Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования «Казанский (Приволжский) федеральный университет»
Институт вычислительной математики и информационных технологий

Кафедра системного анализа и информационных технологий

Направление подготовки: 10.03.01 — Информационная безопасность
Профиль: Безопасность компьютерных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РЕШЕНИЕ ЗАДАЧ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ С ПОМОЩЬЮ ИСКУССТВЕННЫХ ИММУННЫХ СЕТЕЙ

Обучающийся 4 курса
группы 09-841 (Мочалов С.А.)

Руководитель
канд. физ.-мат. наук, доцент (Андрианова А.А.)

Заведующий кафедрой системного анализа
и информационных технологий
д-р техн. наук, профессор (Латыпов Р.Х.)

Казань – 2022

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Согласно данным IT-компании «Positive Technologies» [1] общее число кибератак за 2020 год выросло на 51%. Наиболее часто подвергались атакам государственные и медицинские учреждения, а также промышленные предприятия. Такую тенденцию в основном связывают с текущей эпидемиологической ситуацией в мире в связи с чем произошел активный перевод сотрудников на удаленную работу и вывод внутренних сервисов компаний на сетевой периметр. На рисунке 1 представлена тенденция роста инцидентов в зависимости от месяца.

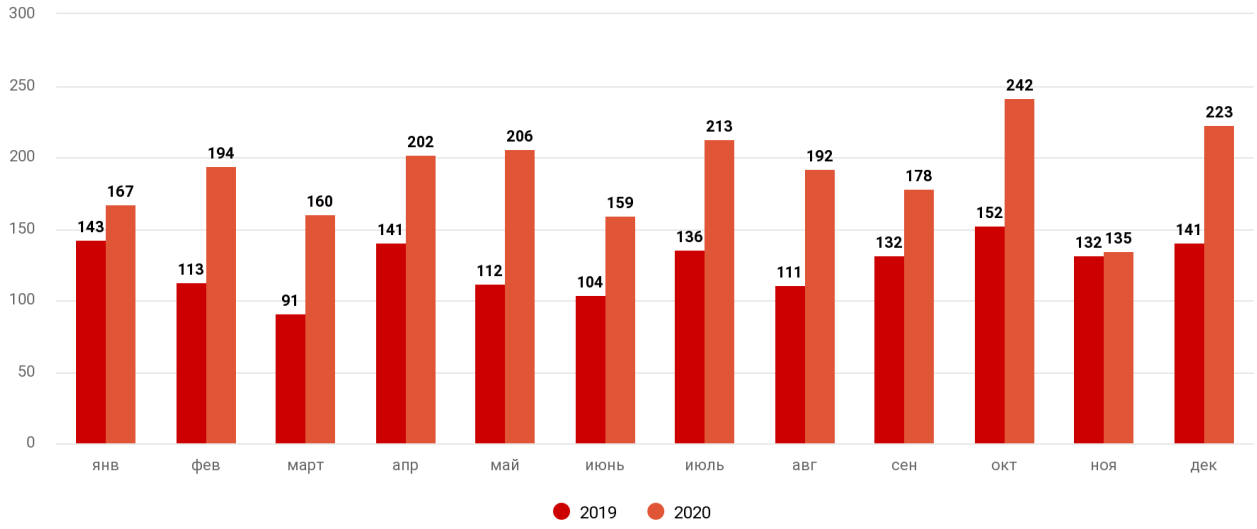

Рисунок 1 - Количество инцидентов в 2019 и 2020 годах

В основном мотивы злоумышленников были направлены на получение данных или получение финансовой выгоды рисунок 2.

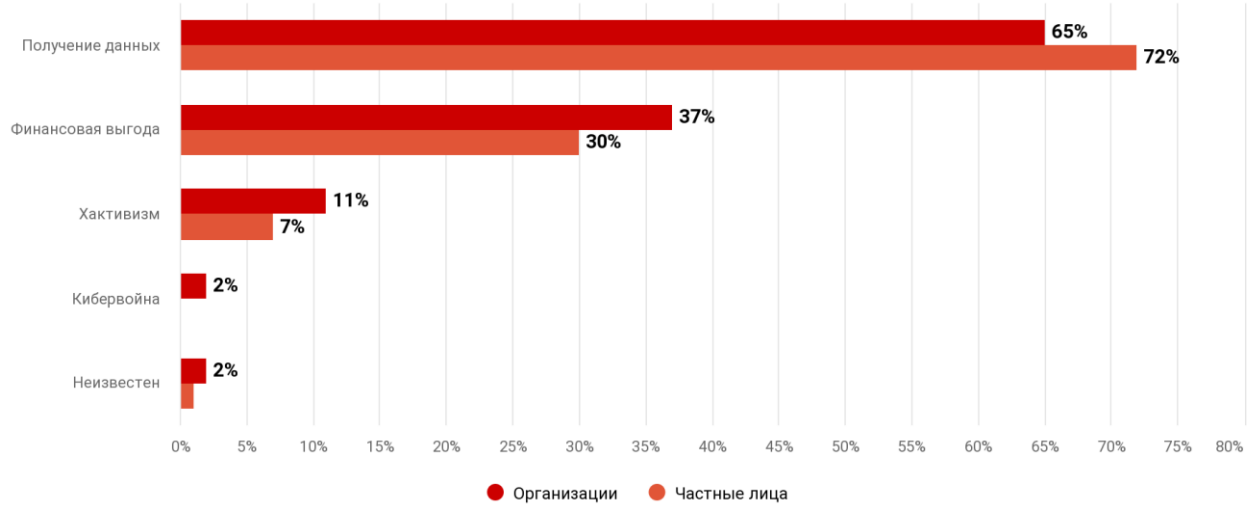

Рисунок 2 - Мотивы злоумышленников

Лишь небольшая часть компаний, которые практиковали удаленную работу, были готовы к такому кризису, остальные столкнулись с нехваткой времени на продумывание и реализацию всех необходимых мер защиты.

Наиболее востребованными средствами защиты от постоянно растущих сетевых атак стали системы обнаружения вторжений (Intrusion Detection Systems) и системы предотвращения вторжений (Intrusion Prevention Systems), зачастую разработанные на базе кластеризации данных. С этим и связана актуальность темы «Решение задач обнаружения вторжений с помощью искусственных иммунных сетей».

Целью выпускной квалификационной работы является создание эффективного алгоритма для задач обнаружения вторжений на основе искусственных иммунных сетей.

Для достижения поставленной цели были выдвинуты следующие задачи:

— поиск наборов данных, применимых для обучения и тестирования искусственных иммунных сетей;
— обработка данных;
— разработка алгоритма искусственных иммунных сетей;
— обучение алгоритма;
— подбор гиперпараметров алгоритма;
— тестирование искусственных иммунных сетей;
— разработка приложения «Система обнаружения вторжений».

Во время написания выпускной квалификационной работы использовались следующие источники.

Данные для обучения, тестирования и исследования модели взяты с электронного ресурса Канадского института по кибербезопасности [2]. Набор данных содержит самые современные и распространенные атаки. Он также включает результаты анализа сетевого трафика с использованием CICFlowMeter с помеченными потоками на основе метки времени, IP-адресов источника и получателя, портов источника и получателя, протоколов и атак.

Для ознакомления с данными и с их дальнейшей обработкой используется статья с электронного ресурса «Хабр» [3]. Автор статьи подробно рассказывает о плюсах и минусах набора данных Канадского института, а также делится идеями обработки этих данных. Помимо описания данных, автор рассказывает о том, как разработать систему обнаружения компьютерных атак на основе машинного обучения самому.

Статья Литвенко В. И. о кластерном анализе данных [4]. Автор рассказывает о проблеме кластеризации при помощи искусственных иммунных сетей, предлагает свой алгоритм обучения иммунной сети и проводит исследование своей разработки.

Статья из сборника материалов Четырнадцатой Всероссийской научно-практической конференции [5] используется как основной источник знаний об искусственных иммунных сетях. Автор статьи описывает алгоритм иммунной сети для решения задачи идентификации рукописного почерка. Для решения этой задачи исследователем выбран алгоритм aiNET. Автор подробно описал методы, которые он применил для решения задачи и подробно рассказал о результатах.

Электронный ресурс «Лекции по машинному обучению и компьютерному зрению от Евгения Разинкова» [6] помогает изучить подходы к решению задач машинного обучения. На канале Евгения Викторовича много интересной базовой и уникальной продвинутой информации о компьютерном зрении, о машинном и о глубоком обучении.

1. Данные для обучения

1.1. Набор данных CICIDS2017

Набор данных CICIDS [2] разработан в 2017 году Канадским институтом кибербезопасности на основе анализа сетевого трафика в изолированной среде. Он содержит 14 типов современных распространённых атак таблица 1.

Таблица 1 – Типы атак в наборе данных CICIDS2017

№	Тип записи	Количество записей
1	Benign	2271320
2	Dos Hulk	230124
3	PortScan	158804
4	DDoS	128025
5	DoS GoldenEye	10293
6	FTP-Patator	7935
7	SSH-Patator	5897
8	DoS Slowloris	5796
9	DoS Slowhttptest	5499
10	Bot	1956
11	Infiltration	36
12	Heartbleed	11
13	Web Attack – Brute Force	1507
14	Web Attack – XSS	652
15	Web Attack – SQL Injection	21

Набор содержит более 50 Гб необработанных данных, представленных в 8 файлах формата CSV [3]. Каждая строка данных содержит 78 параметров сетевого трафика и тип атаки к которому относится запись.

Набор атак в CICIDS2017.

DoS (Denial of Service) атака - это отказ в обслуживании. Злоумышленник нападает с целью вызвать перегрузку подсистемы, в которой работает атакуемый сервис. Воздействие осуществляется с одного сервера и

нацелено на определенный домен или виртуальную машину. Особенности DoS-атак.

1) Одиночный характер – поток трафика запускается из одной-единственной подсети,

2) Высокая заметность – попытки «положить» сайт заметны по содержимому лог-файла,

3) Простота подавления – атаки легко блокируются при помощи брандмауэра.

DDoS (Distributed Denial of Service) атака — это распределенный отказ в обслуживании. Реализуется атака несколько иначе, чем DoS – принципиальное отличие заключается в применении сразу нескольких хостов. Сложность защиты от этого вида нападения зависит от количества машин, с которых осуществляется отправка трафика. Особенности DDoS-атак.

1) Многопоточный характер – такой подход упрощает задачу блокирования сайта, потому что быстро отсеять все атакующие IP-адреса практически нереально;

2) Высокая скрытность – грамотное построение атаки позволяет замаскировать ее начало под естественный трафик и постепенно «забивать» веб-ресурс пустыми запросами;

3) Сложность подавления – проблема заключается в определении момента, когда атака началась.

Bot – это атака на сервер вредоносными ботами, один из возможных инструментов DDoS-атаки. Возможности такой атаки: парсинг (кража контента), накрутка пользователей, скликивание рекламных объявлений.

Dos Hulk — это приложение (стресс-тестер) для тестирования безопасности. Данный стресс-тестер генерирует большой поток уникальных запросов, который максимально потребляет ресурсы веб-сервера.

DoS GoldenEye. GoldenEye — это приложение (стресс-тестер) для тестирования HTTP DoS. Эксплуатируемый вектор атаки: HTTP Keep Alive + NoCache. HTTP — протокол прикладного уровня для передачи произвольных

данных.

DoS Slowloris — это атака, действие которой заключается в очень медленной посылке все новых и новых HTTP заголовков в рамках одного HTTP запроса, никогда его не завершая. Пораженные серверы будут поддерживать соединение открытым, заполняя, таким образом, свой максимальный пул параллельных соединений, в конечном счете, отказывая в дополнительных попытках подключения от клиентов.

DoS Slowhttptest. Slowhttptest — это имеющий множество настроек инструмент (стресс-тестер), симулирующие некоторые атаки отказа в обслуживании (DoS) уровня приложения. Например, в пул атак этого инструмента входят Slowloris, Slow Body и Slow Read. Инструмент Slowhttptest позволяет занимать весь доступный пул подключений.

PortScan — тип атаки, когда порты компьютера сканируются. Согласно этому анализу, злоумышленники могут получить доступ к личной информации такие как состав сетевой архитектуры, операционной системы, возможных дыр в безопасности и так далее.

Brute Force — метод взлома учетных записей путем подбора паролей к ним. Суть подхода заключается в последовательном автоматизированном переборе всех возможных комбинаций символов с целью рано или поздно найти правильную.

FTP-Patator— атака, заключающаяся в подборе пароля учетных записей FTP серверов с помощью инструмента Patator. Протокол FTP (File Transfer Protocol) представляет собой сетевой протокол, используемый для передачи файлов по модели клиент-сервер, когда пользователь подключается к серверу при помощи клиента. Patator — это один из самых продвинутых инструментов подбора пароля, от своих предшественников отличается гибкостью, многопоточностью и использованием постоянных соединений.

SSH-Patator — атака, заключающаяся в подборе пароля на удаленном компьютере для получения к нему доступа по SSH, проводится с помощью инструмента Patator. SSH — сетевой протокол прикладного уровня,

предназначеный для безопасного удаленного доступа к UNIX-системам.

Infiltration — атака проникновения, которая включает неавторизованное приобретение и/или изменение системных привилегий, ресурсов или данных. Нарушает целостность и управляемость системы, противоположность DoS-атакам, которые нарушают доступность ресурсов, и атакам сканирования, которые не делают ничего незаконного.

Heartbleed — уязвимость в безопасности программной библиотеки OpenSSL (открытой реализации протокола шифрования SSL/TLS), которая позволяла хакерам получить доступ к содержимому оперативной памяти серверов, в которых в этот момент могли содержаться приватные данные пользователей различных веб-сервисов. По данным исследовательской компании Netcraft, этой уязвимости могли быть подвержены около 500 тысяч веб-сайтов.

XSS — межсайтовый скриптинг, достаточно распространенная уязвимость суть которой – внедрение на страницу JavaScript-код, который не был предусмотрен разработчиками. Этот код будет выполняться каждый раз, когда жертвы (обычные пользователи) будут заходить на страницу приложения, куда этот код был добавлен. Возможные последствия: получение данных пользователя для входа в аккаунт, перенаправление пользователя на страницу-клона для ввода личных данных и так далее.

SQL Injection — один из самых распространённых способов взлома сайтов и программ, основанный на внедрении в запрос произвольного SQL-кода. В зависимости от типа используемой СУБД и условий внедрения, может дать возможность атакующему выполнить произвольный запрос к базе данных (например, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.

1.2. Обработка данных

Главный минус набора данных CICIDS – отсутствие некоторых значений. Например, в записи может отсутствовать один из параметров или вместо него будет записано значение $\pm\infty$. Устранить эту проблему можно двумя способами – либо удаление всей записи, либо замена отсутствующих параметров.

В данной выборке предоставляется большое количество данных, однако эти данные разбиты по классам неравномерно, поэтому удалять записи для типа атак «Heartbleed» или «SQL Injection» не целесообразно, потому что при небольшом количестве записей модель обучается в разы хуже. Для данной выборки было принято решение заменить все отсутствующие значения на 0.

Рекомендация – в дальнейшем следует удалять записи с отсутствующими значениями для выборок, размерность которых больше 10000, в остальных случаях заменить все отсутствующие значения на 0.

После замены всех отсутствующих значений на 0 было замечено, что некоторые параметры во всех записях равны 0, таких параметров 8. Принято решение исключить эти параметры из выборки. Количество оставшихся параметров 70.

В ходе первичного обучения модели на обучающей выборке было принято решения сократить количество параметров для обучения, так как при обучении модели на 70 параметрах, модель долго обучается и в результате оказывается не точной, также большое количество параметров сказывается отрицательно на памяти. Для сокращения количества параметров был применен коэффициент корреляции Пирсона, который измеряет линейную связь между переменными:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{X})(y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{X})^2 \sum_{i=1}^{n}(y_i - \overline{Y})^2}}.$$

Для оценки силы связи в теории корреляции применяется шкала английского статистика Чеддока таблица 2.

Таблица 2 – Шкала Чеддока

Теснота связи	Значение коэффициента корреляции	
	Прямая связь	Обратная связь
Отсутствует	[0; 0,1]	[-0,1; 0]
Слабая	[0,1; 0,3]	[-0,3; -0,1]
Умеренная	[0,3; 0,5]	[-0,5; -0,3]
Заметная	[0,5; 0,7]	[-0,7; -0,5]
Высокая	[0,7; 0,9]	[-0,9; -0,7]
Весьма высокая	[0,9; 0,1]	[-1; -0,9]

В качестве порогового значения корреляции для определения зависимости между переменными было выбрано значение $k = \pm 0{,}5$. Если корреляция переменных больше k одна из двух переменных заменялась другой.

В результате была получена выборка, состоящая из 2827876 записей, каждая запись которой содержит 25 параметров, таблица 3.

Таблица 3 – Параметры для обучения

Параметр	Описание
Destination Port	Порт назначения
Flow Duration	Продолжительность потока
Total Fwd Packets	Общее количество поступающих пакетов
Total Length of Fwd Packets	Общая длина получаемых пакетов
Fwd Packet Length Max	Максимальная длина получаемого пакета
Fwd Packet Length Min	Минимальная длина получаемого пакета
Bwd Packet Length Max	Максимальная длина отправляемого пакета
Bwd Packet Length Min	Минимальная длина отправляемого пакета
Flow Bytes/s	Поток байт/с
Flow Packets/s	Поток пакетов/с

Продолжение таблицы 3

Параметр	Описание
Flow IAT Min	Среднее время формирования потока
Fwd IAT Min	Среднее время формирования поступающего пакета
Fwd PSH Flags	Поступающие PSH флаги
Fwd URG Flags	Поступающие URG флаги
Fwd Header Length	Длина заголовка поступающего пакета
Bwd Header Length	Длина заголовка отправляемого пакета
Bwd Packets/s	Отправляемые пакеты/с
FIN Flag Count	Количество флагов FIN
RST Flag Count	Количество флагов RST
PSH Flag Count	Количество флагов PSH
Down/Up Ratio	Коэффициент Вниз / Вверх
Init Win bytes backward	Init Win bytes backward
Active Mean	Среднее время действия
Active Std	Стандартное время действия
Idle Std	Стандартное ожидание

Для дальнейшего обучения принято решение поделить выборку по типам атак, чтобы в дальнейшем произвести обучение модели для каждой атаки.

1.3. Связь входных параметров обучающей выборки

Для оценки связи входных параметров каждой из атак выбран критерий схожести – аффинность.

Аффинность – мера взаимодействия (или сила связи) антигена и антитела или двух антител, которая формально может быть представлена в виде одной из метрик, указывающей на степень подобия или различия между соответствующими атрибутами строк. В данном алгоритме в качестве

аффинности используется евклидово расстояние, вычисляющееся по следующей формуле:

$$Aff = \sqrt{\sum_{i=1}^{L}(Ab_i - Ag_i)^2}.$$

Чем меньше значение аффинности, тем сильнее связь параметров атак.

Для вычисления аффинности в качестве параметров каждой из атак выбраны средние значения каждой из выборок. Полученные результаты занесены в таблицу 4. Обозначения в таблице (по алфавиту): 1 – Benign, 2 – Bot, 3 – Brute Force, 4 – DDoS, 5 – DoS GoldenEye, 6 – Dos Hulk, 7 – DOS Slowhttptest, 8 – DoS Slowloris, 9 – FTP-Patator, 10 – Heartbleed, 11 – Infiltration, 12 – PortScan, 13 – SQL Injection, 14 – SSH-Patator, 15 – XSS.

Таблица 4 – Аффинность средних значений параметров атак

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0,407	0,74	0,34	0,552	0,549	0,672	0,66	0,53	1,148	0,781	0,751	0,36	0,297	0,815
2	0,407	0	0,52	0,425	0,432	0,851	0,6	0,658	0,582	1,345	1,016	0,398	0,283	0,303	0,588
3	0,74	0,52	0	0,628	0,5	1,059	0,593	0,698	0,733	1,465	1,158	0,394	0,491	0,535	0,08
4	0,342	0,425	0,628	0	0,315	0,58	0,546	0,586	0,565	1,027	0,847	0,627	0,25	0,242	0,701
5	0,551	0,432	0,5	0,315	0	0,773	0,432	0,495	0,623	1,129	0,952	0,453	0,313	0,353	0,562
6	0,549	0,851	1,059	0,579	0,773	0	0,799	0,782	0,86	0,778	0,689	1,109	0,743	0,69	1,124
7	0,672	0,6	0,593	0,546	0,432	0,799	0	0,251	0,628	1,137	0,741	0,584	0,538	0,54	0,639
8	0,66	0,658	0,698	0,586	0,495	0,782	0,251	0	0,536	1,134	0,611	0,698	0,593	0,584	0,746
9	0,53	0,582	0,733	0,565	0,623	0,86	0,626	0,536	0	1,345	0,717	0,717	0,506	0,496	0,794
10	1,148	1,345	1,465	1,027	1,129	0,777	1,137	1,134	1,345	0	1,001	1,518	1,249	1,225	1,511
11	0,781	1,016	1,158	0,847	0,952	0,689	0,741	0,611	0,717	1,001	0	1,21	0,944	0,896	1,212
12	0,751	0,398	0,394	0,627	0,453	1,109	0,584	0,698	0,717	1,518	1,21	0	0,452	0,515	0,434
13	0,36	0,283	0,491	0,25	0,313	0,743	0,538	0,593	0,506	1,249	0,944	0,452	0	0,079	0,57

Продолжение таблицы 4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14	0,297	0,303	0,535	0,242	0,353	0,69	0,54	0,584	0,496	1,225	0,895	0,515	0,079	0	0,614
15	0,815	0,588	0,08	0,701	0,562	1,124	0,639	0,746	0,794	1,512	1,212	0,434	0,57	0,614	0

Выборки с аффинностью менее 4 и более 1 являются схожими, выборки с аффинностью менее 1, скорей всего, пересекаются и поэтому их крайне сложно кластеризовать, рисунок 3. На практике выборки с аффинность менее 1 можно объединить в один кластер.

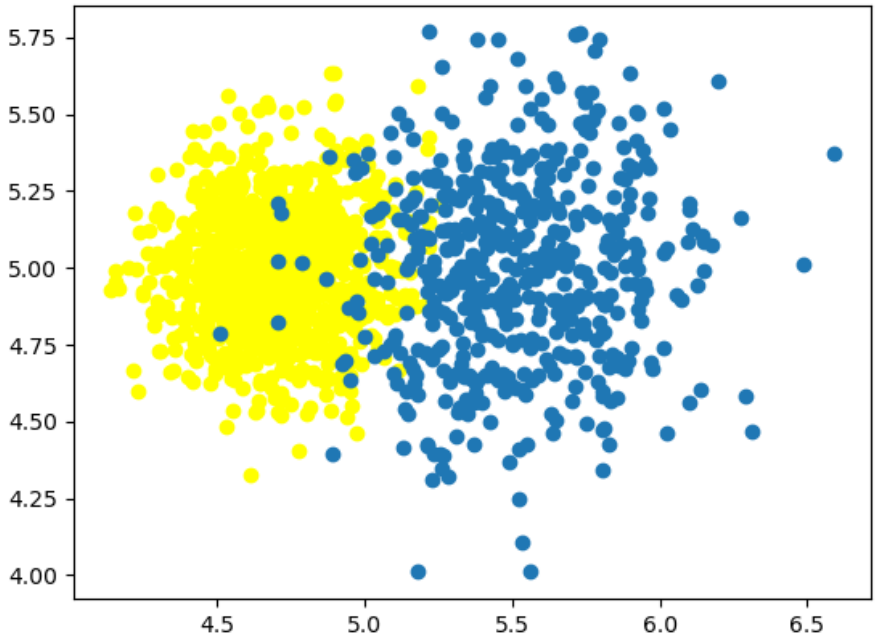

Рисунок 3 – Пересекающиеся выборки

Минус полученных данных заключается в том, что здесь рассмотрены только центры кластеров, и отсутствует информация, как далеко от них разбросаны элементы выборки (рисунки 4, 5). По этой причине в дальнейшем рекомендуется для каждой выборки подобрать параметры, которые будут характеризовать именно ее.

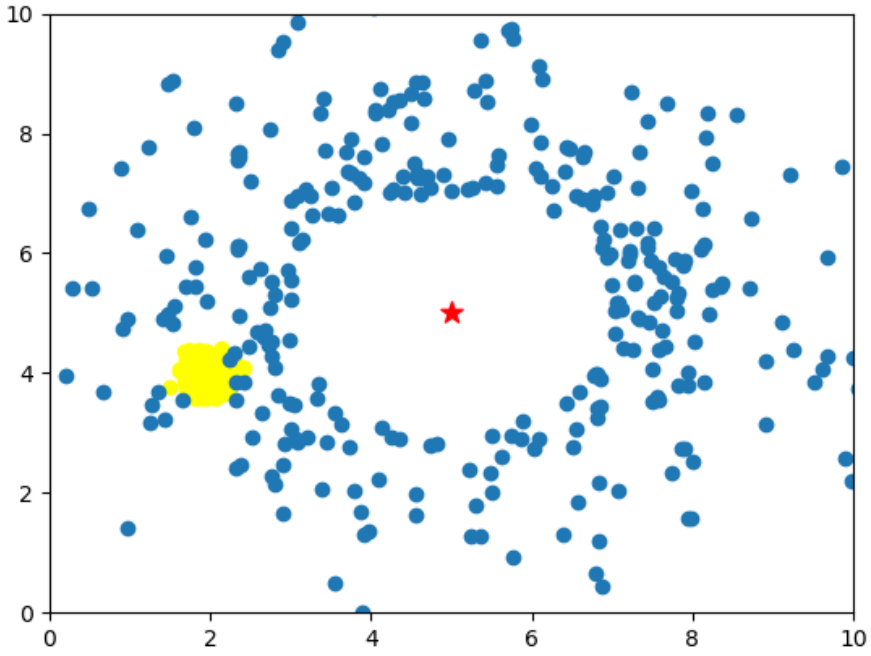

Рисунок 4 – Разбросанные данные

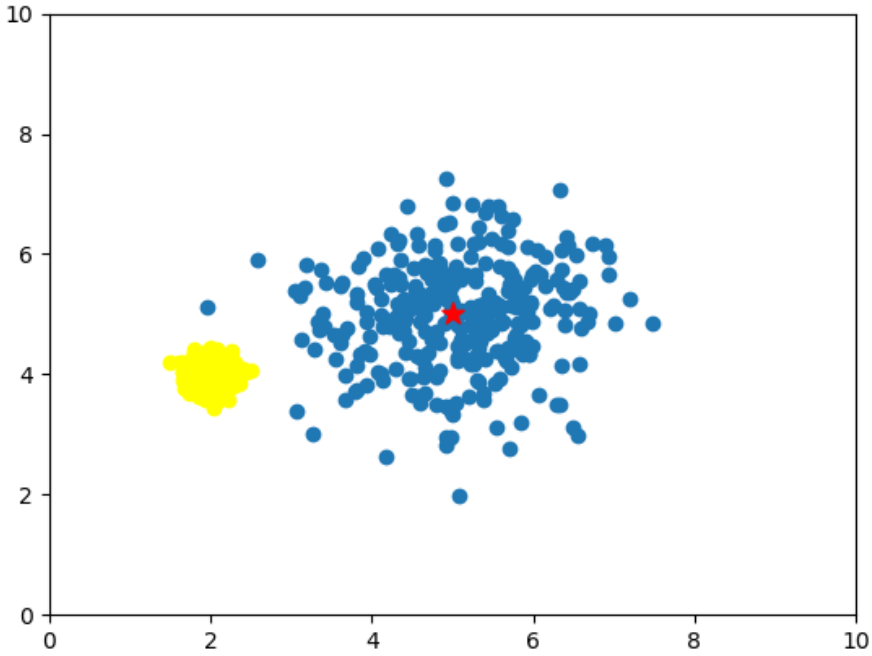

Рисунок 5 – Собранные данные

2. Алгоритм кластеризации искусственных иммунных сетей

2.1. Разработка алгоритма кластеризации искусственных иммунных сетей

Алгоритм кластеризации искусственных иммунных сетей представляет из себя модель машинного обучения, принцип обучения которой основан на самоорганизации в естественных иммунных сетях [4]. Процесс самоорганизации можно наблюдать в нервной и в иммунной системах, в них он характеризуется как необратимый процесс, происходящий под воздействием внешней среды, в результате взаимодействия множества элементов, что приводит к образованию новых более эффективных структур.

Самоорганизация в искусственных иммунных сетях характеризуется отсутствием явно выраженной целевой функцией, вместо нее здесь используются критерии распознавания кластеризуемых данных множеством генерируемых детекторов.

Искусственные иммунные сети позаимствовали у естественных не только процесс самоорганизации, но и ряд важных особенностей и терминов.

— Антитела (иммуноглобулины) – белки сыворотки крови, которые синтезируются в организме как проявление защитной реакции при попадании в него чужеродного вещества.

— Антиген – это вещество, которое организм считает чужеродным или потенциально опасным. Против антигена организм начинает вырабатывать собственные антитела.

— Аффинность (или родственность) – термодинамическая характеристика, количественно описывающая силу взаимодействия веществ (например, антигена и антитела).

— Антителообразование – процесс образования антител в ответ на появление антигенов.

— Соматическая гипермутация – молекулярный механизм, обеспечивающий разнообразие антител. В результате этого процесса в вариабельных участках генов иммуноглобулинов происходит множество

точечных мутаций.

Алгоритм кластеризации искусственных иммунных сетей делят на два этапа [5]:

1) генерация начальной популяции,

2) основной алгоритм.

Первый этап, генерация начальной популяции, включает в себя определение параметров обучения и создание начальной популяции антител. Антитело – набор параметров, которые образуются в сеть по общему признаку. Первый этап определяет следующие параметры.

— Порог клонального расширения ct – число, которое отсеивает антитела с высокой аффинностью. Такие антитела не будут участвовать в клональном отборе. В качестве начального значения выбрано $ct = 0{,}1$.

— Порог добавления клона в новую популяцию nt – число, которое отсеивает антитела с высокой аффинностью после клонирования и гипермутации. В качестве начального значения выбрано $nt = 0{,}1$.

— Коэффициент β – фактор умножения, один из множителей, который определяет количество клонов в популяции клонов. В качестве начального значения выбрано $\beta = 0{,}1$.

— Промежуток мутации – числовой промежуток для гипермутации популяции клонов. В качестве начального значения выбрано $mut = [-0{,}00001; 0{,}00001]$.

— Условие остановки – число, которое может являться максимальным числом антител в популяции, или количество циклов обучения. В качестве условия остановки выбрано максимальное число популяции антител равное 100000.

Второй этап – расширение популяции или алгоритм обучения (рисунок 6).

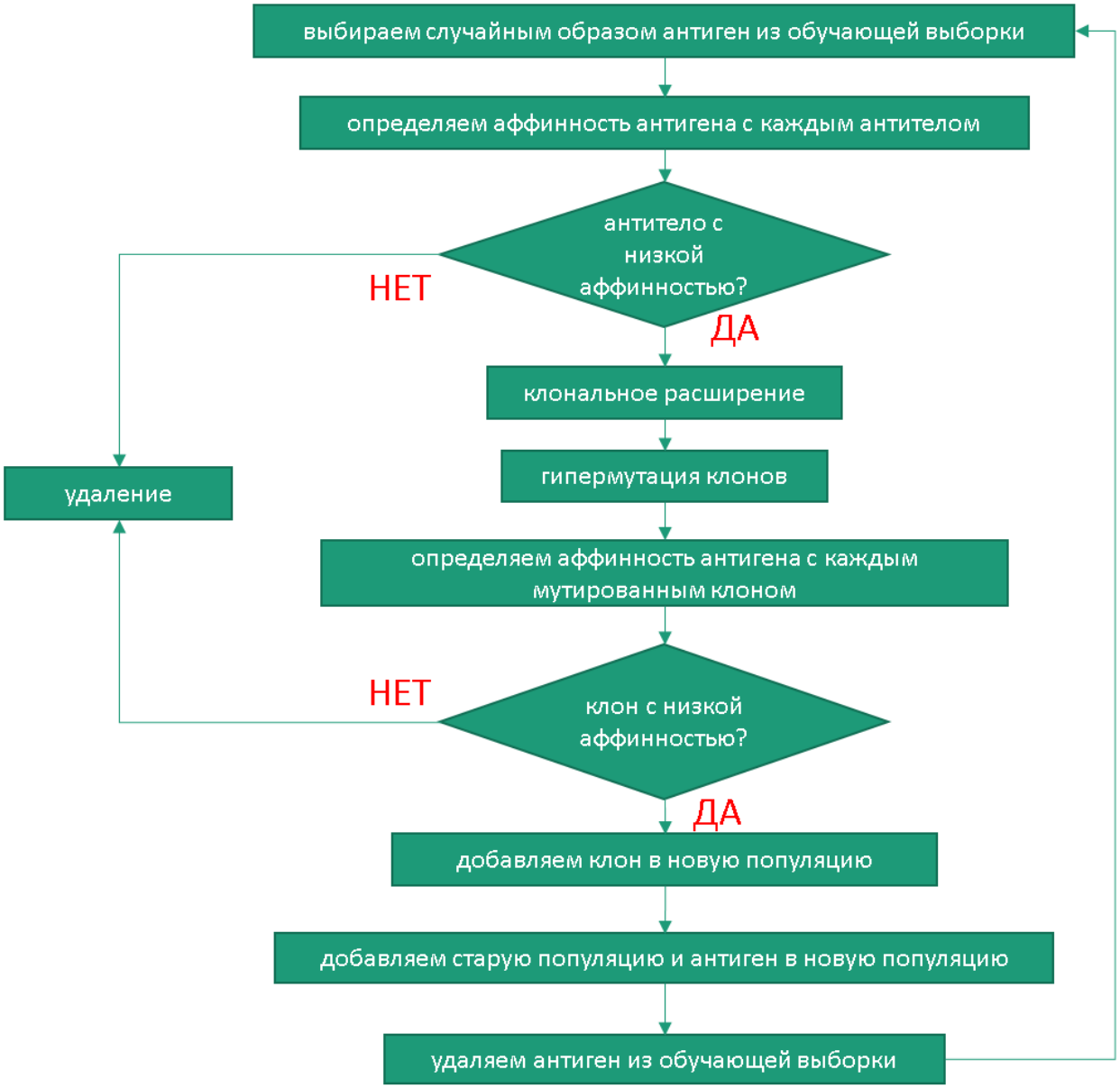

Рисунок 6 – Алгоритм обучения искусственных иммунных сетей

Антиген – набор параметров, на которых обучается модель.

Аффинность – мера взаимодействия (или сила связи) антигена и антитела или двух антител, которая формально может быть представлена в виде одной из метрик, указывающей на степень подобия или различия между соответствующими атрибутами строк. В данном алгоритме в качестве аффинности используется евклидово расстояние, вычисляющееся по следующей формуле:

$$Aff = \sqrt{\sum_{i=1}^{L}(Ab_i - Ag_i)^2}.$$

Клональное расширение – это процесс клонирования (воспроизведения) антител, который происходит в соответствии с аффинностью антитела и антигена по одной из двух формул:

$$C_i = round\left(\frac{\beta * N}{i}\right), \quad C_i = round\left(\frac{\beta * A}{i}\right)$$

где C_i – число клонов i-го антитела,

β – фактор умножения,

A – количество антител, которые прошли порог клонального расширения,

N – общее количество антител в популяции.

Первая формула вычисления аффинности не учитывает насколько далеко антитела с низкой аффинностью находятся от центра сети, что добавляет к начальной сети (рисунок 7) отдельную сеть (рисунок 8). В конечном счете это приведет к смещению центра сети антител (рисунок 9). Такая сеть, состоящая из множества «пучков», имеет место, например, если кластеризовать по лучшему совпадению.

Вторая формула учитывает, количество антител похожих на антиген, чем их больше, тем ближе к центру сети антител находится антиген, и, следовательно, тем больше будет создано клонов (рисунок 10). Такая формула применима, когда нет возможности хранить всю сеть антител (выделено мало памяти на хранение). Тогда можно хранить центр сети и кластеризовать данные по лучшему совпадению с ним.

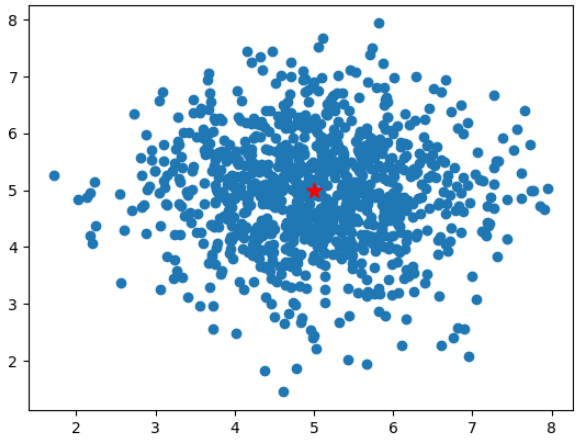

Рисунок 7 – Начальная популяция (сеть)

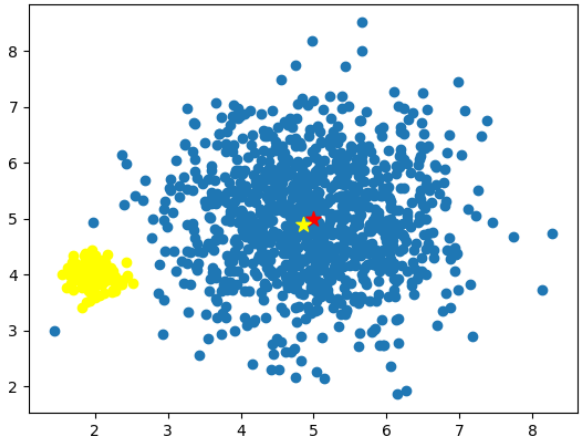

Рисунок 8 – Популяция после первого цикла обучения для 1 формулы

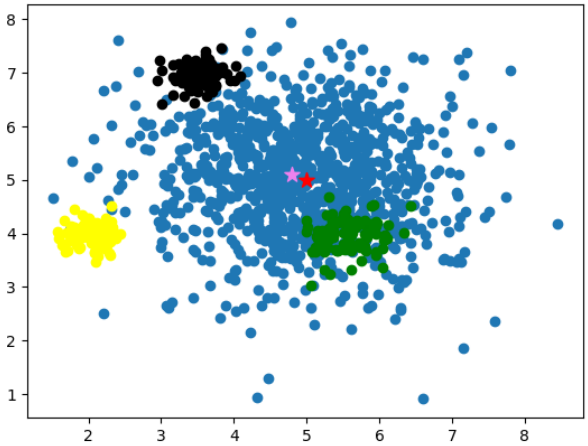

Рисунок 9 – Популяция после нескольких циклов обучения для 1 формулы

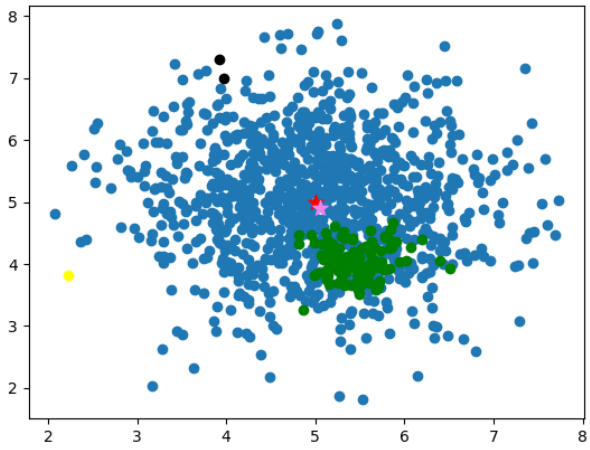

Рисунок 10 - Популяция после нескольких циклов обучения для 2 формулы

Гипермутация клонов – к каждому параметру клона антитела прибавляется случайное число из промежутка мутации. Гипермутация вычисляется по формуле:

$$U_{i_k} = U_{i_k} + \alpha,$$

где U_{i_k} – k-ый клон i-го антитела,

α – случайная величина, полученная из промежутка мутации.

Создание новой популяции может происходить по нескольким сценариям.

— Первый сценарий, добавление:

a) старой популяции;

b) клонов, которые прошли порог добавления;

c) самого антигена.

Антиген нужно добавить в случае, если выбрана вторая формула клонального расширения, так как в случае небольшого числа совпадения с антителами, антиген просто будет проигнорирован несмотря на то что антиген относится к этому кластеру.

— Второй сценарий, добавление:

a) только части старой популяции, например, добавление наиболее молодых антител из старой популяции;

b) клонов, которые прошли порог добавления;

c) самого антигена.

— Третий сценарий, добавление:

a) антител, которые пройдут порог аффинности центра сети клонов;

b) клонов, которые прошли порог добавления;

c) самого антигена.

Обучение модели по алгоритму искусственных иммунных сетей происходит пока не выполнится одно из условий остановки: превышено число антител в популяции или количество циклов обучения. В данной работе обучение модели проходило по первому сценарию.

2.2. Подбор гиперпараметров алгоритма искусственных иммунных сетей

Обычно алгоритм подбора гиперпараметров представляет из себя тестирование всевозможных комбинаций параметров и выбор лучшей комбинации [6]. Такой алгоритм хорошо работает, когда параметров алгоритма немного (2-3 параметра), число их возможных значений не более 10 и для тестирования одной комбинации затрачивается примерно 10 минут. В случае алгоритма искусственных иммунных сетей имеется 6 параметров [4]: N_0, Aff, β, mut, ct, nt, которые могут принимать большое количество значений, в среднем обучение занимает на одной комбинации занимает 10 минут.

Для данного алгоритма подбор гиперпараметров происходил последовательно согласно следующему алгоритму.

1) Выбор начальных параметров. В качестве начальных параметров были выбраны следующие значения, таблица 5.

Таблица 5 – Начальные гипермараметры

N_0	aff	β	mut	ct	nt
1000	0,6	0,1	-0,00001; 0,00001	0,1	0,1

где N_0 – число антител в начальной популяции,

aff – пороговая аффинность для тестирования сети,

β – фактор умножения,

$mut -$ промежуток мутации,

$ct -$ порог клонального расширения,

$nt -$ порог добавления клона в новую популяцию.

Если размер выборки меньше 1000 в качестве начального параметра числа антител использовалось значение равное $0{,}1 * N_V$, где $N_V -$ размер выборки.

2) Подбор числа антител в начальной популяции. В качестве тестируемых значений использовались следующие:

$$N_0 = \{10; 25; 50; 75; 100; 200; 300; 400; 500; 1000; 2500; 5000; 7500; 10000\}.$$

3) Подбор порога аффинности для тестирования сети. Для тестирования сети находят аффинность центра сети антител с новыми (тестовыми) данными, если это значение меньше порога аффинности, новые данные будем относить к данном кластеру (данной сети). В качестве тестируемых значений порога аффинности использовались следующие:

$$aff = \{0{,}1; 0{,}2; 0{,}25; 0{,}3; 0{,}4; 0{,}45; 0{,}5; 0{,}55; 0{,}6; 0{,}65; 0{,}7; 0{,}75; 0{,}8; 0{,}85\}.$$

4) Подбор фактора умножения. В качестве тестируемых значений использовались следующие:

$$\beta = \{0{,}1;\ 0{,}09;\ 0{,}08;\ 0{,}07; 0{,}06;\ 0{,}05;\ 0{,}04; 0{,}03\}.$$

5) Подбор промежутка мутации. В качестве тестируемых значений использовались следующие:

$$mut = \{[-0{,}03; 0{,}03]; [-0{,}01; 0{,}01]; [-0{,}005; 0{,}005]; [-0{,}001; 0{,}001];$$
$$[-0{,}0005;\ 0{,}0005]; [-0{,}0001;\ 0{,}0001]; [-0{,}00005;\ 0{,}00005];$$
$$[-0{,}00001; 0{,}00001]\}.$$

6) Подбор порога клонального расширения и порога добавления клона в новую популяцию. Тестировались всевозможные комбинации этих параметров:

$$ct = \{1{,}0;\ 0{,}75;\ 0{,}5;\ 0{,}25\},$$
$$nt = \{1{,}0;\ 0{,}75;\ 0{,}5;\ 0{,}25\}.$$

Каждая из выборок обучалась по 10 раз на всех комбинациях, это было сделано для поиска наиболее оптимальных неслучайных параметров.

В качестве критерия обучения была выбрана метрика $accuracy$ — доля правильных ответов алгоритма [7]:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN},$$

где TP – True Positive, верная классификация, когда алгоритм распознает объект класса;

TN – True Negative, верная классификация, когда алгоритм не относит объект к этому классу и тот действительно ему не принадлежит;

FP – False Positive, ошибка классификации, когда алгоритм относит объект к данному классу, хотя тот к нему не относится;

FN – False Negative, ошибка классификации, когда алгоритм не относит объект к данному классу, хотя тот к нему относится.

Для тестирования необходимая выборка была создана случайным образом из объектов данного класса и объектов, не принадлежащих данному классу, в соотношении 1:1. В каждом тестировании принимали участие 1000 объектов. Объекты, не принадлежащие данному классу, каждый раз перемешивались, после чего из них случайным образом выбиралось 500 записей.

Аналогичным образом, из не участвующих в обучении объектов данного класса выбиралось 500 записей. Если число записей, которые относятся к данному классу изначально меньше 500 или после обучения осталось меньше 500 записей, объекты используются повторно. В таком случае соотношение 1:1 сохраняется за счёт дробления вероятности. Для объектов, не принадлежащих данному классу, как и для объектов из этого класса выделяется 50% вероятности вне зависимости от количества записей. Это сделано для того, чтобы не отходить от главной цели – распознавание записей, принадлежащих данному классу.

В результате подбора гиперпараметров были получены следующие значения для каждой из выборок, таблица 6.

Таблица 6 - Результат подбора гиперпараметров

Выборка	N_0	aff	β	mut	ct	nt	p
Benign	200	0,7	0,08	$[-0{,}01; 0{,}01]$	0,075	1	0,68
Bot	300	0,125	0,04	$[-0{,}00005; 0{,}00005]$	0,5	1	0,66
Brute Force	100	0,2	0,07	$[-0{,}00001; 0{,}00001]$	0,1	1	0,9
DDoS	400	0,25	0,08	$[-0{,}00001; 0{,}00001]$	0,75	1	0,63
DoS GoldenEye	400	0,25	0,05	$[-0{,}01; 0{,}01]$	0,75	0,5	0,8
Dos Hulk	500	0,4	0,09	$[-0{,}03; 0{,}03]$	0,5	0,5	0,67
DoS Slowloris	300	0,7	0,03	$[-0{,}001; 0{,}001]$	1	0,25	0,7
FTP-Patator	500	0,45	0,1	$[-0{,}00005; 0{,}00005]$	0,25	0,75	0,71
Heartbleed	4	0,85	0,07	$[-0{,}005; 0{,}005]$	0,5	0,25	0,94
Infiltration	23	0,8	0,07	$[-0{,}00001; 0{,}00001]$	0,25	0,75	0,84
PortScan	200	0,8	0,08	$[-0{,}0005; 0{,}0005]$	0,5	0,75	0,89
SQL Injection	11	0,55	0,03	$[-0{,}01; 0{,}01]$	0,75	0,75	0,69
SSH-Patator	200	0,55	0,08	$[-0{,}01; 0{,}01]$	0,5	0,5	0,62
XSS	400	0,1	0,1	$[-0{,}0001; 0{,}0001]$	0,5	0,25	0,96

При подборе гиперпараметров для вычисления число клонов i-го антитела использовалась формула:

$$C_i = round\left(\frac{\beta * A}{i}\right),$$

где C_i – число клонов i-го антитела,

β – фактор умножения,

A – количество антител, которые прошли порог клонального расширения.

Среднее время подбора параметров для каждой из выборок 36 часов.

2.3. Обучение модели

Обучение на каждой из выборок проходило дважды, для каждой из формул числа клонов i-го антитела:

$$C_i = round\left(\frac{\beta * N}{i}\right), \quad C_i = round\left(\frac{\beta * A}{i}\right),$$

где C_i – число клонов i-го антитела,

β – фактор умножения,

A – количество антител, которые прошли порог клонального расширения,

N – общее количество антител в популяции.

В результате было получено 30 моделей, 15 моделей с одним центром (вторая формула) и 15 с несколькими центрами (первая формула), таблицы 7 и 8. Буквенные обозначения в таблице: $N_0 -$ число антител в начальной популяции, $ct -$ порог клонального расширения, $nt -$ порог добавления клона в новую популяцию, $TSC -$ число циклов обучения, $aff -$ пороговая аффинность для тестирования сети, $p -$ вероятность верно распознать данные, $C_1 -$ число элементов из 500 случайных записей, не содержащие элементов кластера и которые были верно распознаны, $C_2 -$ число элементов из 500 случайных записей данного кластера, которые были верно распознаны. Численные обозначения в таблице (по алфавиту): 1 – Benign, 2 – Bot, 3 – Brute Force, 4 – DDoS, 5 – DoS GoldenEye, 6 – Dos Hulk, 7 – DOS Slowhttptest, 8 – DoS Slowloris, 9 – FTP-Patator, 10 – Heartbleed, 11 – Infiltration, 12 – PortScan, 13 – SQL Injection, 14 – SSH-Patator, 15 – XSS.

Таблица 7 – Сети с несколькими центрами

	N_0	ct	nt	TSC	β	aff	N	p	C_1	C_2
1	200	0,075	1	32	0,08	0,7	115736	0,706	385	321
2	300	0,5	1	50	0,04	0,125	102031	0,799	493	306
3	100	0,1	1	32	0,07	0,2	137255	0,901	492	409

Продолжение таблицы 7

	N_0	ct	nt	TSC	β	aff	N	p	C_1	C_2
4	400	0,75	1	29	0,08	0,775	117944	0,622	171	451
5	400	0,75	0,5	39	0,05	0,25	122915	0,82	483	337
6	500	0,5	0,5	36	0,09	0,4	111514	0,703	498	205
7	500	0,5	1	18	0,08	0,4	109892	0,852	491	361
8	300	1	0,25	193	0,03	0,7	105593	0,723	368	355
9	500	0,25	0,75	20	0,1	0,7	110955	0,757	495	262
10	4	0,5	0,25	4	0,07	0,85	8	0,943	488	10
11	23	0,25	0,75	5	0,07	0,8	31	0,845	470	27
12	200	0,5	0,75	18	0,08	0,8	114885	0,903	408	495
13	11	0,75	0,75	5	0,03	0,55	16	0,701	415	12
14	200	0,5	0,5	32	0,08	0,55	103287	0,672	411	261
15	400	0,5	0,25	17	0,1	0,1	124576	0,966	499	467

Таблица 8 – Сети с одним центром

	N_0	ct	nt	TSC	β	aff	N	p	C_1	C_2
1	200	0,075	1	79	0,08	0,7	119492	0,711	401	310
2	300	0,5	1	83	0,04	0,125	114328	0,775	469	306
3	100	0,1	1	34	0,07	0,2	146291	0,911	495	416
4	400	0,75	1	62	0,08	0,775	122393	0,729	489	240
5	400	0,75	0,5	46	0,05	0,25	104252	0,835	481	354
6	500	0,5	0,5	104	0,09	0,4	128804	0,703	496	207
7	500	0,5	1	31	0,08	0,4	129179	0,854	488	366
8	300	1	0,25	208	0,03	0,7	104327	0,751	437	314

Продолжение таблицы 8

	N_0	ct	nt	TSC	β	aff	N	p	C_1	C_2
9	500	0,25	0,75	40	0,1	0,7	157976	0,756	495	261
10	4	0,5	0,25	4	0,07	0,85	8	0,943	488	10
11	23	0,25	0,75	5	0,07	0,8	28	0,853	478	27
12	200	0,5	0,75	21	0,08	0,8	157935	0,902	406	496
13	11	0,75	0,75	5	0,03	0,55	16	0,709	209	21
14	200	0,5	0,5	41	0,08	0,55	128726	0,667	386	281
15	400	0,5	0,25	15	0,1	0,1	153061	0,964	498	466

3. Тестирование алгоритма

3.1. Типы проведенных тестирований

Тестирование алгоритма можно поделить на два типа.

— Обучение одной сети для распознавания одного кластера. В таком случае для распознавания новых данных можно воспользоваться одним из трех типов или всеми тремя одновременно.

a) Если невозможно хранить всю сеть целиком, можно хранить только центр сети. В таком случае будем находить аффинность центра с новыми (тестовыми) данными, рисунок 11, если это значение меньше порогового, новые данные будем относить к данном кластеру.

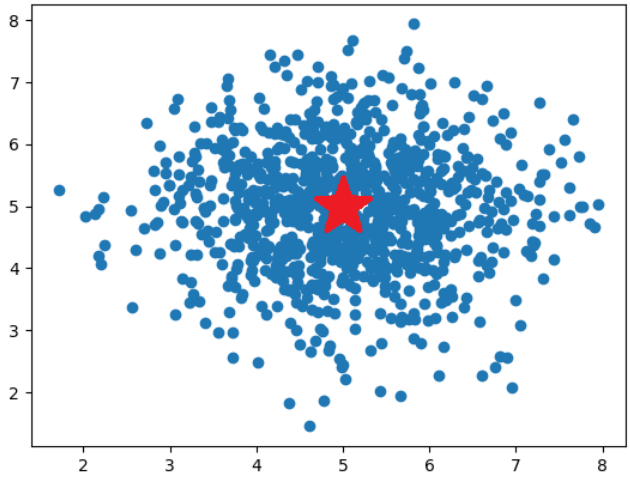

Рисунок 11 – Тестирование центра сети

b) При хранении всей сети целиком необходимо находить аффинность новых данных с каждым антителом и считать какое количество связей меньше порогового значения, рисунок 12. Если получаем большое количество связей, значит данные будем относить к данному кластеру.

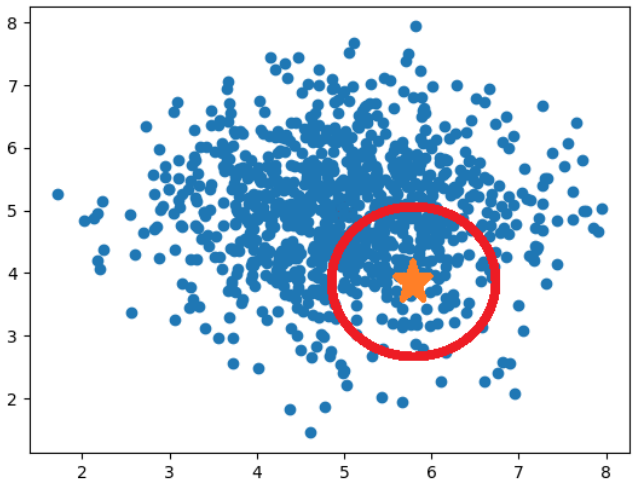

Рисунок 12 - Тестирование множества аффинностей

с) Храним всю сеть целиком. Значит, можно искать значение минимальной аффинности новых данных с антителом, находящимся в сети, рисунок 13. Если это значение меньше порогового, значит данные отнесем к этому кластеру.

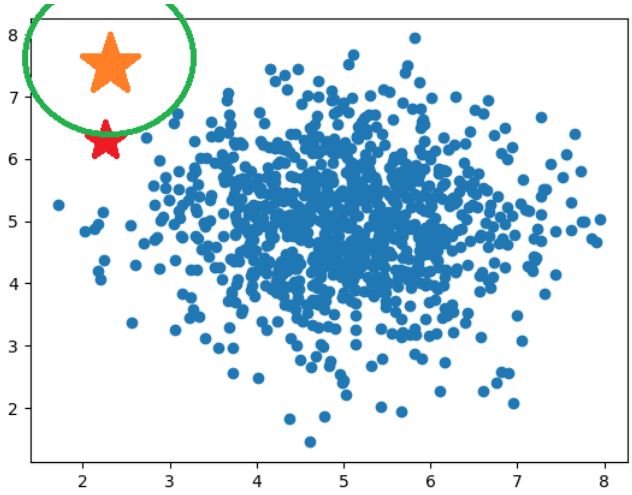

Рисунок 13 – Тестирование лучшей аффинности

— В случае, если обучили несколько моделей для распознавания нескольких кластеров, для распознавания новых данных можно использовать следующие три принципа.

а) Если невозможно хранить всю сеть целиком, достаточно хранить только центры всех сетей. В таком случае необходимо находить аффинность каждого центра с новыми (тестовых) данными, рисунок 11. Данные будем относить к кластеру с наименьшим значением аффинности центра.

b) При хранении всех сетей целиком нужно находить аффинность новых данных с каждым антителом и считать для каждой сети какое количество связей меньше порогового значения, рисунок 12. Данные будем относить к кластеру с наибольшим количеством антител, прошедшим пороговое значение. Для использования этого принципа необходимо, чтобы сети имели одинаковую размерность или чтобы для каждой сети находили среднее значение на минимальную размерность сети.

c) Хранение всей сети целиком. В таком случае можно искать значение минимальной аффинности новых данных с антителом, находящимся в сети, для каждой популяции, рисунок 13. Данные будем относить к кластеру с наименьшей аффинностью.

3.2. Тестирование каждой сети

В результате обучения было получено 30 моделей, 15 моделей с одним центром (вторая формула) и 15 с несколькими центрами (первая формула).

Исследование каждой модели происходило в три этапа:

1) исследовались только центры;

2) исследовалась вся выборка, происходил поиск антител с аффинностью меньше пороговой;

3) исследовалась вся выборка, происходил поиск антитела с лучшей аффинностью.

Целью исследования было нахождение вероятности того, что случайно выбранный элемент выборки при тестировании не будет отнесен к данному классу.

Результаты занесены в таблицы 9-14. В первом столбце записана выборка, на которой обучалась модель, в первой строчке записаны выборки на которых тестировалась модель. Обозначения в таблице (по алфавиту): 1 – Benign, 2 – Bot, 3 – Brute Force, 4 – DDoS, 5 – DoS GoldenEye, 6 – Dos Hulk, 7 – DOS Slowhttptest, 8 – DoS Slowloris, 9 – FTP-Patator, 10 – Heartbleed, 11 – Infiltration, 12 – PortScan, 13 – SQL Injection, 14 – SSH-Patator, 15 – XSS.

Таблица 9 – Модели только с 1 центром, исследовался только центр

	aff	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,7	0,376	0,628	0,905	0,619	0,846	0,703	0,93	0,985	0,998	1	0,917	1	0,571	0,508	0,965
2	0,125	0,971	0,396	1	1	1	1	1	1	1	1	1	0,23	1	1	1
3	0,2	0,981	1	0,195	1	1	1	0,966	0,992	1	1	1	0,994	1	1	0,072
4	0,25	0,994	1	0,997	0,552	0,47	0,938	1	0,999	1	1	1	0,999	0,952	1	0,994
5	0,25	0,967	1	0,996	0,595	0,335	0,939	0,999	0,999	1	1	1	0,999	0,524	1	0,989
6	0,4	0,997	1	1	0,926	1	0,619	1	1	1	1	1	1	1	1	1
7	0,4	0,966	0,999	0,954	1	0,994	1	0,298	0,676	1	1	0,917	1	1	1	0,975
8	0,7	0,858	0,999	0,902	1	0,553	0,998	0,268	0,372	0,499	1	0,833	0,998	0,429	0,498	0,968
9	0,45	0,981	1	1	1	1	1	0,873	0,895	0,504	1	0,889	1	1	0,998	1
10	0,85	0,996	1	1	0,858	1	0,616	1	1	1	0,091	0,861	1	1	1	1
11	0,8	0,972	1	1	0,91	1	0,632	0,92	0,973	1	1	0,25	1	1	1	1
12	0,8	0,798	0,377	0,095	0,546	0,285	0,937	0,454	0,652	0,499	1	0,889	0,001	0,429	0,495	0,035
13	0,55	0,354	0,311	0,853	0,54	0,658	0,733	0,941	0,667	0,497	1	0,889	0,155	0	0,003	0,956
14	0,55	0,809	0,383	0,095	0,593	0,286	0,938	0,454	0,659	0,499	1	0,917	0,071	0,429	0,495	0,035
15	0,1	0,987	1	0,195	1	1	1	0,966	0,995	1	1	1	0,994	1	1	0,072

Таблица 10 – Модели с несколькими центрами, исследовался только центр

	aff	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,7	0,372	0,628	0,905	0,622	0,849	0,706	0,93	0,985	0,998	1	0,917	1	0,571	0,508	0,965
2	0,125	0,996	0,4	1	1	1	1	1	1	1	1	1	0,821	1	1	1
3	0,2	0,981	1	0,195	1	1	1	0,966	0,992	1	1	1	0,994	1	1	0,072
4	0,775	0,296	0,253	0,805	0,119	0,129	0,282	0,925	0,659	0,497	0,909	0,833	0,14	0,	0,003	0,949
5	0,25	0,975	1	0,996	0,595	0,336	0,939	1	0,999	1	1	1	0,999	0,714	1	0,992
6	0,4	0,997	1	1	0,928	1	0,622	1	1	1	1	1	1	1	1	1

Продолжение таблицы 10

	aff	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	0,4	0,966	0,999	0,954	1	0,994	1	0,298	0,681	1	1	0,917	1	1	1	0,975
8	0,7	0,777	0,395	0,901	0,949	0,425	0,994	0,267	0,346	0,499	1	0,833	0,16	0,429	0,495	0,966
9	0,45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	0,85	0,997	1	1	0,862	1	0,617	1	1	1	0,091	0,861	1	1	1	1
11	0,8	0,965	1	1	0,919	1	0,638	0,886	0,87	0,522	1	0,25	1	1	0,998	1
12	0,8	0,798	0,377	0,095	0,546	0,285	0,937	0,454	0,652	0,499	1	0,889	0,012	0,429	0,495	0,035
13	0,55	0,85	0,39	0,902	0,709	0,429	0,945	0,995	0,682	0,499	1	0,972	0,12	0,429	0,495	0,989
14	0,55	0,85	0,39	0,902	0,783	0,516	0,953	0,995	0,679	0,499	1	0,944	0,122	0,429	0,495	0,991
15	0,1	0,987	1	0,195	1	1	1	0,966	0,995	1	1	1	0,994	1	1	0,072

Таблица 11 – Модели только с 1 центром, исследовалось множество аффинностей

	aff	p	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0.15	0,21	0,59	0,91	0,5	0,76	0,3	0,96	0,53	0,06	0,909	0,806	0,46	0,476	0,57	0,96
2	1	0.15	0,69	0,35	0,13	0,52	0,2	0,93	0,22	0,3	0,46	1	0,889	0	0,429	0,51	0,06
3	1	0.15	0,73	0,31	0,1	0,43	0,32	0,9	0,3	0,46	0,42	1	0,889	0,01	0,429	0,49	0,02
4	1	0.15	0,71	0,37	0,06	0,52	0,25	0,97	0,21	0,39	0,44	1	0,889	0	0,429	0,48	0,05
5	1	0.15	0,76	0,29	0,17	0,56	0,21	0,9	0,29	0,31	0,52	0,909	0,833	0	0,429	0,5	0,02
6	1	0.15	0,47	0,88	0,91	0,47	0,81	0,11	0,94	0,98	0,99	0,091	0,722	1	0,571	0,51	0,93
7	1	0.15	0,71	0,35	0,08	0,45	0,31	0,92	0,28	0,33	0,52	0,909	0,833	0,04	0,429	0,49	0,05
8	1	0.15	0,75	0,37	0,07	0,49	0,27	0,88	0,22	0,39	0,5	0,909	0,833	0	0,429	0,48	0,02
9	1	0.15	0,94	1	1	0,99	1	0,93	0,8	0,85	0,41	1	0,472	1	0,952	0,98	0,95
10	1	0.15	0,99	1	1	0,83	1	0,64	1	1	1	0,091	0,806	1	1	1	1
11	1	0.15	0,5	0,94	0,92	0,46	0,79	0,31	0,88	0,71	0,47	0,182	0,167	1	0,571	0,48	0,97

Продолжение таблицы 11

	aff	p	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
12	1	0.15	0,73	0,4	0,1	0,55	0,3	0,95	0,25	0,4	0,51	1	0,889	0	0,429	0,57	0,03
13	1	0.15	0,1	0	0	0,05	0,15	0,32	0,11	0,31	0,6	0,909	0,694	0	0	0	0
14	1	0.15	0,3	0,38	0	0,15	0,24	0,89	0,22	0,38	0,47	1	0,75	0	0	0,01	0
15	1	0.15	0,86	0,41	0,08	0,57	0,35	0,94	0,26	0,36	0,51	1	0,889	0,02	0,429	0,49	0,02

Таблица 12 – Модели с несколькими центрами, исследовалось множество аффинностей

	aff	p	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0.15	0,19	0,62	0,89	0,56	0,81	0,22	0,92	0,61	0,12	0,909	0,806	0,33	0,476	0,48	0,97
2	1	0.15	0,69	0,41	0,07	0,63	0,38	0,98	0,18	0,35	0,51	1	0,889	0	0,429	0,48	0,03
3	1	0.15	0,76	0,36	0,1	0,5	0,27	0,93	0,29	0,52	0,56	1	0,889	0,02	0,429	0,56	0,03
4	1	0.15	0,13	0	0	0	0,06	0,24	0,2	0,35	0,55	0,091	0,611	0	0	0,01	0
5	1	0.15	0,7	0,42	0,04	0,57	0,27	0,99	0,23	0,33	0,49	0,909	0,833	0	0,429	0,4	0,03
6	1	0.15	0,26	0,58	0,92	0,44	0,74	0,04	0,97	1	1	0,091	0,722	1	0,571	0,54	0,95
7	1	0.15	0,78	0,4	0,06	0,65	0,31	0,95	0,27	0,3	0,45	0,909	0,833	0,05	0,429	0,44	0,05
8	1	0.15	0,73	0,39	0,13	0,58	0,31	0,88	0,13	0,38	0,53	0,909	0,833	0	0,429	0,48	0,03
9	1	0.15	0,73	0,27	0,09	0,58	0,29	0,92	0,12	0,19	0,01	1	0,333	0	0,429	0,42	0,04
10	1	0.15	0,99	1	1	0,8	1	0,59	1	1	1	0,091	0,806	1	1	1	1
11	1	0.15	0,39	0,92	0,86	0,56	0,29	0,39	0,08	0,36	0	0,636	0	1	0,048	0,02	0,96
12	1	0.15	0,8	0,4	0,11	0,47	0,31	0,9	0,32	0,42	0,48	1	0,889	0	0,429	0,5	0,04
13	1	0.15	0,06	0	0	0,02	0,08	0,25	0,11	0,38	0,48	0,909	0,694	0	0	0	0
14	1	0.15	0,08	0	0	0,04	0,13	0,22	0,1	0,2	0,17	0,909	0,667	0	0	0	0
15	1	0.15	0,82	0,42	0,07	0,49	0,2	0,95	0,22	0,44	0,45	1	0,889	0,01	0,429	0,49	0,06

Таблица 13 – Модели только с 1 центром, исследовалась лучшая аффинность

	aff	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,05	0,19	0,08	0,05	0,63	0,69	0,75	0,93	0,59	0,18	1	0,833	0,2	0	0	0,03
2	0,05	0,73	0,02	0,9	0,86	1	0,91	1	0,99	1	1	0,944	0,69	0,571	0,48	0,97
3	0,05	0,59	0,98	0,	0,65	0,9	0,82	0,9	0,99	0,54	1	0,889	0,99	0,429	0,01	0,01
4	0,05	0,62	0,98	0,89	0,06	0,89	0,42	1	0,96	1	1	0,917	1	0,429	0,54	0,96
5	0,05	0,66	0,98	0,9	0,31	0	0,82	1	0,99	1	1	0,917	1	0,095	0,49	0,97
6	0,05	0,7	1	0,93	0,35	0,68	0,01	1	0,99	1	1	0,917	1	0,524	0,53	0,97
7	0,05	0,58	0,95	0,13	0,89	0,98	0,91	0,04	0,34	0,8	1	0,889	0,64	0,143	0,41	0,04
8	0,05	0,53	0,98	0,06	0,56	0,89	0,88	0,65	0,01	0,38	1	0,861	0,56	0,476	0,45	0,05
9	0,05	0,59	0,99	0,85	0,81	0,98	0,92	0,99	0,6	0	1	0,889	0,62	0,476	0,46	0,98
10	0,05	1	1	1	1	1	1	1	1	1	0,182	1	1	1	1	1
11	0,05	0,61	0,98	0,93	0,74	0,94	0,91	0,99	0,98	1	1	0,111	1	0,571	0,55	0,97
12	0,05	0,94	0,35	1	1	1	1	1	0,67	0,99	1	0,972	0,04	1	1	1
13	0,05	0,51	0,97	0,83	0,73	0,92	0,91	1	0,67	0,57	1	0,889	0,99	0	0,5	0,99
14	0,05	0,49	0,97	0,87	0,76	0,93	0,92	0,99	0,61	0,51	1	0,889	0,54	0,286	0	0,98
15	0,05	0,66	0,98	0,07	0,72	0,73	0,76	0,99	0,96	0,98	1	0,917	1	0,095	0,53	0

Таблица 14 – Модели с несколькими центрами, исследовалась лучшая аффинность

	aff	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,05	0,23	0,75	0,07	0,54	0,81	0,72	0,98	0,58	0,12	1,	0,889	0,26	0,048	0,01	0,01
2	0,05	0,62	0,02	0,89	0,81	0,98	0,92	1,	0,97	1,	1,	0,944	0,79	0,571	0,49	0,95
3	0,05	0,45	0,97	0,01	0,76	0,97	0,87	0,94	0,94	0,47	1,	0,889	0,99	0,238	0,05	0,02
4	0,05	0,52	0,99	0,91	0,03	0,9	0,51	1,	0,98	1,	1,	0,917	1,	0,429	0,46	0,93
5	0,05	0,6	0,97	0,83	0,37	0,02	0,83	0,46	1,	1,	1,	0,917	1,	0,095	0,22	0,98

Продолжение таблицы 14

	aff	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	0,05	0,51	0,97	0,9	0,44	0,77	0,03	1	0,97	1	1	0,917	1	0,429	0,5	0,95
7	0,05	0,95	1	0,18	1	1	1	0,02	0,36	0,81	1	0,972	0,55	0,905	1	0,11
8	0,05	0,56	0,98	0,07	0,81	0,96	0,85	0,78	0	0,39	1	0,889	0,5	0,476	0,48	0,06
9	0,05	0,94	1	0,95	1	1	1	1	0,59	0	1	0,972	0,6	0,905	1	1
10	0,05	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
11	0,05	0,57	0,97	0,87	0,69	1	0,91	1	0,66	0,55	1	0,083	0,65	0,476	0,5	0,98
12	0,05	0,97	0,48	1	0,9	0,99	1	0,99	0,65	1	1	0,972	0,02	1	1	1
13	0,05	0,49	0,98	0,87	0,82	0,93	0,97	0,99	0,71	0,49	1	0,889	1	0,048	0,51	1
14	0,05	0,6	0,95	0,87	0,82	0,96	0,88	0,99	0,71	0,48	1	0,889	0,58	0,143	0	0,98
15	0,05	0,57	0,99	0,14	0,69	0,81	0,93	0,93	1	1	1	0,917	0,98	0,19	0,44	0

В исследовании только центров модели проходило полное тестирование выборок, в остальных случаях из каждой выборки случайным образом выбиралось 100 или менее (в случае, если размер выборки меньше 100) элементов. Такое небольшое количество элементов связано с тем, что аффинность элемента выборки со всеми антителами модели (количество антител модели не менее 100000) считается около 1 секунды. Несложными подсчетами ($100 * 15 * 15$) можно определить, что на подсчет каждой из четырех таблиц ушло 22500 секунд или 375 минут.

Для исследования только центра выборки, таблицы 9 и 10, применялся порог аффинности aff, который относил или не относил элемент к данному классу.

Для исследования всей модели с множеством аффинностей, таблицы 11 и 12, применялась пороговая аффинность aff и отношение числа антител, которые прошли пороговую аффинность, ко всем антителам модели p.

Для исследования всей модели с поиском лучшей аффиности, таблицы 13 и 14, применялась пороговая аффинность aff, если лучшая аффинность

меньше пороговой, элемент относился к данному классу.

Для более точного тестирования необходим подбор параметров тестирования, но на самом деле алгоритмы тестирования построены таким образом, что нам не важно, как каждая из моделей отработает по отдельности, нас интересует, как модели отработают в совокупности. В этом случае тестирование не будет сильно зависеть от входных параметров (aff и p).

3.3. Тестирование совокупности сетей

Задача обнаружения вторжений не только в определении угрозы, то есть в том, что она существует, а еще в установке, что это за угроза и на что она нацелена [4].

Для определения конкретной угрозы необходимо, чтобы искусственные иммунные сети работали вместе. В этом случае антиген тестируется на каждой сети, а уже после определяется к какой сети он относится.

Результаты тестирования занесены в таблицы 15-20. В первом столбце записана выборка, на которой обучалась модель, во втором столбце записана вероятность p_1, что запрос будет распознан как атака, в третьем столбце записана вероятность p_2, что запрос будет верно отнесен к кластеру, в первой строчке записаны выборки на которых тестировалась модель, в ячейках таблицы занесено, сколько запросов отнесено к каждому из кластеров. Обозначения в таблице (по алфавиту): 1 – Benign, 2 – Bot, 3 – Brute Force, 4 – DDoS, 5 – DoS GoldenEye, 6 – Dos Hulk, 7 – DOS Slowhttptest, 8 – DoS Slowloris, 9 – FTP-Patator, 10 – Heartbleed, 11 – Infiltration, 12 – PortScan, 13 – SQL Injection, 14 – SSH-Patator, 15 – XSS.

Таблица 15 – Совокупность моделей с 1 центром, исследовался только центр

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,333	0,667	1515836	11644	13100	167	13427	44809	57774	11712 7	10 8287	8	16201	45494	0	281528	459188
2	0,628	0,036	728	70	0	0	0	0	2	0	0	0	0	1156	0	0	0
3	0,905	0	143	0	0	1	4	0	0	0	0	0	0	0	0	146	1213

Продолжение таблицы 15

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
4	0,622	0,237	48368	0	0	30366	27029	21136	0	0	0	336	0	0	0	790	0
5	0,839	0,365	1659	0	0	1347	3758	1268	31	7	0	0	0	0	0	2223	0
6	0,676	0,609	74488	0	124	3522	10706	140165	9	0	0	883	0	0	0	224	3
7	0,93	0,513	385	0	2	0	0	0	2821	1183	898	0	0	0	0	25	185
8	0,985	0,314	85	0	57	0	1	1	123	1819	701	0	1136	0	0	1843	30
9	0,998	0,497	17	0	0	0	0	0	0	0	3944	0	0	0	0	3973	1
10	1	0,909	0	0	0	0	0	1	0	0	0	10	0	0	0	0	0
11	0,917	0,444	3	0	0	0	0	7	2	3	4	0	16	0	0	1	0
12	1	0,611	65	14964	0	0	83	3	75	38	0	0	0	97038	0	45570	968
13	0,571	0	9	0	0	0	1	0	0	0	0	0	0	0	0	11	0
14	0,508	0,505	2904	0	0	0	0	0	0	2	14	0	0	0	0	2977	0
15	0,965	0,928	23	0	0	1	3	0	16	0	0	0	0	0	0	4	605

Таблица 16 – Совокупность моделей с несколькими центрами, исследовался только центр

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,335	0,665	151075 0	16935	13428	997	25589	48886	175234	377	10695 5	17	17536	305283	0	0	493333
2	0,628	0,011	728	22	0	0	0	0	2	0	0	0	0	1204	0	0	0
3	0,905	0	143	0	0	0	6	0	15	0	0	0	0	130	0	0	1213
4	0,636	0,013	46647	0	0	1707	57395	19403	0	0	0	2083	0	790	0	0	0
5	0,879	0,661	1245	0	0	69	6800	1613	42	0	0	0	0	524	0	0	0
6	0,677	0,598	74350	0	124	2	14233	1376 20	9	0	0	3564	0	219	0	0	3

Продолжение таблицы 16

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	0,93	0,718	385	0	1	0	2	0	3949	55	895	0	3	23	0	0	186
8	0,985	0,008	85	0	43	0	3	1	1889	48	611	0	1226	1841	0	0	49
9	0,998	0,497	17	0	0	0	0	0	0	0	3944	0	0	3973	0	0	1
10	1	0,909	0	0	0	1	0	0	0	0	0	10	0	0	0	0	0
11	0,917	0,472	3	0	0	0	0	7	5	0	3	0	17	1	0	0	0
12	1	0,863	64	20546	0	1	84	3	48	112	0	0	0	136978	0	0	968
13	0,571	0	9	0	0	0	1	0	0	0	0	0	0	11	0	0	0
14	0,508	0	2904	0	0	0	0	0	2	0	14	0	0	2977	0	0	0
15	0,965	0,928	23	0	0	0	5	0	16	0	0	0	0	3	0	0	605

Таблица 17 – Совокупность моделей с 1 центром, исследовалось множество аффинностей

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,38	0,62	62	0	9	0	0	4	0	0	5	0	0	16	0	4	0
2	0,61	0	39	0	4	0	0	0	0	0	0	0	0	57	0	0	0
3	0,93	0,9	7	0	90	0	0	0	0	0	0	0	0	3	0	0	0
4	0,6	0	40	0	42	0	0	18	0	0	0	0	0	0	0	0	0
5	0,97	0,01	3	0	77	0	1	17	0	0	0	0	2	0	0	0	0
6	0,64	0,62	36	0	2	0	0	62	0	0	0	0	0	0	0	0	0
7	0,94	0	6	0	77	0	0	0	0	0	16	0	0	0	0	0	1
8	0,99	0	1	0	30	0	0	1	0	0	11	0	15	42	0	0	0
9	1	0,42	0	0	57	0	0	0	0	0	42	0	0	1	0	0	0
10	1	0	0	0	0	0	0	10	0	1	0	0	0	0	0	0	0
11	0,917	0,056	3	0	3	0	0	7	0	0	18	0	2	1	0	2	0
12	1	1	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0

Продолжение таблицы 17

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	0,571	0	9	0	10	0	0	0	0	0	0	0	0	2	0	0	0
14	0,41	0	59	0	41	0	0	0	0	0	0	0	0	0	0	0	0
15	0,96	0	4	0	96	0	0	0	0	0	0	0	0	0	0	0	0

Таблица 18 – Совокупность моделей с несколькими центрами, исследовалось множество аффинностей

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,42	0,58	58	0	0	0	0	5	4	0	5	0	0	27	0	0	1
2	0,67	0	33	0	0	0	0	0	0	0	0	0	0	66	0	0	1
3	0,84	0	16	0	0	0	0	0	0	0	0	0	0	84	0	0	0
4	0,61	0	39	0	0	0	0	8	0	0	0	13	0	40	0	0	0
5	0,82	0,04	18	0	0	0	4	10	0	0	0	0	3	65	0	0	0
6	0,72	0,64	28	0	0	0	0	64	0	0	0	3	0	5	0	0	0
7	0,94	0,03	6	0	0	0	0	0	3	0	20	0	0	51	0	0	20
8	0,99	0	1	0	0	0	0	0	0	0	13	0	15	37	0	0	34
9	1	0,46	0	0	0	0	0	0	0	0	46	0	0	54	0	0	0
10	1	0,909	0	0	0	0	0	0	0	1	0	10	0	0	0	0	0
11	0,917	0,472	3	0	0	0	0	1	2	0	3	6	17	3	0	0	1
12	1	1	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
13	0,571	0	9	0	0	0	0	0	0	0	0	0	0	12	0	0	0
14	0,52	0	48	0	0	0	0	0	0	0	0	0	0	52	0	0	0
15	0,95	0	5	0	0	0	0	0	0	0	0	0	0	95	0	0	0

Таблица 19 – Совокупность моделей с 1 центром, исследовалась лучшая аффинность

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,18	0,82	82	2	0	1	6	1	0	0	1	0	1	3	1	2	0
2	1	1	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0,41	0	0	41	0	1	0	0	0	0	0	0	0	3	1	54
4	0,99	0,94	1	0	0	94	0	5	0	0	0	0	0	0	0	0	0
5	1	0,96	0	0	1	0	96	1	0	2	0	0	0	0	0	0	0
6	0,99	0,96	1	0	0	1	0	96	0	0	0	0	0	0	0	0	2
7	1	0,99	0	0	0	0	0	0	99	0	0	0	0	0	0	0	1
8	1	0,99	0	0	0	0	0	0	0	99	0	0	0	0	0	0	1
9	1	1	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
10	1	1	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0
11	0,972	0,944	1	0	0	0	0	0	0	0	1	0	34	0	0	0	0
12	0,97	0,94	3	0	0	0	0	0	2	0	0	0	0	94	0	1	0
13	1	0,905	0	0	2	0	0	0	0	0	0	0	0	0	19	0	0
14	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
15	0,99	0,89	1	0	9	0	0	0	1	0	0	0	0	0	0	0	89

Таблица 20 – Совокупность моделей с несколькими центрами, исследовалась лучшая аффинность

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,21	0,79	79	4	0	5	1	1	2	1	1	0	0	2	1	1	2
2	0,98	0,98	2	98	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0,99	0,41	1	0	41	0	0	0	0	0	0	0	0	0	4	1	53
4	1	0,99	0	0	0	99	0	1	0	0	0	0	0	0	0	0	0
5	1	0,99	0	0	0	0	99	1	0	0	0	0	0	0	0	0	0
6	1	0,98	0	0	0	0	0	98	0	0	0	0	0	0	0	0	2

Продолжение таблицы 20

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	1	0,99	0	0	0	0	0	0	99	1	0	0	0	0	0	0	0
8	0,99	0,97	1	0	0	0	0	0	2	97	0	0	0	0	0	0	0
9	1	0,99	0	0	0	0	0	0	0	0	99	0	0	0	0	1	0
10	1	1	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0
11	1	0,944	0	0	0	0	0	1	1	0	0	0	34	0	0	0	0
12	0,98	0,95	2	0	0	0	0	0	0	0	1	0	0	95	0	0	2
13	1	0,905	0	0	2	0	0	0	0	0	0	0	0	0	19	0	0
14	1	0,99	0	0	0	0	0	0	0	0	1	0	0	0	0	99	0
15	1	0,95	0	0	5	0	0	0	0	0	0	0	0	0	0	0	95

В исследовании только центров модели проходило полное тестирование выборок, в остальных случаях из каждой выборки случайным образом выбиралось 100 или менее (в случае, если размер выборки меньше 100) элементов. Такое небольшое количество элементов связано с тем, что аффинность элемента выборки со всеми антителами модели (количество антител модели не менее 100000) считается около 1 секунды. Несложными подсчетами ($100 * 15 * 15$) можно определить, что на подсчет каждой из четырех таблиц ушло 22500 секунд или 375 минут.

В тестировании только центра, таблицы 15 и 16, элемент относился к тому кластеру, аффинность центра с которым была наименьшая.

В тестировании всей выборки, таблицы 17 и 18, находился процент от сети антител, которые находился на аффинности меньше 1 по отношению к элементу. Элемент относился к тому кластеру, процент которого был наибольшим.

В тестировании по лучшему совпадению, таблицы 19 и 20, элемент относился к тому кластеру, аффинность анитела с которым была наименьшая.

3.4. Анализ результатов

Для анализа результатов также были протестированы в совокупности входные данные [4], как модели сетей.

Результаты тестирования занесены в таблицы 21 и 22. В первом столбце записана выборка, на которой обучалась модель, во втором столбце записана вероятность p_1, что запрос будет распознан как атака, в третьем столбце записана вероятность p_2, что запрос будет верно отнесен к кластеру, в первой строчке записаны выборки на которых тестировалась модель, в ячейках таблицы занесено, сколько запросов отнесено к каждому из кластеров. Обозначения в таблице (по алфавиту): 1 – Benign, 2 – Bot, 3 – Brute Force, 4 – DDoS, 5 – DoS GoldenEye, 6 – Dos Hulk, 7 – DOS Slowhttptest, 8 – DoS Slowloris, 9 – FTP-Patator, 10 – Heartbleed, 11 – Infiltration, 12 – PortScan, 13 – SQL Injection, 14 – SSH-Patator, 15 – XSS.

Таблица 21 – Совокупность входных данных, исследовался только центр

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,336	0,664	15090071	10318	12396	0	2654	78091	173619	0	79685	63	18231	332858	0	0	543334
2	0,628	0	728	0	0	0	0	0	2	0	0	0	0	1226	0	0	0
3	0,905	0	143	0	0	0	3	0	0	0	0	0	0	148	0	0	1213
4	0,633	0,001	469980	0	0	1338	20885	157753	0	0	0	5766	3	373000	0	0	0
5	0,886	0,253	1173	0	0	0	2600	1754	42	0	0	0	0	4724	0	0	0
6	0,711	0,575	664110	0	0	0	1980	132223	9	0	0	16601	302	124722	0	0	127
7	0,93	0,728	385	0	0	0	0	0	4004	0	613	0	285	25	0	0	187
8	0,985	0	85	0	0	0	0	1	1921	0	593	0	1244	1844	0	0	108
9	0,998	0,497	17	0	0	0	0	0	0	0	3944	0	0	3973	0	0	1

Продолжение таблицы 21

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	1	0,909	0	0	0	0	0	1	0	0	0	10	0	0	0	0	0
11	0,917	0,583	3	0	0	0	0	6	5	0	0	0	21	1	0	0	0
12	1	0,937	64	8889	0	0	43	4	44	67	0	0	0	148725	0	0	968
13	0,571	0	9	0	0	0	0	0	0	0	0	0	0	12	0	0	0
14	0,508	0	2904	0	0	0	0	0	2	0	14	0	0	2977	0	0	0
15	0,965	0,928	23	0	0	0	2	0	16	0	0	0	0	6	0	0	605

Таблица 22 – Совокупность входных данных, исследовалось множество аффинностей

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,38	0,62	62	5	0	0	0	6	2	0	4	0	2	18	0	0	1
2	0,6	0	40	0	0	0	0	0	0	0	0	0	0	59	0	0	1
3	0,93	0	7	0	0	0	0	0	0	0	0	0	0	93	0	0	0
4	0,65	0	35	0	0	0	0	0	0	0	0	17	0	48	0	0	0
5	0,93	0,02	7	0	0	0	2	18	0	0	0	0	0	73	0	0	0
6	0,71	0,25	29	0	0	0	0	25	0	0	0	40	0	6	0	0	0
7	0,95	0,01	5	0	0	0	0	0	1	0	3	0	13	59	0	0	19
8	0,98	0,05	2	0	0	0	0	0	1	5	0	0	32	37	0	0	23
9	0,99	0	1	0	0	0	0	0	0	0	0	0	48	51	0	0	0
10	1	0,909	0	0	0	0	0	0	0	1	0	10	0	0	0	0	0
11	0,917	0,556	3	0	0	0	0	1	2	0	0	6	20	3	0	0	1
12	1	1	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0

Продолжение таблицы 22

	p_1	p_2	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
13	0,571	0	9	0	0	0	0	0	0	0	0	0	0	12	0	0	0
14	0,58	0	42	0	0	0	0	0	0	0	0	0	0	58	0	0	0
15	0,98	0	2	0	0	0	0	0	0	0	0	0	0	98	0	0	0

Тестирование проходило аналогично совокупности искусственных иммунных сетей. Максимальная размерность модели была равна 100000. Тестировать лучшее совпадение аффинности нет смысла, потому что необученная модель просто хранит в себе входные параметры.

Среди искусственных иммунных сетей лучший результат синтеза сетей и метода тестирования показали: модели с несколькими центрами и метод по поиску лучшей аффинности, а также модели с одним центром и метод по поиску лучшей аффинности.

Модели с несколькими центрами и метод по поиску лучшей аффинности: вероятность модели распознать опасные и безопасные параметры – 0,982, вероятность модели отнести данные к верному кластеру – 0,922.

Модели с одним центром и метод по поиску лучшей аффинности: вероятность модели распознать опасные и безопасные параметры – 0,982, вероятность модели отнести данные к верному кластеру – 0,917.

Обе сети модели отработали отлично с большой вероятностью угадывания. В дальнейшей разработке рекомендовано более точно изучить какие параметры характеризуют каждую из выборок и провести анализ точности модели от количества антител обучения.

Метод тестирования «только центр» показал следующие результаты угадывания: 0,814 и 0,442 модели с одним центром, 0,818 и 0,423 модели с несколькими центрами, 0,82 и 0,406 модели данных.

Метод тестирования «множество аффинностей» показал следующие результаты угадывания: 0,811 и 0,242 модели с одним центром, 0,809 и 0,275

модели с несколькими центрами, 0,827 и 0,228 модели данных.

Метод тестирования «множество аффинностей» на практике показал самые плохие результаты.

4. Разработка приложения «Система обнаружения вторжений»

Приложение «Система обнаружения вторжений» представляет собой GUI приложение для решения задачи обнаружения компьютерных атак. По входным параметрам информационного потока приложение определяет наличие угрозы.

Разработанное приложение выглядит следующим образом, рисунок 14.

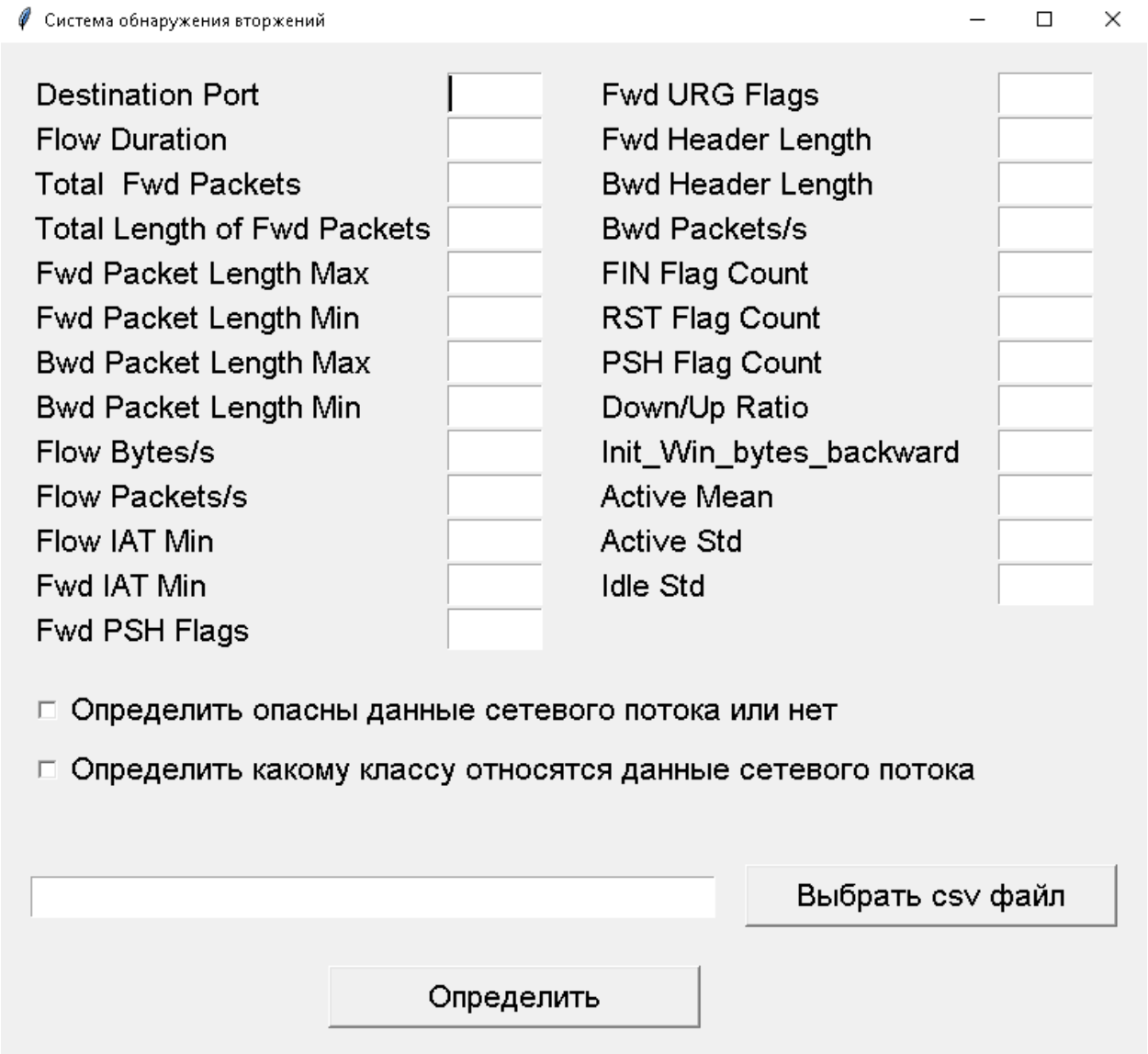

Рисунок 14 – «Система обнаружения вторжений»

По следующим входным параметрам система определяет наличие угрозы: «Destination Port», «Flow Duration», «Total Fwd Packets», «Total Length of Fwd Packets», «Fwd Packet Length Max», «Fwd Packet Length Min», «Bwd Packet Length Max», «Bwd Packet Length Min», «Flow Bytes/s», «Flow Packets/s», «Flow IAT Min», «Fwd IAT Min», «Fwd PSH Flags», «Fwd URG

Flags», «Fwd Header Length», «Bwd Header Length», «Bwd Packets/s», «FIN Flag Count», «RST Flag Count», «PSH Flag Count», «Down/Up Ratio», «Init Win bytes backward», «Active Mean», «Active Std», «Idle Std».

Можно для конкретного информационного потока определить наличие угрозы с вероятностью 0,982 и узнать к какому типы относятся данные с вероятностью 0,917. В результате тестирования при отсутствии атаки будет выведено следующее сообщение рисунок 15, которое можно сохранить в текстовый файл.

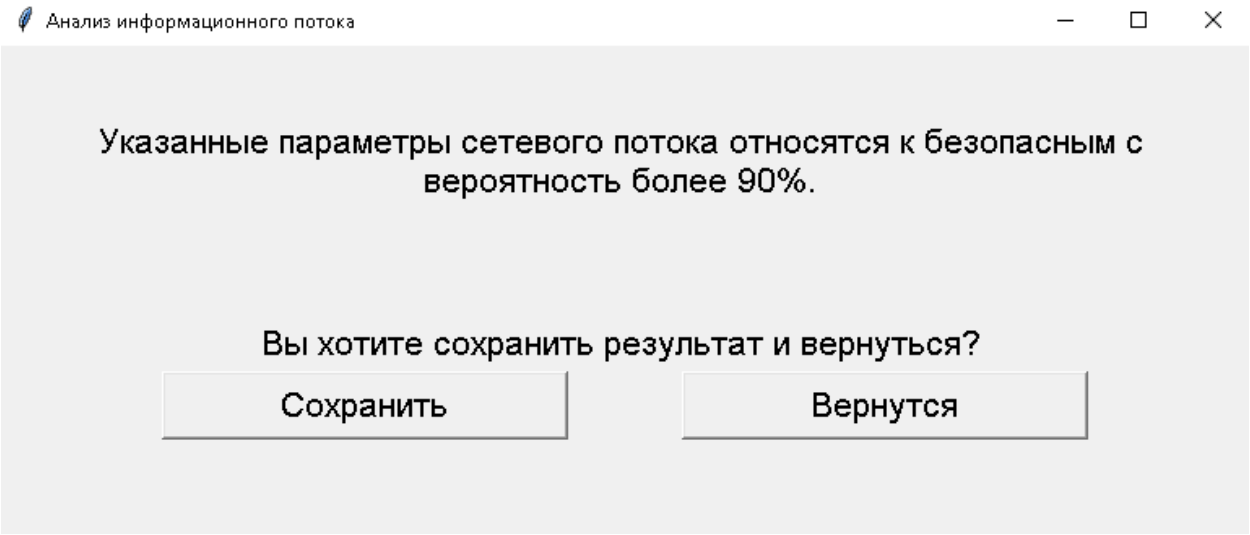

Рисунок 15 – Анализ информационного потока

Если система при тестировании обнаружит наличие атаки, будет выведено сообщение рисунок 16, которое также можно сохранить.

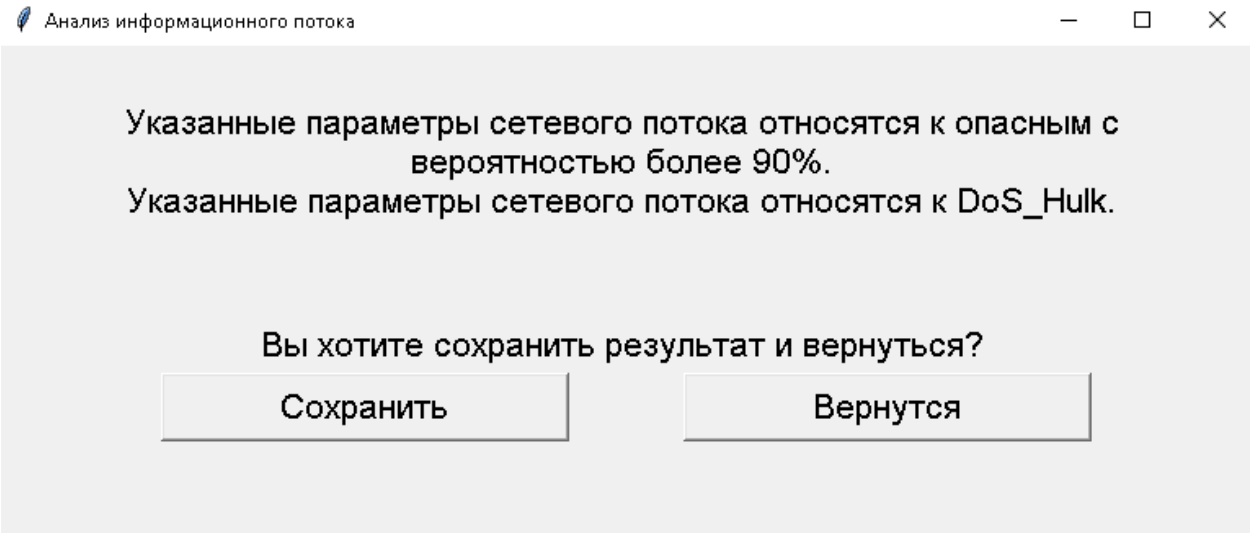

Рисунок 16 – Анализ опасного информационного потока

На анализ параметров сетевого потока уходит в среднем 12-13 секунд.

Также для удобства можно ввести параметры с помощью текстового файла формата csv. Тогда результат будет сохранен также в формате csv, рисунок 17 и 18.

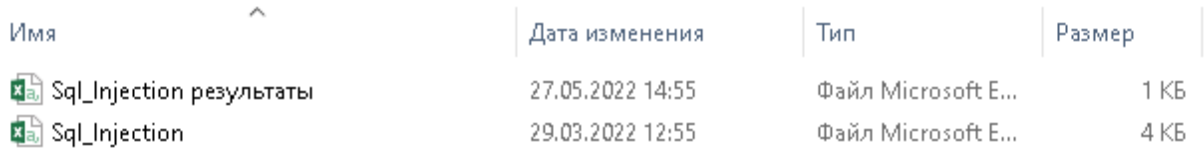

Рисунок 17 – Сохранение результатов

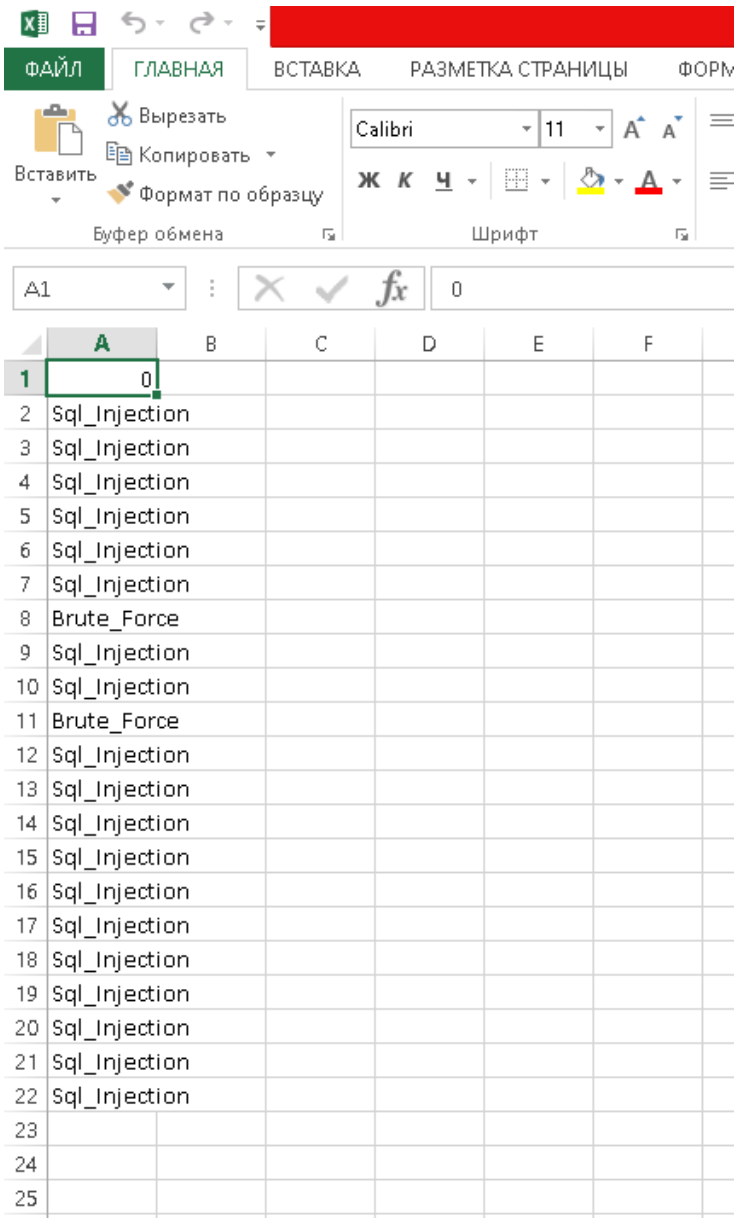

Рисунок 18 - Результаты

ЗАКЛЮЧЕНИЕ

В результате проделанной работы был разработан алгоритм искусственных иммунных сетей для решения задач обнаружения вторжений. В основу интеллектуального алгоритма лег процесс самоорганизации в биологических иммунных сетях позвоночных организмов. Выбор искусственных иммунных систем обусловлен тем, что они включают в себя преимущества генетических алгоритмов и нейронных сетей. К положительным свойствам искусственных иммунных сетей можно отнести адаптивность, динамическое обучение, отсутствие целевой функции, множество сгенерированных детекторов и гибкую размерность. В ходе работы были выявлены следующие недостатки алгоритма: сложность обучения; большое число гиперпараметров, для которых необходимо выполнить циклы подбора; нетривиальность операций гипермутации и клонального расширения.

В качестве методик тестирования алгоритма были выбраны: распознавание «по средним параметрам», распознавание «по нескольким аффинностям» и распознавание «по лучшей аффинности». Методики распознавания «по средним параметрам» и «по нескольким аффинностям» показали примерно один результат угадывания. Распознавание атак методом «по лучшей аффинности», когда каждое антитело выступает в роли детектора, показало наилучший результат. Вероятность детектирования угрозы этим методом – 0,982, вероятность верно распознать класс атаки – 92,2.

На основе метода тестирования «по лучшей аффинности» было разработано приложение для обнаружения атак.

В ходе проведения экспериментального исследования алгоритма искусственных иммунных сетей было установлено, что архитектура этого алгоритма может успешно применяться для обнаружения сетевых вторжений. Однако необходимо дальнейшее усовершенствование полученного алгоритма, для оптимизации работы и увеличения производительности.

Задачи выпускной квалификационной работы были выполнены, а цель была достигнута.

За период выпускной квалификационной работы были приобретены следующие компетенции, представленные в таблице 23.

Таблица 23 - Компетенции

Шифр компетенции	Расшифровка проверяемой компетенции	Расшифровка освоения компетенции
ОК-1	способность использовать основы философских знаний для формирования мировоззренческой позиции	способность формулировки целей и задач
ОК-2	способность использовать основы экономических знаний в различных сферах деятельности	способность оценки финансово-экономических рисков
ОК-3	способность анализировать основные этапы и закономерности исторического развития России, ее место и роль в современном мире для формирования гражданской позиции и развития патриотизма	знание исторических актов и умение их анализировать
ОК-4	способность использовать основы правовых знаний в различных сферах деятельности	способность применять знания о правовых актах в сфере ИБ
ОК-5	способность понимать социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности в области обеспечения информационной безопасности и защиты интересов личности, общества и государства, соблюдать нормы профессиональной этики	умение мотивировать себя для выполнения поставленной цели
ОК-6	способность работать в коллективе, толерантно воспринимая социальные, культурные и иные различия	умение понимать и выполнять выдвинутые требования руководителем
ОК-7	способность к коммуникации в устной и письменной формах на русском и иностранном языках для решения задач межличностного и межкультурного взаимодействия, в том числе в сфере профессиональной деятельности	умение находить общий язык с руководителем в устной и письменной формах
ОК-8	способность к самоорганизации и самообразованию	приобретен навык изучение нового ранее не изученного материала

Продолжение таблицы 23

Шифр компетенции	Расшифровка проверяемой компетенции	Расшифровка освоения компетенции
ОК-9	способность использовать методы и средства физической культуры для обеспечения полноценной социальной и профессиональной деятельности	умение разгружаться после работы по средствам физической культуры
ОПК-1	способность анализировать физические явления и процессы для решения профессиональных задач	способность анализа физических закономерностей материального мира
ОПК-2	способность применять соответствующий математический аппарат для решения профессиональных задач	приобретен навык оценки и обработки данных
ОПК-3	способность применять положения электротехники, электроники и схемотехники для решения профессиональных задач	умение использовать ПК для разработки ПО
ОПК-4	способность понимать значение информации в развитии современного общества, применять информационные технологии для поиска и обработки информации	знания в области машинного обучения и их применение в кластеризации данных
ОПК-5	способность использовать нормативные правовые акты в профессиональной деятельности	умение оформлять правовые документы в математической области
ОПК-6	способность применять приемы оказания первой помощи, методы и средства защиты персонала предприятия и населения в условиях чрезвычайных ситуаций, организовать мероприятия по охране труда и технике безопасности	умение соблюдать правила для безопасного использования ПК
ОПК-7	способность определять информационные ресурсы, подлежащие защите, угрозы безопасности информации и возможные пути их реализации на основе анализа структуры и содержания информационных процессов и особенностей функционирования объекта защиты	умение анализировать сетевой поток данных
ПК-1	способность выполнять работы по установке, настройке и обслуживанию программных, программно-аппаратных (в том числе криптографических) и технических средств защиты информации	приобретено умение работы со средой разработки PyCharm языка программирования Python

Продолжение таблицы 23

Шифр компетенции	Расшифровка проверяемой компетенции	Расшифровка освоения компетенции
ПК-2	способность применять программные средства системного, прикладного и специального назначения, инструментальные средства, языки и системы программирования для решения профессиональных задач	умение работать с библиотеками pandas, sklearn и math языка программирования Python
ПК-3	способность администрировать подсистемы информационной безопасности объекта защиты	умение разработки и администрирования системы обнаружения вторжений
ПК-4	способность участвовать в работах по реализации политики информационной безопасности, применять комплексный подход к обеспечению информационной безопасности объекта защиты	способность разрабатывать приложения для обеспечения ИБ
ПК-5	способность принимать участие в организации и сопровождении аттестации объекта информатизации по требованиям безопасности информации	способность тестирования приложений для задач ИБ
ПК-6	способность принимать участие в организации и проведении контрольных проверок работоспособности и эффективности применяемых программных, программно-аппаратных и технических средств защиты информации	умение тестировать разработанное приложение
ПК-7	способность проводить анализ исходных данных для проектирования подсистем и средств обеспечения информационной безопасности и участвовать в проведении технико-экономического обоснования соответствующих проектных решений	умение выделять основные аспекты задач для разработки приложения
ПК-8	способность оформлять рабочую техническую документацию с учетом действующих нормативных и методических документов	умение оформлять официальные документы и документацию
ПК-9	способность осуществлять подбор, изучение и обобщение научно-технической литературы, нормативных и методических материалов, составлять обзор по вопросам обеспечения информационной безопасности по профилю своей профессиональной деятельности	способность поиска, выбора и изучения новых материалов
ПК-10	способность проводить анализ информационной безопасности объектов и систем на соответствие требованиям стандартов в области информационной безопасности	способность анализа алгоритмов и приложений на соответствие требованиям

Продолжение таблицы 23

Шифр компетенции	Расшифровка проверяемой компетенции	Расшифровка освоения компетенции
ПК-11	способность проводить эксперименты по заданной методике, обработку, оценку погрешности и достоверности их результатов	приобретен навык проведения тестирования алгоритмов
ПК-12	способность принимать участие в проведении экспериментальных исследований системы защиты информации	приобретено умение работы со средой разработки PyCharm языка программирования Python
ПК-13	способность принимать участие в формировании, организовывать и поддерживать выполнение комплекса мер по обеспечению информационной безопасности, управлять процессом их реализации	умение формирования алгоритмов для задач ИБ
ПК-14	способность организовывать работу малого коллектива исполнителей в профессиональной деятельности	умение работать слаженно в команде со своим руководителем
ПК-15	способность организовать технологический процесс защиты информации ограниченного доступа в соответствии с нормативными правовыми актами и нормативными методическими документами Федеральной службы безопасности Российской Федерации, Федеральной службы по техническому и экспортному контролю	способность разработки ПО для защиты ИБ

СПИСОК ЛИТЕРАТУРЫ

1) Positive Technologies – URL: https://www.ptsecurity.com/ru-ru/ (дата обращения: 9.02.2022).

2) Набор данных для оценки обнаружения вторжений (CIC-IDS2017) – URL: https://www.unb.ca/cic/datasets/ids-2017.html (дата обращения: 19.02.2022).

3) Как самому разработать систему обнаружения компьютерных атак на основе машинного обучения – URL: https://habr.com/ru/post/538296/ (дата обращения: 11.03.2022).

4) Литвиненко В. И. Кластерный анализ данных на основе модифицированной иммунной сети // Управляющие системы и машины. – 2009. – № 1. – С. 54–85. (дата обращения: 21.02.2022).

5) Мельникова И. В. Исследование решения задачи идентификации рукописного почерка с применением модели искусственной иммунной сети aiNET // Современные проблемы горно-металлургического комплекса. Наука и производство / ред. Ю. И. Еременко, Е. В. Ильичева, Л.Н. Крахт, А. А. Кожухов, А. В. Макаров, М. С. Демьяненко. – Старый Оскол, 2017. – С. 57–63. (дата обращения: 14.03.2022).

6) Разинков Е. В. Лекции по машинному обучению и компьютерному зрению – URL: https://www.youtube.com/channel/UCcY6LFZNgZHR2skk4K_PKw (дата обращения: 10.02.2022).

7) Метрики в задачах машинного обучения – URL: https://habr.com/ru/company/ods/blog/328372/ (дата обращения: 18.05.2022).

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1. Алгоритм искусственных иммунных сетей

```
import csv
import numpy as np
import pandas as pd
from sklearn import preprocessing
import math
import docx
import os
import datetime
import time
import matplotlib.pyplot as plt

# аффиность антиген-антитело
def affinity(Ab, Ag):
    Ab = (Ab - Ag) ** 2
    D = math.sqrt(np.sum(Ab))
    return D

def aiNET(C_initial, a, ct, nt, B, mut):
    c = len(C_initial)
    Aff = np.array([])

    # вычисляем аффиность Ab-Ag
    for j in range(c):
        Aff = np.append(Aff, affinity(a, C_initial[j]))

    C_help = C_initial
    len_C = len(C_help)
    Aff = np.sort(Aff)
    C_help = C_help[Aff.argsort()]
    C_new = np.zeros((1, 25))
    Al = Aff <= ct
    Al = Al[Al == 1]
    B = (B * len(Al)) / len_C

    new_pop = np.array([int((B * len_C) / (i + 1)) for i in
range(len(Al))])
    new_pop = new_pop[new_pop != 0]

    for i in range(len(new_pop)):
        D = np.random.uniform(low=mut[0], high=mut[1],
size=(new_pop[i], 25))
        R = np.repeat(np.array([C_help[i]]), new_pop[i], axis=0)
        C_new = np.vstack((C_new, R + D))

    C_new = C_new[1:]

    # вычисляем аффиность клонов и антигена, для дальнейшего
отсеивания
    c = len(C_new)
    Aff = np.array([])
```

```
    # вычисляем аффиность Ab-Ag
    for j in range(c):
        Aff = np.append(Aff, affinity(a, C_new[j]))

    C_help = C_new

    # добавлем в новую популяцию только клонов с аффиностью меньше
пороговой
    C_new = C_help[Aff < nt]

    # добавляем клонов в старую популяци
    if len(C_new) != 0:
        C_initial = np.vstack((C_initial, C_new))

    # добавляем также антиген
    C_initial = np.vstack((C_initial, a))

    return C_initial
```

ПРИЛОЖЕНИЕ 2. Обработка данных

```
# нормализуем данные [0, 1]
def data_normalization(data):
    scaler = preprocessing.MinMaxScaler()

    names = data.columns
    d = scaler.fit_transform(data)

    scaled_df = pd.DataFrame(d, columns=names)
    return scaled_df

# приводим данные к одному типу, чтобы их потом нормализовать
def clean_dataset(df):
    assert isinstance(df, pd.DataFrame), "df needs to be a
pd.DataFrame"
    df.dropna(inplace=True)
    indices_to_keep = ~df.isin([np.nan, np.inf, -np.inf]).any(1)
    return df[indices_to_keep].astype(np.float64)

# корреляция данных, ищем пары столбцов с высокой корреляцией и
удаляем один из них
def sample_cleaning():
    directory = '/Users/User/Desktop/ДИПЛОМ/MachineLearningCVE'
    files = os.listdir(directory)
    T = pd.DataFrame()

    for i in range(len(files)):
        print(i + 1)
        data = pd.read_csv('MachineLearningCVE/' + files[i])

        T = pd.concat([T, data])
    del T[' Label']
```

```
    T = clean_dataset(T)
    Q = []
    M = T.to_numpy()

    for i in range(len(M[0])):
        if np.sum(M[:, i]) == 0:
            Q.append(i)

    M = np.delete(M, Q, axis=1)
    Mat = np.transpose(M)
    C = np.corrcoef(Mat)

    f = open('корреляция.txt', 'w')
    f.write(str(C))
    f.close()

    K = np.ones((len(C)), dtype=int)
    i = 0
    while i < (len(C) - 1):
        j = i + 1
        while j < len(C) and K[i] != 0:
            if abs(C[i, j]) >= 0.5:
                K[j] = 0
            j += 1
        i += 1

    K = list(K)
    for i in range(len(Q)):
        K.insert(Q[i], 0)
    return K
```

ПРИЛОЖЕНИЕ 3. Циклы обучения

```
# Циклы обучения
def main_sup(np_attack, np_no_attack, N, ct, nt, TSC, B, mut, M,
aff):

    # перемешиваем обучающую выборку
    ind = np.arange(len(np_attack))
    np.random.shuffle(ind)
    np_attack = np_attack[ind]

    ind = np.arange(len(np_no_attack))
    np.random.shuffle(ind)
    np_no_attack = np_no_attack[ind]

    no_train = np_no_attack[:int(len(np_no_attack) * 0.8)]

    yes_test = np_attack
    no_test = np_no_attack[int(len(np_no_attack) * 0.8):]

    # создаем начальные популяции
    C_no = no_train[:N]
    no_train = np.delete(no_train, range(N), axis=0)
```

```
    i = 0
    while len(C_no) < 100000 and i < len(no_train):
        C_no = aiNET(C_no, no_train[i], ct, nt, B, mut)
        print(f"Шаг {i + 1}")
        print(f"Норм попул {len(C_no)}")
        i += 1
    TSC = i

    # нахоим центр антител
    c_mid = np.sum(C_no, axis=0) / len(C_no)


    if len(no_test) >= M // 2:
        a = np.vstack((yes_test[:M // 2],  no_test[:M // 2]))
    elif len(np_no_attack) >= M // 2:
        a = np.vstack((yes_test[:M // 2], np_no_attack[:M // 2]))
    else:
        a = np.vstack((yes_test[:M // 2], np_no_attack))


    a = np.array([affinity(i, c_mid) for i in a])
    A_no = np.zeros(len(a))
    f = list(a[:M // 2] > aff)
    f = f + list(a[M // 2:] <= aff)
    A_no[f] = 1

    return C_no, np.sum(A_no[:M // 2]) / 1000 + ((np.sum(A_no[M //
2:])   /   len(A_no[M   //   2:])))   /   2,   len(C_no),   c_mid,
np.sum(A_no[:int(M // 2)]), np.sum(A_no[int(M // 2):]), TSC
```

ПРИЛОЖЕНИЕ 4. Методы тестирования

```
# исследуем только центр
def Test1(attack, Q):
    c_mid = [np.sum(i, axis=0) / len(i) for i in attack]

    F = np.zeros(15)

    for i in range(len(Q)):
        print(i)
        a = np.array([affinity(Q[i], j) for j in c_mid])
        f = np.argmin(a)
        F[f] += 1

    return F

# исследуем множество аффинностей
def Test2(attack, Q):
    ind = np.arange(len(Q))
    np.random.shuffle(ind)
    Q = Q[ind]
    F = np.zeros(15)
```

```
    i = 0
    while i < 100 and i < len(Q):
        print(i)
        T = np.zeros(len(attack))
        for k in range(len(attack)):
            a = np.array([affinity(j, Q[i]) for j in attack[k]])

            A_no = np.zeros(len(a))
            f = list(a <= 1)
            A_no[f] = 1
            p = np.sum(A_no) / len(A_no)

            T[k] = p

        f = np.argmax(T)
        F[f] += 1
        i += 1

    return F

# исследуем лучшую аффинность
def Test3(attack, Q):
    ind = np.arange(len(Q))
    np.random.shuffle(ind)
    Q = Q[ind]
    F = np.zeros(15)

    i = 0
    while i < 100 and i < len(Q):
        print(i)
        T = np.zeros(len(attack))
        for k in range(len(attack)):
            a = np.array([affinity(j, Q[i]) for j in attack[k]])

            T[k] = np.min(a)

        f = np.argmin(T)
        F[f] += 1

        i += 1

    return F
```