845	0,99863
39	39	0,00879	0,99869
40	40	0,00870	0,99869
41	41	5,87296	0,63499
42	42	0,00456	0,99887

Продолжение таблицы 3.3

43	43	0,00834	0,99876
44	44	0,00911	0,99869
45	45	0,00856	0,99872
46	46	0,00472	0,99885
47	47	0,00528	0,99869
48	48	0,00892	0,99867
49	49	0,01034	0,99858
50	50	5,87522	0,63470

Результаты приставлены на рисунке 3.3. По итогам тестирования оптимальным количеством нейронов в скрытых слоях является 42, так как в таком случае получается наибольшая точность и наименьшие потери.

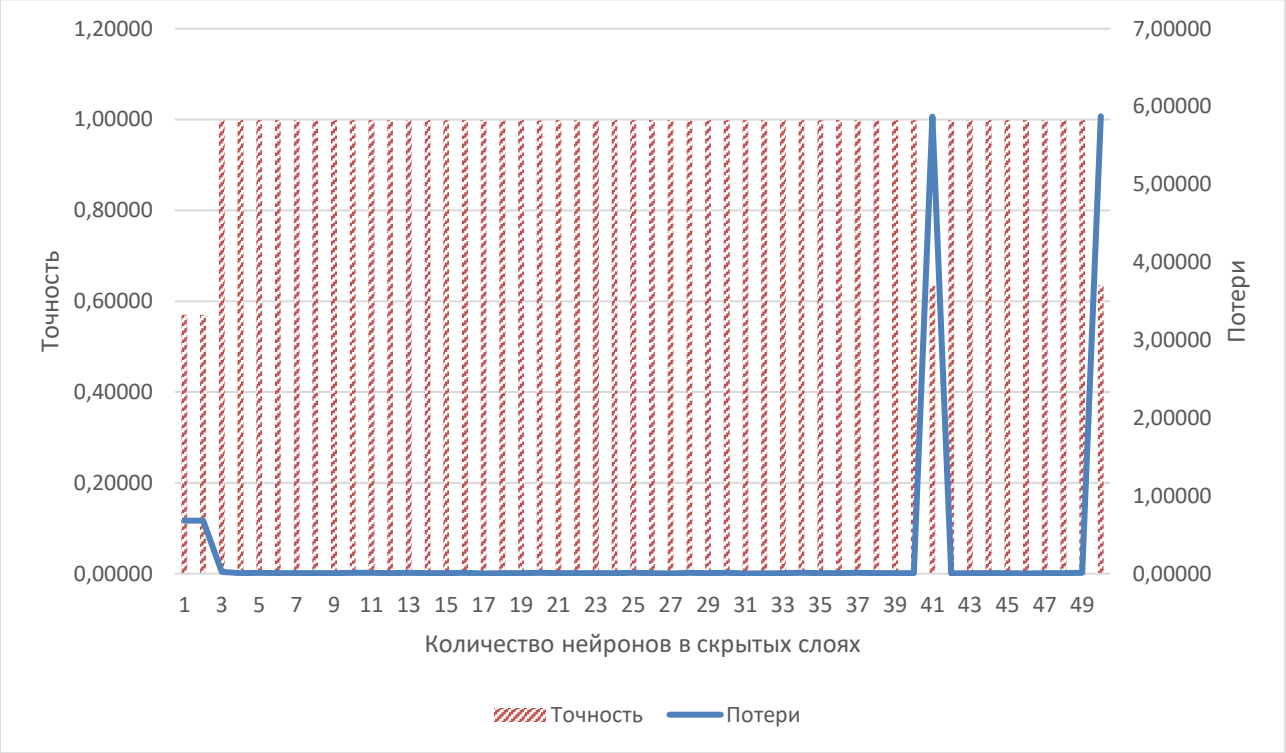


Рисунок 3.3 – Результаты тестирования оптимального количества нейронов в скрытых слоях

Таким образом модель будет иметь 6 скрытых слоев и в каждом скрытом слое будет находиться по 42 нейрона. Точность модели достигла 99,88%. Разработанная модель сохранена в файл dos.h5 и может быть подключена к ранее разработанному анализирующему модулю.

3.2.Разработка модели для обнаружения PortScan атак

Так же как и в предыдущей главе определим что такое PortScan атака и чем она характеризуется. PortScan это процесс, который отправляет запросы клиента к диапазону портов сервера адресов на хосте, с целью нахождения активного порта [22]. Исходя из характера воздействия PortScan атаки можно отнести к пассивным атакам.

Для PortScan атак характерное малое количество небольших пакетов. Так же соединение, которое сканнер устанавливает с атакуемым устройством длится очень малое время. Исходя из этого из этого из таблицы 1.1 были выбраны следующие параметры: Destination Port, Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std, Bwd Packet Length Max, Bwd Packet Length Min, Bwd Packet Length Mean, Bwd Packet Length Std, Fwd Header Length, Bwd Header Length, Fwd Packets/s, Bwd Packets/s, Min Packet Length, Max Packet Length, Packet Length Mean, Packet Length Std, Packet Length Variance, FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Avg Bwd Segment Size, Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes, Init_Win_bytes_forward, Init_Win_bytes_backward, act_data_pkt_fwd, min_seg_size_forward, Label.

Таким образом модель будет иметь 38 нейронов на входе и 2 на выходе. Выходные нейроны будут определены как Normal и PortScan, а их выходные значения будут показывать в какой степени нейронная сеть склоняется к тому или иному ответу.

В качестве активационной функции во всех слоях кроме выходного будет выступать функция relu, на выходном слое будет функция softmax. Выбор relu обусловлен тем, что она является наиболее популярной и наиболее

используемой в создании нейронных сетей. Выбор функции softmax на выходном слое обусловлен видом выходных данных.

Для того чтобы избежать переобучения нейронной сети в библиотеке Keras предусмотрена функций ранней остановки.

Так как количество эпох обучения будет контролироваться ранней остановкой, то остается подобрать количество скрытых слоев и количество нейронов в скрытых слоях.

Найдем оптимальное количество скрытых слоев. Количество нейронов в каждом скрытом слое будет ровняться 38. Оценка качества обучения будет проводиться по двум параметрам. Точность и потери, точность должна стремиться к 1, а потери к 0. Результаты подбора представлены в таблице 3.4.

Таблица 3.4 – Подбор оптимального количества скрытых слоев

№ п/п	Количество скрытых слоев	Потери	Точность
1	2	3	4
1	1	0,77454	0,951914686
2	2	0,66013	0,959000943
3	3	8,93537	0,445631305
4	4	0,08174	0,993786435
5	5	0,08399	0,992861382
6	6	0,05737	0,995200195
7	7	0,0618	0,993297728
8	8	0,02638	0,997521555
9	9	0,05061	0,990138583
10	10	0,03054	0,99581108
11	11	0,02781	0,997853178
12	12	0,0296	0,996736133
13	13	0,04935	0,993210458
14	14	0,02609	0,997818271
15	15	0,03108	0,997050302
16	16	0,03206	0,994798757
17	17	0,02788	0,99745174
18	18	0,03445	0,994886026
19	19	0,02999	0,996317241
20	20	0,03727	0,994973296
21	21	0,03036	0,99640451
22	22	0,03096	0,996212518
23	23	0,03043	0,996666318
24	24	0,68726	0,554368695

Продолжение таблицы 3.4

№ п/п	Количество скрытых слоев	Потери	Точность
1	2	3	4
25	25	0,68733	0,554368695
26	26	0,68726	0,554368695
27	27	0,68722	0,554368695
28	28	0,68729	0,554368695
29	29	0,68723	0,554368695
30	30	0,68724	0,554368695

Результаты приставлены на рисунке 3.4. По итогам тестирования оптимальным количеством скрытых слоев является 14, так как в таком случае получается наибольшая точность и наименьшие потери.

Далее найдем оптимальное количество нейронов в скрытых слоях. Количество скрытых слоев будет равняться 14. Результаты подбора представлены в таблице 3.5.

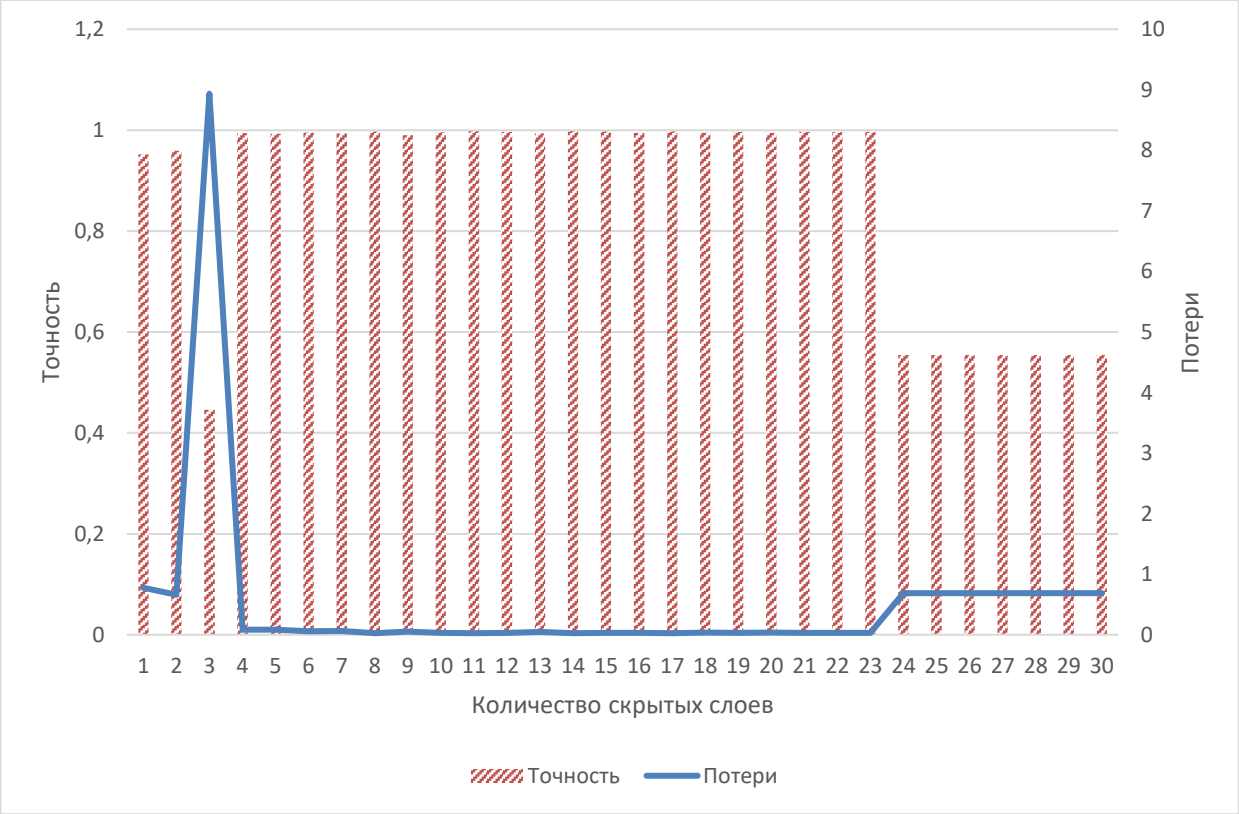


Рисунок 3.4 – Результаты тестирования оптимального количество слоев

Таблица 3.5 – Подбор оптимального количества нейронов в скрытых слоях

№ п/п	Количество нейронов в скрытых слоях	Потери	Точность
1	2	3	4
1	1	0,68724	0,554368695
2	2	0,68722	0,554368695
3	3	0,68725	0,554368695
4	4	0,68723	0,554368695
5	5	0,68722	0,554368695
6	6	0,68726	0,554368695
7	7	0,68722	0,554368695
8	8	0,03793	0,992250497
9	9	0,68723	0,554368695
10	10	0,02732	0,99708521
11	11	0,03046	0,996055433
12	12	0,02655	0,997032848
13	13	0,02669	0,996910671
14	14	0,04446	0,992669389
15	15	0,03115	0,996352149
16	16	0,03293	0,996020526
17	17	0,03476	0,996334695
18	18	0,03157	0,996910671
19	19	0,03471	0,995601634
20	20	0,05284	0,988218662
21	21	0,06192	0,99408315
22	22	0,02651	0,996928125
23	23	0,03173	0,995741264
24	24	0,02888	0,996648864
25	25	0,03809	0,994065696
26	26	0,0799	0,993402451
27	27	0,05415	0,996090341
28	28	0,02688	0,997591371
29	29	0,05823	0,984972248
30	30	0,03932	0,99513038
31	31	0,02987	0,996247426
32	32	0,02874	0,997347017
33	33	0,02867	0,996928125
34	34	0,05028	0,990958914
35	35	0,05611	0,994938388
36	36	0,02964	0,99663141
37	37	0,0324	0,99581108
38	38	0,03413	0,99535728
39	39	0,03951	0,996160156
40	40	0,05177	0,996299787
41	41	0,0818	0,993001012
42	42	0,02605	0,997469194

Продолжение таблицы 3.5

№ п/п	Количество нейронов в скрытых слоях	Потери	Точность
1	2	3	4
43	43	0,04115	0,99431005
44	44	0,06058	0,989841868
45	45	0,02857	0,997032848
46	46	0,05755	0,994920934
47	47	0,02811	0,997626278
48	48	0,05754	0,989684784
49	49	0,03143	0,997259748
50	50	0,03291	0,996090341
51	51	0,04797	0,996195064
52	52	0,08009	0,993524627
53	53	0,02998	0,997155025
54	54	0,05547	0,995549272
55	55	0,07225	0,992529759
56	56	7,18273	0,554368695
57	57	0,03767	0,993908612
58	58	0,05888	0,995235103
59	59	7,18266	0,554368695
60	60	0,03585	0,995723811
61	61	0,03558	0,993018466
62	62	0,04722	0,993053374
63	63	0,06523	0,992896289
64	64	0,03649	0,994275142
65	65	0,02938	0,997294656
66	66	0,08681	0,993594443
67	67	0,08031	0,993873704
68	68	0,03433	0,995514365
69	69	0,08241	0,993891158
70	70	0,04659	0,992093413
71	71	0,17195	0,968809998
72	72	0,04732	0,99221559
73	73	0,13207	0,970572835
74	74	0,07791	0,993891158
75	75	0,06514	0,993786435
76	76	0,10259	0,991447621
77	77	0,06179	0,989283346
78	78	0,08095	0,99371662
79	79	0,02548	0,99790554
80	80	0,05432	0,995688903
81	81	0,062	0,995741264
82	82	0,0796	0,994048242
83	83	0,05671	0,995392188
84	84	0,03129	0,995915803
85	85	0,05911	0,994624219
86	86	0,03058	0,996613956
87	87	0,05476	0,996299787

Продолжение таблицы 3.5

№ п/п	Количество нейронов в скрытых слоях	Потери	Точность
1	2	3	4
88	88	0,13811	0,989265892
89	89	7,18273	0,554368695
90	90	7,18273	0,554368695
91	91	0,03707	0,995741264
92	92	7,18273	0,554368695
93	93	0,08534	0,992651936
94	94	0,05691	0,995304918
95	95	0,0513	0,99603798
96	96	0,03068	0,996020526
97	97	0,05749	0,994990749
98	98	0,03293	0,996701225
99	99	0,03494	0,995619088
100	100	0,03573	0,99581108

Результаты представлены на рисунке 3.5. По итогам тестирования оптимальным количеством нейронов в скрытых слоях является 79, так как в таком случае получается наибольшая точность и наименьшие потери.

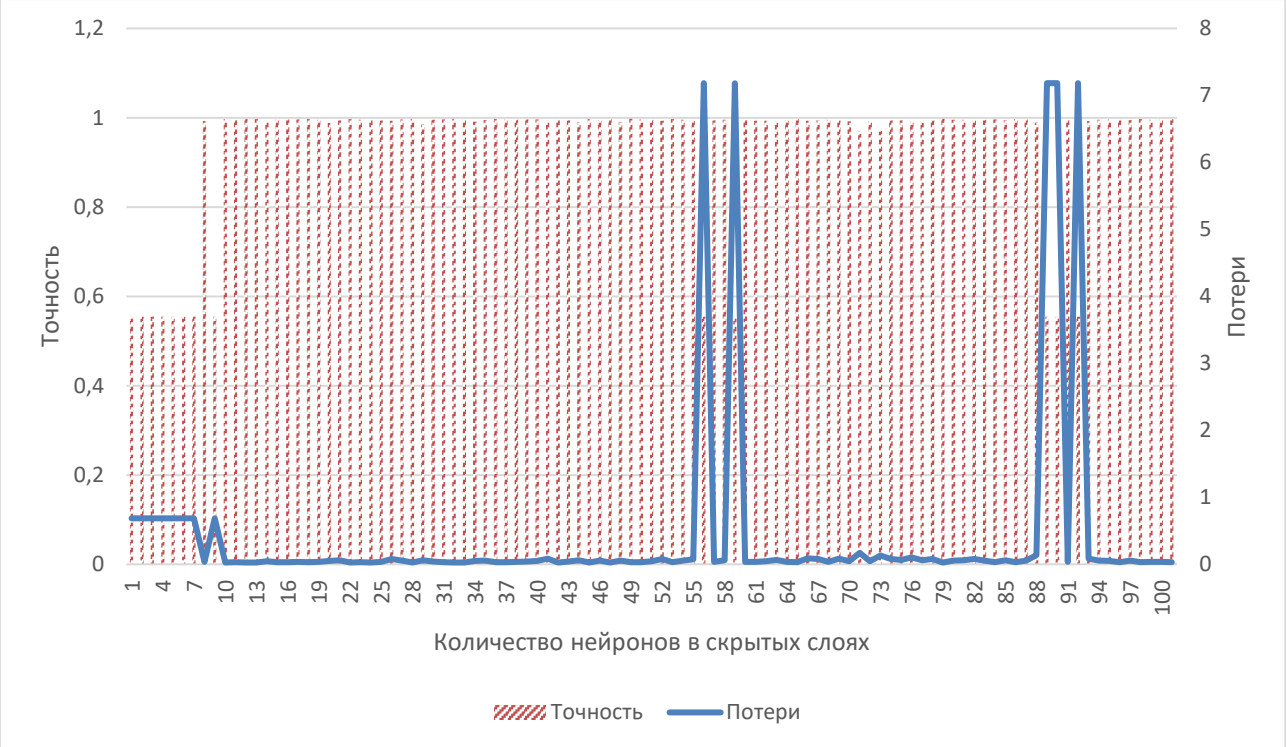


Рисунок 3.5 – Результаты тестирования оптимального количества нейронов в скрытых слоях

Таким образом модель будет иметь 14 скрытых слоев и в каждом скрытом слое будет находиться по 79 нейронов. Точность модели достигла 99,79%. Разработанная модель сохранена в файл portscan.h5 и может быть подключена к ранее разработанному анализирующему модулю.

3.3. Тестирование разработанных моделей

После того как были разработаны все компоненты системы их можно объединить вместе и протестировать. Тестирование будет проведено на записи трафика, проложенному к датасету. К этой записи приложено описание, в какие часы и какие атаки проводились. Благодаря этому описанию можно будет оценить работоспособность системы. Согласно описанию проводились следующие атаки:

– Port Scan. С включенным файрволлом атаки проводились в: 13:55 – 13:57, 13:58 – 14:00, 14:01 – 14:04, 14:05 – 14:07, 14:08 - 14:10, 14:11 – 14:13, 14:14 – 14:16, 14:17 – 14:19, 14:20 – 14:21, 14:22 – 14:24, 14:33 – 14:33, 14:35-14:35. С выключенным файрволлом атаки проводились в: 14:51 – 14:53, 14:54 – 14:56, 14:57 – 14:59, 15:00 – 15:02, 15:03 – 15:05, 15:06 – 15:07, 15:08 – 15:10, 15:11 – 15:12, 15:13 – 15:15, 15:16 – 15:18, 15:19 – 15:21, 15:22 – 15:24, 15:25 – 15:25, 15:26 – 15:27, 15:28 – 15:29. Атака происходила с IP 205.174.165.73 на IP 172.16.0.1, между атакующим и атакуемым был файрволл с IP 205.174.165.80.

– DDoS LOIT атака проводилась с 15:56 до 16:16. Атака происходила с IP 205.174.165.69, 70, 71 на IP 172.16.0.1, между атакующим и атакуемым был файрволл с IP 205.174.165.80.

Перед началом тестирования необходимо немного изменить тестирующий модуль. Нужно изменить данные, которые будут подаваться на вход модели нейронной сети, на те же что и использовались при обучении.

Запустим модули системы:

1) база данных;
2) веб-приложение;

3) собирающий модуль. После запуска он будет ожидать появления в указанной папке csv файла. После его появления он начнет его считывать, а считанные данные будут поставлены в очередь для анализа;

4) CICFlowMeter. После запуска необходимо выбрать расположение файла с записанным трафиком, путь сохранения выходного csv файла и нажать кнопку «ОК». Путь сохранения должен совпадать с путем указанным в собирающем модуле. Интерфейс CICFlowMeter с запущенным чтением трафика представлен на рисунке 3.6;

5) анализирующий модуль. После запуска начнется анализ данных, вся найденная подозрительная активность будет записана в базу данных и ее можно будет просмотреть через веб-приложение.

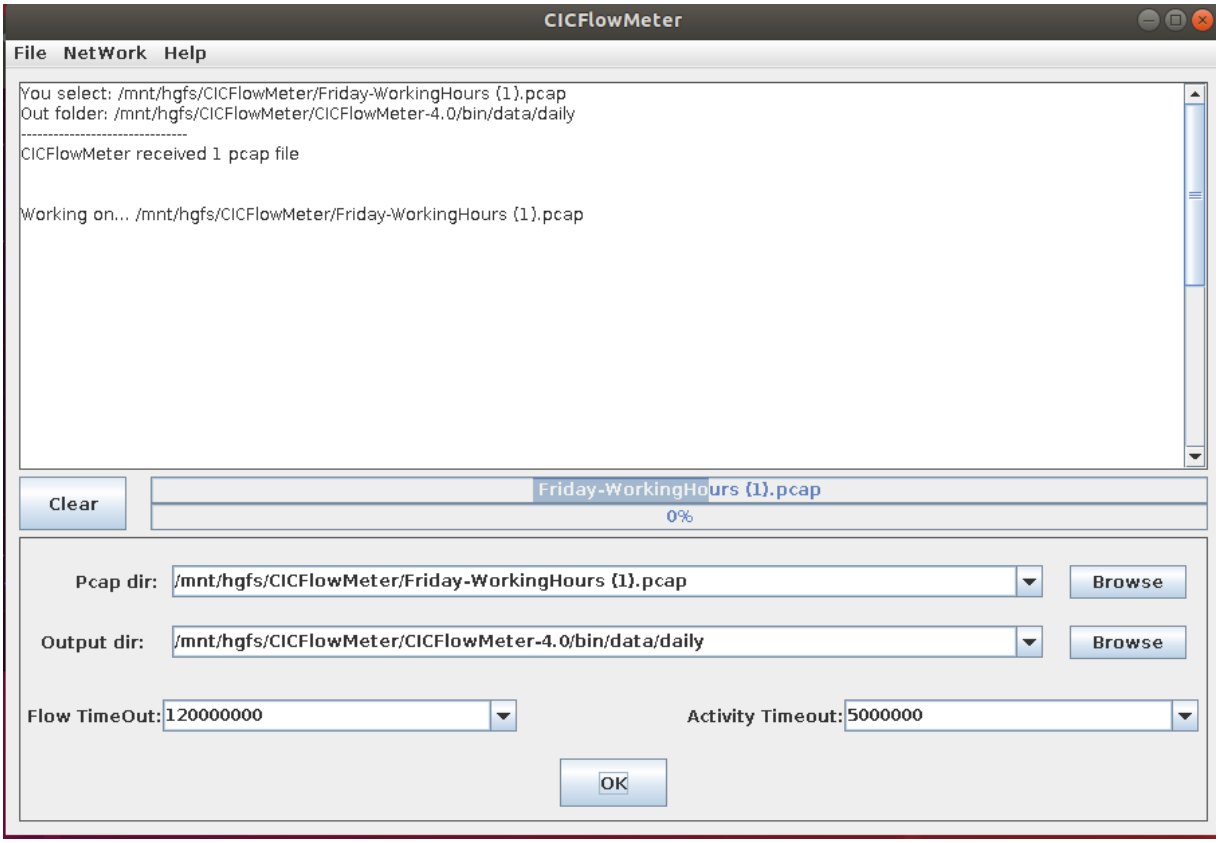


Рисунок 3.6 – Интерфейс программы CICFlowMeter с запущенным чтением трафика из файла

В ходе чтения трафика из файла система обнаружила 26 инцидентов. Отображение помеченных атак в веб-приложении представлено на рисунке 3.7.

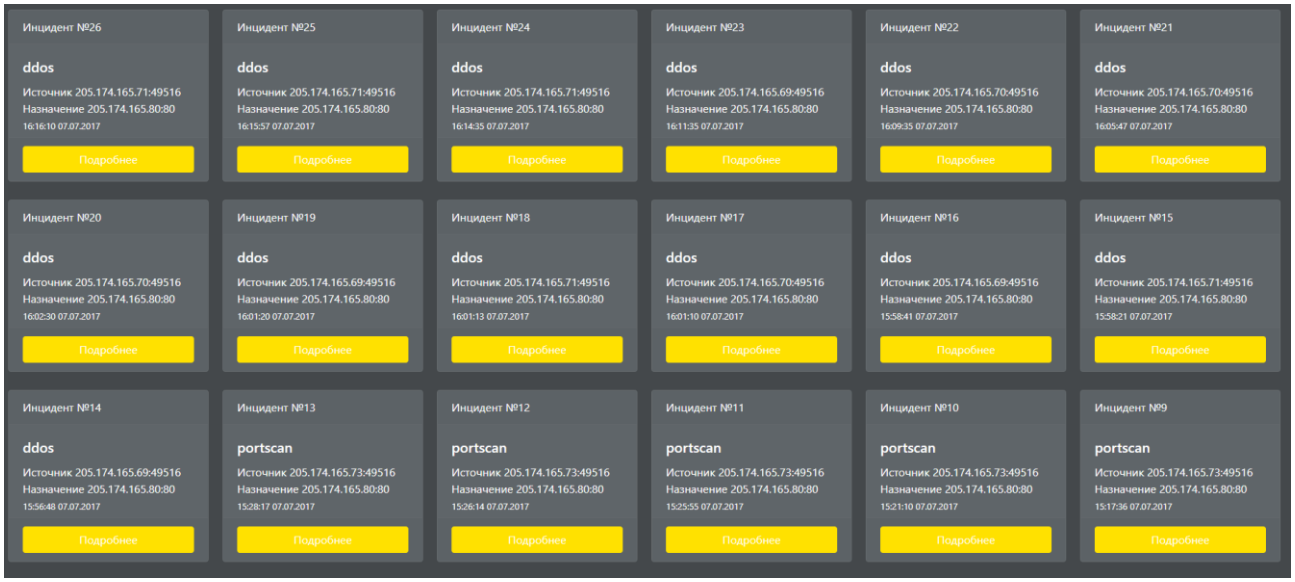


Рисунок 3.7 – Интерфейс веб-приложения с помеченными атаками

Из 26 инцидентов 13 были помечены как DDoS, а 13 как PortScan. DDoS инциденты помечались на всем временном промежутке, указанном в описании к записи трафика, при включенном файрволле не было распознано ни одной PortScan атаки, при выключенном файрволле была пропущена 1 PortScan атака, остальные были помечены как инциденты и сохранены в базе данных.

3.4. Выводы

В главе произведена разработка двух моделей нейронных сетей.

Разработка включала себя:

– выбор архитектуры сети – была выбрана архитектура перцептрона;

– оптимизацию данных – параметр оценки трафика получил номер, где 1 – аномальный трафик, 0 – нормальный. Все числа были приведены к вещественному типу с плавающей запятой. Также данные были поделены на тестовый и тренировочный набор. Тестовый набор данных составил 20% от общего количества данных;

– подбор параметров для достижения наименьших потерь и наибольшей точности.

Получившиеся модели получили следующие параметры:

– модель для обнаружения DoS атак – 20 нейронов на входе, 2 на выходе, 6 скрытых слоев, в каждом скрытом слое по 42 нейрона, точность модели 99,88%;

– модель для обнаружения PortScan атак – 38 нейронов на входе, 2 на выходе, 14 скрытых слоев, в каждом слое по 79 нейронов, точность модели достигла 99,79%.

Так же в главе проведено общее тестирование системы в ходе которого система обнаружила 26 инцидентов, 13 из которых были помечены как DDoS, а 13 как PortScan. DDoS инциденты помечались на всем временном промежутке, указанном в описании к записи трафика, при включенном файрволле не было распознано ни одной PortScan атаки, при выключенном файрволле была пропущена 1 PortScan атака, остальные были помечены как инциденты и сохранены в базе данных.

4. Безопасность и экологичность проекта

4.1. Требования к производственным помещениям

Научно-технический прогресс внес серьезные изменения в условия производственной деятельности работников умственного труда. Их труд стал более интенсивным, напряженным, требующим значительных затрат умственной, эмоциональной и физической энергии. Это потребовало комплексного решения проблем эргономики, гигиены и организации труда, регламентации режимов труда и отдыха.

В настоящее время компьютерная техника широко применяется во всех областях деятельности человека. При работе с компьютером человек подвергается воздействию ряда опасных и вредных производственных факторов, например, электромагнитных полей (диапазон радиочастот: ВЧ, УВЧ и СВЧ), инфракрасного и ионизирующего излучений, шума и вибрации, статического электричества и др.

Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ЭВМ. В процессе работы с компьютером необходимо соблюдать правильный режим труда и отдыха. Большое значение имеет рациональная конструкция и расположение элементов рабочего места, что важно для поддержания оптимальной рабочей позы человека-оператора.

Правильно спроектированное и выполненное производственное освещение улучшает условия зрительной работы, снижает утомляемость, способствует повышению производительности труда, благотворно влияет на производственную среду, оказывая положительное психологическое воздействие на работающего, повышает безопасность труда и снижает травматизм.

Недостаточность освещения приводит к напряжению зрения, ослабляет внимание, приводит к наступлению преждевременной утомленности. Чрезмерно яркое освещение вызывает ослепление, раздражение и резь в глазах. Неправильное направление света на рабочем месте может создавать резкие тени, блики, дезориентировать работающего. Все эти причины могут привести к несчастному случаю или профзаболеваниям, поэтому столь важен правильный расчет освещенности.

Существует три вида освещения – естественное, искусственное и совмещенное (естественное и искусственное вместе).

Естественное освещение – освещение помещений дневным светом, проникающим через световые проемы в наружных ограждающих конструкциях помещений. Естественное освещение характеризуется тем, что меняется в широких пределах в зависимости от времени дня, времени года, характера области и ряда других факторов.

Искусственное освещение применяется при работе в темное время суток и днем, когда не удается обеспечить нормированные значения коэффициента естественного освещения (пасмурная погода, короткий световой день). Освещение, при котором недостаточное по нормам естественное освещение дополняется искусственным, называется совмещенным освещением. Искусственное освещение подразделяется на рабочее, аварийное, эвакуационное, охранное. Рабочее освещение, в свою очередь, может быть общим или комбинированным. Общее – освещение, при котором светильники размещаются в верхней зоне помещения равномерно или применительно к расположению оборудования. Комбинированное – освещение, при котором к общему добавляется местное освещение. Согласно СНиП 23-05-95 в помещениях вычислительных центров необходимо применить систему комбинированного освещения.

При выполнении работ категории высокой зрительной точности (наименьший размер объекта различения 0,3-0,5мм) величина коэффициента

естественного освещения (КЕО) должна быть не ниже 1,5%, а при зрительной работе средней точности (наименьший размер объекта различения 0,5.1,0 мм) КЕО должен быть не ниже 1,0%. В качестве источников искусственного освещения обычно используются люминесцентные лампы типа ЛБ или ДРЛ, которые попарно объединяются в светильники, которые должны располагаться над рабочими поверхностями равномерно.

Требования к освещенности в помещениях, где установлены компьютеры, следующие: при выполнении зрительных работ высокой точности общая освещенность должна составлять 300лк, а комбинированная – 750лк; аналогичные требования при выполнении работ средней точности – 200 и 300лк соответственно. Кроме того все поле зрения должно быть освещено достаточно равномерно – это основное гигиеническое требование. Иными словами, степень освещения помещения и яркость экрана компьютера должны быть примерно одинаковыми, т.к. яркий свет в районе периферийного зрения значительно увеличивает напряженность глаз и, как следствие, приводит к их быстрой утомляемости.

Параметры микроклимата могут меняться в широких пределах, в то время как необходимым условием жизнедеятельности человека является поддержание постоянства температуры тела благодаря терморегуляции, т.е. способности организма регулировать отдачу тепла в окружающую среду.

Принцип нормирования микроклимата – создание оптимальных условий для теплообмена тела человека с окружающей средой. Вычислительная техника является источником существенных тепловыделений, что может привести к повышению температуры и снижению относительной влажности в помещении. В помещениях, где установлены компьютеры, должны соблюдаться определенные параметры микроклимата. В санитарных нормах СНиП 2.04.05-91 установлены величины параметров микроклимата, создающие комфортные условия. Эти нормы устанавливаются в зависимости от времени года, характера трудового процесса и характера производственного помещения.

Параметры микроклимата для помещений, где установлены компьютеры показаны в таблице 4.1.

Объем помещений, в которых размещены работники вычислительных центров, не должен быть меньше 19,5м^3/человека с учетом максимального числа одновременно работающих в смену.

Таблица 4.1 – Параметры микроклимата для помещений, где установлены компьютеры

Период года	Параметр микроклимата	Величина
Холодный	Температура воздуха в помещении	22.24°C
	Относительная влажность	40.60%
	Скорость движения воздуха	до 0,1м/с
Теплый	Температура воздуха в помещении	23.25°C
	Относительная влажность	40.60%
	Скорость движения воздуха	0,1.0,2м/с

Для обеспечения комфортных условий используются как организационные методы (рациональная организация проведения работ в зависимости от времени года и суток, чередование труда и отдыха), так и технические средства (вентиляция, кондиционирование воздуха, отопительная система).

Шум ухудшает условия труда, оказывая вредное действие на организм человека.

Работающие в условиях длительного шумового воздействия испытывают раздражительность, головные боли, головокружение, снижение памяти, повышенную утомляемость, понижение аппетита, боли в ушах и т. д. Такие нарушения в работе ряда органов и систем организма человека могут

вызвать негативные изменения в эмоциональном состоянии человека вплоть до стрессовых.

Под воздействием шума снижается концентрация внимания, нарушаются физиологические функции, появляется усталость в связи с повышенными энергетическими затратами и нервно-психическим напряжением, ухудшается речевая коммутация. Все это снижает работоспособность человека и его производительность, качество и безопасность труда. Длительное воздействие интенсивного шума [выше 80 дБ(А)] на слух человека приводит к его частичной или полной потере. В таблице 4.2 указаны предельные уровни звука в зависимости от категории тяжести и напряженности труда, являющиеся безопасными в отношении сохранения здоровья и работоспособности.

Таблица 4.2 – Предельные уровни звука, дБ, на рабочих местах

Категория напряженности труда	Категория тяжести труда			
	Легкая	Средняя	Тяжелая	Очень тяжелая
Мало напряженный	80	80	75	75
Умеренно напряженный	70	70	65	65
Напряженный	60	60	-	-
Очень напряженный	50	50	-	-

Уровень шума на рабочем месте математиков-программистов и операторов видеоматериалов не должен превышать 50дБА, а в залах обработки информации на вычислительных машинах – 65дБА.

Для снижения уровня шума стены и потолок помещений, где установлены компьютеры, могут быть облицованы звукопоглощающими материалами.

4.2. Электромагнитные и ионизирующие излучения

Большинство ученых считают, что как кратковременное, так и длительное воздействие всех видов излучения от экрана монитора не опасно для здоровья персонала, обслуживающего компьютеры. Однако исчерпывающих данных относительно опасности воздействия излучения от мониторов на работающих с компьютерами не существует и исследования в этом направлении продолжаются. Допустимые значения параметров неионизирующих электромагнитных излучений от монитора компьютера 4.3 в соответствии с СанПиНом 2.2.2.542-2003 представлены в таблице.

Максимальный уровень рентгеновского излучения на рабочем месте оператора компьютера обычно не превышает 10мкбэр/ч, а интенсивность ультрафиолетового и инфракрасного излучений от экрана монитора лежит в пределах 10.100мВт/м^2.

Таблица 4.3 – Допустимые значения параметров неионизирующих электромагнитных излучений

Наименование параметра	Допустимые значения
Напряженность электрической составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	10В/м
Напряженность магнитной составляющей электромагнитного поля на расстоянии 50см от поверхности видеомонитора	0,3А/м

Продолжение таблицы 4.3

Напряженность электростатического поля для взрослых пользователей не должна превышать	20кВ/м

Для снижения воздействия этих видов излучения рекомендуется применять мониторы с пониженным уровнем излучения устанавливать защитные экраны, а также соблюдать регламентированные режимы труда и отдыха.

Как уже было неоднократно отмечено, при работе с персональным компьютером очень важную роль играет соблюдение правильного режима труда и отдыха.

В таблице 4.4 представлены сведения о регламентированных перерывах, которые необходимо делать при работе на компьютере, в зависимости от продолжительности рабочей смены, видов и категорий трудовой деятельности с видеодисплейным терминалом и ПЭВМ в соответствии с СанПиНом 2.2.2 542-2003 «Гигиенические требования к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работ».

Таблица 4.4 – Время регламентированных перерывов при работе на компьютере

Категория работы с ВДТ или ПЭВМ	Уровень нагрузки за рабочую смену при видах работы с ВДТ			Суммарное время регламентированных перерывов в минутах	
	Группа А, количество знаков	Группа Б, количество знаков	Группа В, часов	При 8-часовой смене	При 12-часовой смене
I	до 20000	до 15000	до 2,0	30	70
II	до 40000	до 30000	до 4,0	50	90
III	до 60000	до 40000	до 6,0	70	120

В противном случае у персонала отмечаются значительное напряжение зрительного аппарата с появлением жалоб на неудовлетворенность работой, головные боли, раздражительность, нарушение сна, усталость и болезненные ощущения в глазах, в пояснице, в области шеи и руках.

Примечание. Время перерывов дано при соблюдении указанных санитарных правил и норм.

При несоответствии фактических условий труда требованиям Санитарных правил и норм время регламентированных перерывов следует увеличить на 30%.

В соответствии со СанПиН 2.2.2 546-2003 все виды трудовой деятельности, связанные с использованием компьютера, разделяются на три группы:

- группа А: работа по считыванию информации с экрана ВДТ или ПЭВМ с предварительным запросом;
- группа Б: работа по вводу информации;
- группа В: творческая работа в режиме диалога с ЭВМ.

Эффективность перерывов повышается при сочетании с производственной гимнастикой или организации специального помещения для отдыха персонала с удобной мягкой мебелью, аквариумом, зеленой зоной и т.п.

4.3. Эргономические требования к рабочему месту

Проектирование рабочих мест, снабженных видеотерминалами, относится к числу важных проблем эргономического проектирования в области вычислительной техники.

Рабочее место и взаимное расположение всех его элементов должно соответствовать антропометрическим, физическим и психологическим требованиям. Большое значение имеет также характер работы. В частности, при организации рабочего места программиста должны быть соблюдены следующие основные условия: оптимальное размещение оборудования,

входящего в состав рабочего места и достаточное рабочее пространство, позволяющее осуществлять все необходимые движения и перемещения.

Эргономическими аспектами проектирования видеотерминальных рабочих мест, в частности, являются: высота рабочей поверхности, размеры пространства для ног, требования к расположению документов на рабочем месте (наличие и размеры подставки для документов, возможность различного размещения документов, расстояние от глаз пользователя до экрана, документа, клавиатуры и т.д.), характеристики рабочего кресла, требования к поверхности рабочего стола, регулируемость элементов рабочего места.

Главными элементами рабочего места программиста являются стол и кресло. Основным рабочим положением является положение сидя. Рабочая поза сидя вызывает минимальное утомление программиста. Рациональная планировка рабочего места предусматривает четкий порядок и постоянство размещения предметов, средств труда и документации.

То, что требуется для выполнения работ чаще, расположено в зоне легкой досягаемости рабочего пространства. Моторное поле – пространство рабочего места, в котором могут осуществляться двигательные действия человека. Максимальная зона досягаемости рук – это часть моторного поля рабочего места, ограниченного дугами, описываемыми максимально вытянутыми руками при движении их в плечевом суставе. Оптимальная зона – часть моторного поля рабочего места, ограниченного дугами, описываемыми предплечьями при движении в локтевых суставах с опорой в точке локтя и с относительно неподвижным плечом.

Большое значение придается характеристикам рабочего кресла. Так, рекомендуемая высота сиденья над уровнем пола находится в пределах 420-550мм. Поверхность сиденья мягкая, передний край закругленный, а угол наклона спинки – регулируемый. Необходимо предусматривать при проектировании возможность различного размещения документов: сбоку от видеотерминала, между монитором и клавиатурой и т.п. Кроме того, в случаях,

когда видеотерминал имеет низкое качество изображения, например, заметны мелькания, расстояние от глаз до экрана делают больше (около 700мм), чем расстояние от глаза до документа (300-450мм).

Вообще при высоком качестве изображения на видеотерминале расстояние от глаз пользователя до экрана, документа и клавиатуры может быть равным.

Большое значение также придается правильной рабочей позе пользователя. При неудобной рабочей позе могут появиться боли в мышцах, суставах и сухожилиях. Причина неправильной позы пользователей обусловлена следующими факторами: нет хорошей подставки для документов, клавиатура находится слишком высоко, а документы – низко, некуда положить руки и кисти, недостаточно пространство для ног.

В целях преодоления указанных недостатков даются общие рекомендации: лучше передвижная клавиатура; должны быть предусмотрены специальные приспособления для регулирования высоты стола, клавиатуры и экрана, а также подставка для рук.

Существенное значение для производительной и качественной работы на компьютере имеют размеры знаков, плотность их размещения, контраст и соотношение яркостей символов и фона экрана. Если расстояние от глаз оператора до экрана дисплея составляет 60.80 см, то высота знака должна быть не менее 3мм, оптимальное соотношение ширины и высоты знака составляет 3:4, а расстояние между знаками – 15.20% их высоты. Соотношение яркости фона экрана и символов – от 1:2 до 1:15.

Во время пользования компьютером медики советуют устанавливать монитор на расстоянии 50-60 см от глаз. Когда человек смотрит прямо перед собой, его глаза открываются шире, чем когда он смотрит вниз. За счет этого площадь обзора значительно увеличивается, вызывая обезвоживание глаз. К тому же если экран установлен высоко, а глаза широко открыты, нарушается функция моргания.

Это значит, что глаза не закрываются полностью, не омываются слезной жидкостью, не получают достаточного увлажнения, что приводит к их быстрой утомляемости.

Создание благоприятных условий труда и правильное эстетическое оформление рабочих мест на производстве имеет большое значение, как для облегчения труда, так и для повышения его привлекательности, положительно влияющей на производительность труда.

4.4. Выводы

В главе проведен анализ неблагоприятных факторов, воздействующих на пользователя, а также приведены общие мероприятия по безопасности жизнедеятельности на объекте.

Даны характеристики рабочего места оператора, параметры микроклимата в серверном помещении, а также характеристики выделяемой и избыточной теплоты в серверном помещении.

5. Технико-экономические показатели дипломного проекта

5.1. Определение трудоемкости разработки

Технико-экономическое обоснование должно содержать:

1) определение трудоемкости разработки;
2) подсчет затрат на разработку проекта;
3) вычисление ориентировочной цены проекта;
4) обоснованный выбор программных и аппаратно-программных средств;
5) оценку социально-экономических результатов функционирования дипломного проекта.

Для определения трудоемкости разработки необходимо выделить основные этапы разработки, которые необходимо выполнить. Однако стоит отметить, что при анализе затрат времени на проектирование и разработку программного обеспечения возникает сложность, такая же как и при нормировании творческого труда, имеющего технические работы. Техническая реализация работы программистов практически не имеет нормы, поэтому оценка дается либо на основе экспертных оценок опытных программистов, либо жестко зафиксированными сроками на реализацию, в рамках которых программист должен найти решение. Техническая реализация труда программистов способна нормироваться, но точность нормирования в таком случае имеет большой разброс в зависимости от целого ряда факторов. В таблице 5.1 выделены основные этапы и виды работы, с примерной оценкой их трудоемкости выполнения.

Таблица 5.1 – Распределение работ по этапам и видам и оценка их трудоемкости.

Этап проведения	Вид работы на данном этапе	Трудоемкость выполнения, чел.-ч.
Сбор информации о предметной области	Сбор данных о предметной области: классификация сетевых атак, методы обнаружения сетевых атак, нейронные сети.	145
	Обработка собранных данных	29
Разработка программы	Определение компонентов системы	10
	Изучение технологий для разработки компонентов системы	90
	Разработка и отладка компонентов системы	210
Итоговая трудоемкость выполнения дипломного проекта		484

5.2. Расчет затрат на разработку системы

Для оценки расходов на реализацию составляется определенная смета, которая имеет следующие пункты:

1) затраты на оплату труда;
2) отчисления на социальные нужды;
3) амортизация основных фондов;
4) прочие затраты.

Общая сумма затрат на оплату труда ($З_{ТР}$) определяется по форме, приведенной в таблице 5.2. В статью «Затраты на оплату труда» включаются расходы по оплате труда всех работников, занятых разработкой.

Общее время работы программиста Т определяется из таблицы 5.1 и равно 484 часа. Федеральным агентством по образованию РФ установлены следующие нормы затрат рабочего времени на одну дипломную работу:

руководитель работы 20 ч, консультант по БЖД – 2 ч, консультант по экономической части – 2 ч.

Таблица 5.2 – Затраты на оплату труда

Категория работника	Квалификация	Трудоемкость разработки, чел.-ч.	Часовая ставка, руб/ч	Сумма, руб
Разработчик программы	студент-программист	484	16	7744
Руководитель диплома	старший преподаватель	20	150	3 000
Консультант по БЖД	доцент	2	300	600
Консультант по экономической части	доцент	2	300	600
Итого				11944

Общая сумма затрат на оплату труда ($З_{ТР}$) определяется по формуле:

$$З_{ТР} = \sum_{i=1}^{n} ЧС_i \times T_i \quad (5)$$

где $ЧС_i$ – часовая ставка i-го работника, руб.,

T_i – время на разработку, час

i – категория работника,

n – количество работников, занятых разработкой.

Среднечасовая заработная плата рассчитывается по формуле:

$$ЧС_i = \frac{ЗП_i}{ФРВ_i} \quad (6)$$

где $ЗП_i$ – среднемесячная заработная плата разработчика, руб.;

$ФРВ_i$ – среднемесячный фонд рабочего времени (приблизительно 100 часов в месяц).

Стоимость одного часа работы студента-программиста равна:

$$\text{ЧС}_i = \frac{1600}{100} = 16\text{руб.}$$

Стоимость одного часа работы старшего преподавателя равна:

$$ЧС_i = \frac{15000}{100} = 150\text{руб.}$$

Стоимость одного часа работы доцента равна:

$$ЧС_i = \frac{30000}{100} = 300\text{руб.}$$

Общая сумма затрат на оплату труда равна:

$$\text{З}_{\text{ТР}} = 484 \times 16 + 20 \times 150 + 2 \times 300 + 2 \times 300 = 11\ 944\text{руб.}$$

В статью «Отчисления на социальные нужды» включаются сумма единого социального налога и взносы на страхование от несчастных случаев и профессиональных заболеваний, которые составляют соответственно 35,6% и 0,2% (для НИ РХТУ) от затрат на оплату труда всех работников, занятых выполнением. Студенческие стипендии данным налогом не облагаются.

Отчисления на социальные нужды составят:

$$З_{СН} = (3000 + 600 + 600) \times 0.365 + (3000 + 600 + 600) \times 0.002 = 1\ 541{,}4\text{руб}$$

В статью «Амортизация основных фондов» включается сумма амортизационных отчислений от стоимости оборудования и приборов, используемых при разработке. Расчет амортизационных отчислений приведен в таблице 5.3.

Общая сумма амортизационных отчислений определяется по формуле:

$$З_{АМ} = \sum_{i=1}^{n} \frac{Ф_i \times Н_{Ai} \times Т_{АИСi}}{100 \times Т_{ЭФi}} \quad (7)$$

где $Ф_i$ – стоимость i-го оборудования, руб.;

$Н_{Ai}$ – годовая норма амортизации i-го оборудования, %;

$Т_{АИСi}$ – время работы i-го оборудования за весь период разработки, ч;

$Т_{ЭФi}$ – эффективный фонд времени работы i-го оборудования за год, ч/год;

i – вид оборудования;

n – количество оборудования.

Таблица 5.3 – Расчет амортизационных отчислений

Наименование оборудования	Стоимость оборудования, руб	Годовая норма амортизации, %	Время работы оборудования во время разработки, ч	Сумма, руб.
Компьютер	70000	20	484	3 025
Итог				3 025

Сумма амортизационных отчислений составит

$$З_{АМ} = \frac{70000 \times 20 \times 484}{100 \times 2240} = 3\ 025 руб$$

В статью «Прочие затраты» включаются расходы на содержание административно-управленческого и учебно-вспомогательного персонала, на отопление, освещение и текущий ремонт помещений, канцелярские, командировочные и прочие хозяйственные расходы. Затраты по этой статье принимаются в размере 70 % от затрат на оплату труда

$$З_{ПР} = 0.7 \times 11\ 944 = 8\ 360{,}8 руб.$$

Общая сумма затрат на электроэнергию ($З_э$) рассчитывается по формуле:

$$З_э = \sum_{i=1}^{n} M_i \times K_i \times T_i \times Ц \quad (8)$$

где M_i – паспортная мощность i-го электрооборудования, кВт;

K_i – коэффициент использования мощности i-го электрооборудования (принимается $K_i = 0.7 \div 0.9$);

T_i – время работы i-го оборудования за весь период разработки, ч;

$Ц$ – цена электроэнергии, руб/кВт·ч.

i – вид электрооборудования;

n – количество электрооборудования.

Таблица 5.4 – Смета затрат на разработку

Наименование оборудования	Паспортная мощность, кВт	Коэффициент использования мощности	Время работы оборудования для разработки, ч	Цена электроэнергии, $\frac{руб.}{кВт \cdot ч}$	Сумма, руб.
Компьютер №1	0.4	0.85	484	4.63	627.4576
ИТОГО затраты на электроэнергию					627.4576

Общая сумма затрат на электроэнергию составляет:

$З_э = 0.4 \times 0.7 \times 484 \times 4.63 = 627.4576$руб.

На основании полученных данных по отдельным статьям составляется смета затрат на разработку дипломного проекта по форме, приведенной в таблице 5.5.

Таблица 5.5 – Смета затрат на разработку

Статьи затрат	Сумма, руб.
1. Затраты на оплату труда	11944
2. Отчисления на социальные нужды	1541,4
3. Амортизация основных фондов	3025
4. Прочие затраты	8360,8
5.Затраты на электроэнергию	627,4576
Итого по смете	25 498,6576

Затраты на разработку составят $З_{АИС} = 25\ 498{,}6576$руб.

Величина договорной цены устанавливается с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам потребителя и исполнителя.

Договорная цена ($Ц_Д$) рассчитывается по формуле:

$$Ц_Д = З_{АИС} \times \left(1 + \frac{Р}{100}\right) \quad (9)$$

где $З_{АИС}$ – затраты на разработку, руб.;

$Р$ – средний уровень рентабельности, % (принимается в размере 25%).

Исходя из этого, договорная цена данной будет следующей:

$$Ц_Д = 25498{,}6576 \times \left(1 + \frac{25}{100}\right) = 31\ 873{,}322\text{руб.}$$

Таким образом, учитывая стоимость вычислительной техники, общая стоимость данного проекта будет приблизительно составлять:

$$Ц = 31\ 873{,}322 + 70000 = 101\ 873{,}322\ \text{руб.}$$

5.3. Экономическое обоснование выбора комплекса технических и программных средств

Для быстрой и качественной разработки требуется компьютер с GPU от компании NVIDIA, так как технология CUDA доступна только на видеокартах этой компании. Изучив рынок персональных компьютеров был выбран Lenovo Legion T530. Его характеристики:

- операционная система: Windows 10 Home;
- процессор: Intel Core i5-8400;
- socket: LGA1151 v2;
- частота процессора: 2800 МГц;
- количество ядер процессора: 6;
- размер оперативной памяти: 8 ГБ;
- тип памяти: DDR4;
- частота оперативной памяти: 2666 МГц;
- общий объем накопителей HDD: 1 ТБ;
- общий объем накопителей SSD: 256 ГБ;
- видеокарта: NVIDIA GeForce GTX 1060;
- объем видеопамяти: 6 ГБ;
- мощность блока питания: 450 Вт;
- размеры (ШхВхГ): 185x456x440 мм.

В качестве IDE выбрана Visual Studio Code. Visual Studio Code позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

5.4. Социально-экономический эффект от разработки

Целью разработки является достижение социального эффекта. Но также нужно учитывать материальные затраты на реализацию и установку. Затраты необходимы для:

- обновления мощностей, где функционирует разработка;
- обучения персонала.

Социальный эффект от программы:

- облегчение работы системного администратора;
- своевременная реакция на сетевые атаки.

Одно из перспективных направление применения искусственных нейронных сетей (ИНС) – промышленное производство. В этой области ощутима тенденция перехода к производственным модулям с высоким уровнем автоматизации, что требует увеличения количества интеллектуальных саморегулирующихся и самонастраивающихся машин. Однако, производственным процессам свойственно большое разнообразие динамически взаимодействующих параметров, что усложняет создание адекватных аналитических моделей. Современное производство постоянно усложняется. Это замедляет внедрение новых технологических решений. Кроме того, в ряде случаев удачные аналитические математические модели показывают несостоятельность из-за недостатка вычислительных мощностей. В связи с этим возрастает интерес к альтернативным подходам моделирования производственных процессов с использованием ИНС, предоставляющим возможности создавать модели, работающие в реальном времени с малыми погрешностями, способные дообучаться в процессе использования.

5.5. Выводы

В главе произведено определение трудоемкости разработки, в ходе которого были выделены основные этапы разработки и оценена трудоемкость их выполнения.

Исходя из трудоемкости выполнения произведен расчет затрат на разработку системы. Затраты составляют 101873,322 рублей. Так же был выбран компьютер, подходящий под требования разработки, им стал выбран Lenovo Legion T530 стоимостью 69879 рублей. В качестве IDE для разработки выбрана Visual Studio Code.

Так же определён социально-экономический эффект от разработки. Система облегчит работу системного администратора и позволит своевременно реагирования на сетевые атаки.

Заключение

В результате выполнения дипломного проекта разработана система обнаружения атак на сетевые ресурсы с применением нейросетевых технологий.

В процессе достижения поставленной цели решены задачи:

1. Проведен анализ существующих методов анализа и классификации трафика.
2. Исследованы технологий и программного обеспечения используемые для разработки компонентов системы.
3. Спроектирована и разработана программа сборщика данных.
4. Спроектирована и разработана программа анализатор.
5. Спроектировано и разработано веб приложение для просмотра подозрительной активности.
6. Спроектированы и разработаны две модели нейронных сетей.

Модели способны анализировать данные из сети, даже если данные неполные или искажены. Каждая модель может проводить анализ в нелинейной форме. Обе эти характеристики важны для использования в сетевых технологиях, где информация часто подвергается случайным ошибкам оборудования.

Обе модели нейронных сетей показали точность близкую к 100%, что подтвердилось в ходе тестирования системы.

Дипломный проект учитывает современные тенденции в информационных технологиях и является экономически выгодным решением.

Список использованной литературы

1. ГОСТ Р 53114-2008 - Защита информации. Обеспечение информационной безопасности в организации. Основные термины и определения.
2. Сетевые атаки и технологии их обнаружения. [Электронный ресурс]. - http://www.inf74.ru/safety/ofitsialno/setevyie-ataki-i-tehnologii-ihobnaruzheniya/
3. Астахов А. Актуальные вопросы выявления сетевых атак. Jet Info – информационный бюллетень, 2002.
4. Технологии обнаружения и предотвращения атак. [Электронный ресурс]. - http://www.cnews.ru/reviews/free/security/part8/
5. И.С. Регистрация событий в системах обнаружения компьютерных атак // Информационное противодействие угрозам терроризма. – 2005. – № 5. – С. 106-109.
6. Гамаюнов Д.Ю. Обнаружение компьютерных атак на основе анализа поведения сетевых объектов: дис./ Мос. гос. ун-т имени М.В. Ломоносова, Москва 2007. – С. 7.
7. Марченко А.А., Матвиенко С.В., Нестерук Ф.Г. К обнаружению атак в компьютерных системах нейросетевыми средствами // Научно-технический вестник информационных технологий, механики и оптики. – 2007. – № 39. – С. 83-93.
8. Абдулхаков А.Р., Катасёв А.С., Кирпичников А.П. Методы редукции нечетких правил в базах знаний интеллектуальных систем // Вестник Казан. технол. ун-та. – 2014. – Т. 17. – № 23. – С. 389-392.
9. Нейронные сети - математический аппарат. [Электронный ресурс]. - http://www.basegroup.ru/library/analysis/neural/math/
10. ИНТУИТ. Основы теории нейронных сетей. [Электронный ресурс]. - http://www.intuit.ru/studies/courses/88/88/info

11. Обучение нейронной сети [Электронный ресурс]. - http://www.aiportal.ru/articles/neural-networks/learning-neunet.html

12. Курочкина И.П., Калинин И.И., Маматова Л.А., Шувалова Е.Б. Метод нейронных сетей в моделировании финансовых показателей компании // Статистика и экономика. 2017. №5. – 37.

13. Python - Википедия [Электронный ресурс]. - https://ru.wikipedia.org/wiki/Python

14. PostgreSQL – Википедия [Электронный ресурс] - https://ru.wikipedia.org/wiki/PostgreSQL

15. Pandas - Википедия [Электронный ресурс] - https://ru.wikipedia.org/wiki/Pandas

16. Numpy – Pythonworld [Электронный ресурс] - https://pythonworld.ru/numpy/1.html

17. TensorFlow - Википедия [Электронный ресурс]. - https://ru.wikipedia.org/wiki/TensorFlow

18. Keras - Википедия [Электронный ресурс]. - https://ru.wikipedia.org/wiki/Keras

19. Модель нейронной сети (Network model) [Электронный ресурс] - https://wiki.loginom.ru/articles/network-model.html

20. Свёрточная нейронная сеть – Википедия [Электронный ресурс] - https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть

21. DoS - атака – Википедия [Электронный ресурс] - https://ru.wikipedia.org/wiki/DoS-атака

22. Сканнер портов – Port scanner – Википедия [Электронный ресурс] - https://ru.qwe.wiki/wiki/Port_scanner

Приложение А

Исходный код программы сборщика данных

main.py

```
import logging
import asyncio
import time
import os
from aiohttp import web
import functools
from functools import partial

routes = web.RouteTableDef()

@routes.get("/")
async def index(request):
    send_object = request.app.objects_data
    request.app.objects_data = []
    return web.json_response(send_object)

class App(web.Application):
    def __init__(self):
        super().__init__()
        self.on_startup.append(self.prepare)

    async def prepare(self, app):
        self.objects_data = []
        loop = asyncio.get_event_loop()
        thing = functools.partial(self.file_write, file_folder)
        asyncio.ensure_future( loop.run_in_executor(None, thing))
        self.add_routes(routes)

    def file_write(self, path):
        file_list = os.listdir(path)
        for file in file_list:
            os.remove(f"{path}/{file}")
        empty_folder = True
        while empty_folder:
            file_list = os.listdir(path)
            if len(file_list) != 0:
                empty_folder = False
        path = f"{path}/{file_list[0]}"
        time.sleep(5)
        with open(path, 'r') as f:
```

```
            prew_len_objects_data = 0
            headers_line = f.readline()
            headers_line = headers_line[:-1]
            headers_line = headers_line.split(',')
            while True:
                line = f.readline()
                if line == headers_line:
                    continue
                if line:
                    new_line = line.strip()
                    data = new_line.split(',')
                    scan_object = {}
                    for i in range(len(headers_line)):
                        scan_object[f'{headers_line[i]}'] = data[i]
                    self.objects_data.append(scan_object)
                    logging.debug(f'{scan_object}\n')
                else:
                    if prew_len_objects_data != len(self.objects_data):
                        logging.info(f'There are {len(self.objects_data)} objects in the queue')
                        prew_len_objects_data = len(self.objects_data)
                    time.sleep(.100)

if __name__ == '__main__':
    file_folder = 'file.csv'
    app_port = 5000
    app = App()
    logging.basicConfig(level=logging.DEBUG)
    web.run_app(app, port=app_port)
```

Приложение Б

Исходный код программы анализатора

main.py

```
import logging
import asyncio
import aiohttp

from config import data_url
from database import Data_model

from nn_analyse import nn_analyse

async def get_new_data():
    try:
        async with aiohttp.ClientSession() as session:
            async with session.get(data_url, ssl=False) as r:
                data = await r.json()
                if r.status == 200:
                    return data
                else:
                    return None
    except:
        return None

async def logging_result(result, data_frame, nn_label):
    logging.info(f'{data_frame["Flow ID"]} [{nn_label}], result: {result}')
    if result == True:
        await Data_model.add(nn_label, data_frame)

async def main():
    logging.basicConfig(level=logging.INFO)
    while True:
        data = await get_new_data()
        if data == None:
            continue
        for data_frame in data:
            logging.debug(f'{data_frame}')
            await logging_result(await nn_analyse.analyse(data_frame), data_frame,
'nn_test')

loop = asyncio.get_event_loop()
loop.run_until_complete(main())
```

nn_analyse.py

```
import asyncio

import functools
from functools import partial

import pandas as pd
import numpy as np

from keras.models import load_model

class nn_analyse():
    model = load_model('nn.h5')
    model._make_predict_function()

    @classmethod
    async def analyse(cls, data):
        loop = asyncio.get_event_loop()
        thing = functools.partial(cls.syn_analyse, cls, data)
        result = await loop.run_in_executor(None, thing)
        return result

    def syn_analyse(cls, data):
        columns = ['Tot Fwd Pkts', 'Tot Bwd Pkts', 'Fwd Pkt Len Max',
                'Fwd Pkt Len Min', 'Fwd Pkt Len Mean',
                'Fwd Pkt Len Std', 'Bwd Pkt Len Max', 'Bwd Pkt Len Min',
                'Bwd Pkt Len Mean', 'Bwd Pkt Len Std', 'Fwd Header Len',
                'Bwd Header Len', 'Pkt Len Min', 'Pkt Len Max',
                'Pkt Len Mean', 'Pkt Len Std', 'PSH Flag Cnt',
                'ACK Flag Cnt', 'Down/Up Ratio', 'Pkt Size Avg',
                'Fwd Seg Size Avg', 'Bwd Seg Size Avg', 'Subflow Fwd Pkts',
                'Subflow Bwd Pkts', 'Fwd Act Data Pkts', 'Fwd Seg Size Min']
        index = [1]
        df = pd.DataFrame(data, columns=columns, index=index)
        for item in list(df.columns.values):
            if item != 'Label':
                df[item]=df[item].astype("Float32")
        try:
            pred = cls.model.predict(df)
        except Exception as f:
            print(f)
            return None
        else:
            pred = np.argmax(pred[0])
```

```
            #Normal = 0 Anomaly=1
            if pred == 0:
                return False
            else:
                return True

database.py
import asyncpg
import asyncio

import config

from datetime import datetime

async def create_pool():
    pool = await asyncpg.create_pool(**config.postgres, min_size=15, max_size=50,
timeout=15)
    return pool

async def close_pool():
    await database_pool.close()

loop = asyncio.get_event_loop()
database_pool = loop.run_until_complete(create_pool())


class Data_model():
    table = 'data_frames'

    @classmethod
    async def add(cls, nn_label, data_frame):
        timestamp = data_frame['Timestamp']
        datetime_object = datetime.strptime(timestamp, '%d/%m/%Y %I:%M:%S %p')
        async with database_pool.acquire() as connection:
            await connection.execute(f"""INSERT INTO {cls.table}
            (Src_IP, Src_Port, Dst_IP, Dst_Port, Protocol, Timestamp,Flow_Duration,
            Tot_Fwd_Pkts, Tot_Bwd_Pkts, TotLen_Fwd_Pkts, TotLen_Bwd_Pkts,
Fwd_Pkt_Len_Max,
            Fwd_Pkt_Len_Min, Fwd_Pkt_Len_Mean, Fwd_Pkt_Len_Std,
Bwd_Pkt_Len_Max, Bwd_Pkt_Len_Min,
            Bwd_Pkt_Len_Mean, Bwd_Pkt_Len_Std, Flow_Byts_s, Flow_Pkts_s,
Flow_IAT_Mean, Flow_IAT_Std,
            Flow_IAT_Max, Flow_IAT_Min, Fwd_IAT_Tot, Fwd_IAT_Mean,
Fwd_IAT_Std, Fwd_IAT_Max, Fwd_IAT_Min,
```

Bwd_IAT_Tot, Bwd_IAT_Mean, Bwd_IAT_Std, Bwd_IAT_Max, Bwd_IAT_Min, Fwd_PSH_Flags, Bwd_PSH_Flags,
Fwd_URG_Flags, Bwd_URG_Flags, Fwd_Header_Len, Bwd_Header_Len, Fwd_Pkts_s, Bwd_Pkts_s, Pkt_Len_Min,
Pkt_Len_Max, Pkt_Len_Mean, Pkt_Len_Std, Pkt_Len_Var, FIN_Flag_Cnt, SYN_Flag_Cnt, RST_Flag_Cnt, PSH_Flag_Cnt,
ACK_Flag_Cnt, URG_Flag_Cnt, CWE_Flag_Count, ECE_Flag_Cnt, Down_Up_Ratio, Pkt_Size_Avg, Fwd_Seg_Size_Avg,
Bwd_Seg_Size_Avg, Fwd_Byts_b_Avg, Fwd_Pkts_b_Avg, Fwd_Blk_Rate_Avg, Bwd_Byts_b_Avg, Bwd_Pkts_b_Avg,
Bwd_Blk_Rate_Avg, Subflow_Fwd_Pkts, Subflow_Fwd_Byts, Subflow_Bwd_Pkts, Subflow_Bwd_Byts,
Init_Fwd_Win_Byts, Init_Bwd_Win_Byts, Fwd_Act_Data_Pkts, Fwd_Seg_Size_Min, Active_Mean,
Active_Std, Active_Max, Active_Min, Idle_Mean, Idle_Std, Idle_Max, Idle_Min, Label)
VALUES(
'{data_frame['Src IP']}', '{data_frame['Src Port']}', '{data_frame['Dst IP']}', '{data_frame['Dst Port']}',
'{data_frame['Protocol']}', '{datetime_object}', '{data_frame['Flow Duration']}', '{data_frame['Tot Fwd Pkts']}',
'{data_frame['Tot Bwd Pkts']}', '{data_frame['TotLen Fwd Pkts']}', '{data_frame['TotLen Bwd Pkts']}', '{data_frame['Fwd Pkt Len Max']}',
'{data_frame['Fwd Pkt Len Min']}', '{data_frame['Fwd Pkt Len Mean']}', '{data_frame['Fwd Pkt Len Std']}', '{data_frame['Bwd Pkt Len Max']}',
'{data_frame['Bwd Pkt Len Min']}', '{data_frame['Bwd Pkt Len Mean']}', '{data_frame['Bwd Pkt Len Std']}', '{data_frame['Flow Byts/s']}',
'{data_frame['Flow Pkts/s']}', '{data_frame['Flow IAT Mean']}', '{data_frame['Flow IAT Std']}', '{data_frame['Flow IAT Max']}',
'{data_frame['Flow IAT Min']}', '{data_frame['Fwd IAT Tot']}', '{data_frame['Fwd IAT Mean']}', '{data_frame['Fwd IAT Std']}',
'{data_frame['Fwd IAT Max']}', '{data_frame['Fwd IAT Min']}', '{data_frame['Bwd IAT Tot']}', '{data_frame['Bwd IAT Mean']}',
'{data_frame['Bwd IAT Std']}', '{data_frame['Bwd IAT Max']}', '{data_frame['Bwd IAT Min']}', '{data_frame['Fwd PSH Flags']}',
'{data_frame['Bwd PSH Flags']}', '{data_frame['Fwd URG Flags']}', '{data_frame['Bwd URG Flags']}', '{data_frame['Fwd Header Len']}',
'{data_frame['Bwd Header Len']}', '{data_frame['Fwd Pkts/s']}', '{data_frame['Bwd Pkts/s']}', '{data_frame['Pkt Len Min']}',
'{data_frame['Pkt Len Max']}', '{data_frame['Pkt Len Mean']}', '{data_frame['Pkt Len Std']}', '{data_frame['Pkt Len Var']}',
'{data_frame['FIN Flag Cnt']}', '{data_frame['SYN Flag Cnt']}', '{data_frame['RST Flag Cnt']}', '{data_frame['PSH Flag Cnt']}',

```
            '{data_frame['ACK Flag Cnt']}', '{data_frame['URG Flag Cnt']}',
'{data_frame['CWE Flag Count']}', '{data_frame['ECE Flag Cnt']}',
            '{data_frame['Down/Up Ratio']}', '{data_frame['Pkt Size Avg']}',
'{data_frame['Fwd Seg Size Avg']}', '{data_frame['Bwd Seg Size Avg']}',
            '{data_frame['Fwd Byts/b Avg']}', '{data_frame['Fwd Pkts/b Avg']}',
'{data_frame['Fwd Blk Rate Avg']}', '{data_frame['Bwd Byts/b Avg']}',
            '{data_frame['Bwd Pkts/b Avg']}', '{data_frame['Bwd Blk Rate Avg']}',
'{data_frame['Subflow Fwd Pkts']}', '{data_frame['Subflow Fwd Byts']}',
            '{data_frame['Subflow Bwd Pkts']}', '{data_frame['Subflow Bwd Byts']}',
'{data_frame['Init Fwd Win Byts']}', '{data_frame['Init Bwd Win Byts']}',
            '{data_frame['Fwd Act Data Pkts']}', '{data_frame['Fwd Seg Size Min']}',
'{data_frame['Active Mean']}', '{data_frame['Active Std']}',
            '{data_frame['Active Max']}', '{data_frame['Active Min']}',
'{data_frame['Idle Mean']}', '{data_frame['Idle Std']}',
            '{data_frame['Idle Max']}', '{data_frame['Idle Min']}', '{nn_label}'
        )
        """)

    @classmethod
    async def Create_table(cls) -> True:
        async with database_pool.acquire() as connection:
            await connection.execute(f"
            CREATE TABLE {cls.table}(
                id serial PRIMARY KEY,
                Src_IP cidr,
                Src_Port int,
                Dst_IP cidr,
                Dst_Port int,
                Protocol int,
                Timestamp timestamp,
                Flow_Duration float,
                Tot_Fwd_Pkts float,
                Tot_Bwd_Pkts float,
                TotLen_Fwd_Pkts float,
                TotLen_Bwd_Pkts float,
                Fwd_Pkt_Len_Max float,
                Fwd_Pkt_Len_Min float,
                Fwd_Pkt_Len_Mean float,
                Fwd_Pkt_Len_Std float,
                Bwd_Pkt_Len_Max float,
                Bwd_Pkt_Len_Min float,
                Bwd_Pkt_Len_Mean float,
                Bwd_Pkt_Len_Std float,
                Flow_Byts_s float,
```

Flow_Pkts_s float,
Flow_IAT_Mean float,
Flow_IAT_Std float,
Flow_IAT_Max float,
Flow_IAT_Min float,
Fwd_IAT_Tot float,
Fwd_IAT_Mean float,
Fwd_IAT_Std float,
Fwd_IAT_Max float,
Fwd_IAT_Min float,
Bwd_IAT_Tot float,
Bwd_IAT_Mean float,
Bwd_IAT_Std float,
Bwd_IAT_Max float,
Bwd_IAT_Min float,
Fwd_PSH_Flags float,
Bwd_PSH_Flags float,
Fwd_URG_Flags float,
Bwd_URG_Flags float,
Fwd_Header_Len float,
Bwd_Header_Len float,
Fwd_Pkts_s float,
Bwd_Pkts_s float,
Pkt_Len_Min float,
Pkt_Len_Max float,
Pkt_Len_Mean float,
Pkt_Len_Std float,
Pkt_Len_Var float,
FIN_Flag_Cnt int,
SYN_Flag_Cnt int,
RST_Flag_Cnt int,
PSH_Flag_Cnt int,
ACK_Flag_Cnt int,
URG_Flag_Cnt int,
CWE_Flag_Count float,
ECE_Flag_Cnt int,
Down_Up_Ratio float,
Pkt_Size_Avg float,
Fwd_Seg_Size_Avg float,
Bwd_Seg_Size_Avg float,
Fwd_Byts_b_Avg float,
Fwd_Pkts_b_Avg float,
Fwd_Blk_Rate_Avg float,
Bwd_Byts_b_Avg float,

```
            Bwd_Pkts_b_Avg float,
            Bwd_Blk_Rate_Avg float,
            Subflow_Fwd_Pkts float,
            Subflow_Fwd_Byts float,
            Subflow_Bwd_Pkts float,
            Subflow_Bwd_Byts float,
            Init_Fwd_Win_Byts float,
            Init_Bwd_Win_Byts float,
            Fwd_Act_Data_Pkts float,
            Fwd_Seg_Size_Min float,
            Active_Mean float,
            Active_Std float,
            Active_Max float,
            Active_Min float,
            Idle_Mean float,
            Idle_Std float,
            Idle_Max float,
            Idle_Min float,
            Label text
        )
        '')
        return True

config.py
data_url = 'http://localhost:5000/'

postgres = {
    'host': '111.111.111.111',
    'port': '5432',
    'user': 'user,
    'password': '!XjJt54A@D^v',
    'database': 'diplom',
}
```

Приложение В

Исходный код веб приложения

main.py

```
from aiohttp import web
import logging

from routes import routes

import aiohttp_jinja2
import jinja2

from config import app_port

class App(web.Application):
    def __init__(self):
        super().__init__()
        self.on_startup.append(self.prepare)

    async def prepare(self, app):
        self.add_routes(routes)

if __name__ == '__main__':
    app = App()
    logging.basicConfig(level=logging.DEBUG)
    aiohttp_jinja2.setup(
        app,
        loader=jinja2.PackageLoader('main', 'templates')
    )
    web.run_app(app, port=app_port)
```

routes.py

```
import aiohttp_jinja2
from aiohttp import web

from database import Data_model
routes = web.RouteTableDef()

@routes.get("/")
@aiohttp_jinja2.template('index.html')
async def index(request):
    return {'frames': await Data_model.get_all_frames()}
```

```
@routes.get("/frame/{frame_id}")
@aiohttp_jinja2.template('frame.html')
async def guild(request):
    frame_id = int(request.match_info.get('frame_id'))
    frame = await Data_model.get_frame(frame_id)
    if frame == None:
        raise web.HTTPNotFound

    collumn = 4
    keys = []
    values = []
    for key in frame:
        if str(key) != 'timestamp' and str(key) != 'id' and str(key) !='label':
            keys.append(key)
            values.append(frame[key])

    rows = []
    count = 0
    for i in range(int(len(keys)/collumn)+1):
        collumns = []
        for j in range(collumn):
            if count < len(keys):
                if keys[count] == 'src_ip' or keys[count] == 'dst_ip':
                    values[count] = str(values[count].broadcast_address)
                collumns.append(keys[count])
                collumns.append(values[count])
            count += 1
        rows.append(collumns)
    return {'frame_id': frame['id'], 'frame_label': frame['label'], 'frame_timestamp':
frame['timestamp'], 'rows':rows}

database.py
import asyncpg
import asyncio
import config

async def create_pool():
    pool = await asyncpg.create_pool(**config.postgres, min_size=15, max_size=50,
timeout=15)
    return pool

async def close_pool():
    await database_pool.close()
```

```
loop = asyncio.get_event_loop()
database_pool = loop.run_until_complete(create_pool())


class Data_model():
    table = 'data_frames'
    @classmethod
    async def get_all_frames(cls):
        async with database_pool.acquire() as connection:
            incedents = await connection.fetch(f"""
                SELECT
                id, label, src_ip,
                src_port, dst_ip,
                dst_port, timestamp
                FROM {cls.table}
                ORDER BY timestamp DESC
                """)
            return incedents

    @classmethod
    async def get_frame(cls, frame_id):
        async with database_pool.acquire() as connection:
            frame = dict( await connection.fetchrow(f"""
                SELECT
                *
                FROM {cls.table}
                WHERE id = $1
                """, frame_id))
            return frame
```

config.py

```
app_port = 5000


postgres = {
    'host': '111.111.111.111',
    'port': '5432',
    'user': 'user,
    'password': '!XjJt54A@D^v',
    'database': 'diplom',
}
```

base.html

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
    {% if title %}
    <title>{{ title }}</title>
    {% else %}
    <title>Incidents</title>
    {% endif %}
  </head>
  <body style="background-color: #2C2F33;">
    {% block content %}
    {% endblock %}
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
    {% block javascript %}
    {% endblock %}

  </body>
</html>
```

index.html

```
{% extends "base.html" %}
```

```
{% block content %}
<div class="container-fluid">
   <div class="row">
      {% for frame in frames %}
      <div class="card text-white bg-secondary mb-3" style="width: 18rem; margin:
10px;">
         <div class="card-header">Инцидент №{{ frame.id }}</div>
         <div class="card-body d-flex flex-column" style="padding-bottom: 5px;">
            <div>
               <h5 class="card-title">{{ frame.label }}</h5>
            </div>
            <div class="mt-auto">
               <p class="card-text">
                  {% set x = frame.src_ip |string %}
                  {% set y = frame.dst_ip |string %}
                  Источник {{ x[:-3] }}:{{ frame.src_port}}
                  Назначение {{  y[:-3] }}:{{ frame.dst_port}}
               </p>
            </div>
            <div class="mt-auto">
               <p class="card-text">
                  <small  >{{  frame.timestamp.strftime('%H:%M:%S  %d.%m.%Y  ')
}}</small>
               </p>
            </div>
         </div>
         <div class="card-footer">
            <a  href="/frame/{{  frame.id  }}"  class="card-link  btn  btn-warning"
style="width:100%">Подробнее</a>
         </div>
      </div>
      {% endfor %}
   </div>
</div>

{% endblock %}
```

frame.html

```
{% extends "base.html" %}

{% block content %}
<div class="container-fluid" style="padding: 0; height: 100%;">
   <div class="card text-white bg-secondary h-100">
      <div class="card-header">Инцидент №{{ frame_id }}</div>
```

```
        <div class="card-body d-flex flex-column" style="padding-bottom: 5px;">
          <div>
            <h5 class="card-title">{{ frame_label }}</h5>
          </div>
          <table class="table table-Secondary">
            <tbody>
              {% for row in rows %}
              <tr>
                {% for collumn in row  %}
                  {% if loop.index is divisibleby 2 %}
                    <td>{{collumn}}</td>
                  {% else %}
                    <td style="background-color: #444a4f;">{{collumn}}</td>
                  {% endif %}
                {% endfor %}
              </tr>
              {% endfor %}
            </tbody>
          </table>
        </div>
        <div class="card-footer">
          <small    >{{    frame_timestamp.strftime('%H:%M:%S    %d.%m.%Y    ')
}}</small>
        </div>
      </div>
</div>
{% endblock %}
```