Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук
Кафедра компьютерной безопасности

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ООП
канд. техн. наук, доцент

_____ В.Н. Тренькаев
подпись
«25» 01 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА СПЕЦИАЛИСТА (ДИПЛОМНАЯ РАБОТА)

ИНТЕЛЛЕКТУАЛЬНЫЙ ТЕЛЕГРАМ-БОТ ПО КОМПЬЮТЕРНОЙ БЕЗОПАСНОСТИ

по специальности 10.05.01 Компьютерная безопасность,
специализация (профиль) «Анализ безопасности компьютерных систем»

Белая Наталья Александровна

Руководитель ВКР
канд. физ.-мат. наук, доцент

_____ С.И. Самохина
подпись
«25» 01 2023 г.

Автор работы
студент группы № 931724

_____ Н.А. Белая
подпись
«25» 01 2023 г.

Томск – 2023

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Руководитель ООП
канд. техн. наук, доцент

_____ В.Н. Тренькаев
подпись
« 08 » июня 2022 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы специалиста обучающемуся
Белой Наталье Александровне, 931724
Фамилия Имя Отчество обучающегося
по специальности 10.05.01 Компьютерная безопасность, специализация (профиль)
«Анализ безопасности компьютерных систем»

1 Тема выпускной квалификационной работы
«Интеллектуальный телеграм-бот по компьютерной безопасности»

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:
а) в учебный офис / деканат – 16.01.2023 б) в ГЭК – 27.01.2023

3 Исходные данные к работе:
Объект исследования – подготовка студентов к CTF соревнованиям
Предмет исследования – интеллектуальный поиск по материалам подготовки к CTF соревнованиям
Цель исследования – Разработать телеграм-бота, выполняющего запросы по категориям задач CTF

Задачи:
1. Изучить существующие поисковые боты;
2. Разработать принцип работы бота;
3. Изучить существующие подходы к реализации нейронных сетей и существующие платформы для ботов;
4. Реализовать бота;

Методы исследования:
обзор существующих методов хранения и поиска, разработка поискового бота
Организация или отрасль, по тематике которой выполняется работа, –
Работа выполняется по заданию кафедры компьютерной безопасности ТГУ

4 Краткое содержание работы
Разработка и реализация поискового бота, предназначенного для подготовки студентов к участию в CTF соревнованиях

Научный руководитель выпускной квалификационной работы
канд. физ.-мат. наук, доцент _____ / С.И. Самохина
должность, место работы *подпись* *И.О. Фамилия*

Задание принял к исполнению
студент группы № 931724 _____ / Н.А. Белая
должность, место работы *подпись* *И.О. Фамилия*

АННОТАЦИЯ

Актуальность: Участие в CTF соревнованиях способствует хорошей профессиональной подготовке молодых ИБ-специалистов. На нашей кафедре также есть команда, участвующая в CTF соревнованиях, которая регулярно принимает в свои ряды поступивших на первый курс студентов. Новичку приходится справляться с большим объемом информации, который необходимо как-то структурировать и не терять полезные ссылки. На сегодняшний день актуальным решением проблемы «хранить всё в одном месте, чтобы потом легко можно было найти» является поисковый бот. Он полезен в случае, если искомая информация может иметь разные наименования и осуществлять поиск необходимо по всем. С целью помощи новичкам в CTF соревнованиях был разработан поисковый бот.

Объект исследования: подготовка студентов к CTF соревнованиям

Предмет исследования: интеллектуальный поиск по материалам подготовки к CTF соревнованиям

Цель работы: разработать телеграм-бота, выполняющего запросы по категориям задач CTF

Задачи работы:

- изучить существующие поисковые боты;
- разработать принцип работы бота;
- изучить существующие подходы к реализации нейронных сетей и существующие платформы для ботов;
- реализовать бота.

Результат: Разработан поисковый интеллектуальный телеграм-бот, предназначенный для поиска по категориям CTF соревнований

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 3
1 Обзор существующих систем хранения и поиска 5
1.1 Системы, которыми сейчас пользуются CTF команды 5
1.2 Решения, используемые в других предметных областях 7
2 Нейронные сети 8
3 Выбор инструментов реализации обучающей программы 12
3.1 Язык программирования Python 12
3.2 Telegram Bot API 12
3.3 Библиотека scikit-learn (sklearn) 12
4 Создание интеллектуального бота 13
4.1 Схема работы интеллектуального бота 13
4.2 Подготовка набора данных для обучения нейронной сети 15
4.3 Обучение нейронной сети 16
4.4 Примеры работы телеграм-бота 17
ЗАКЛЮЧЕНИЕ 20
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ 21

ВВЕДЕНИЕ

Capture the Flag или CTF в ИБ — это соревнования в форме командной игры, главная цель которой — захватить «флаг» у соперника в приближенных к реальности условиям. Команды решают прикладные задачи, чтобы получить уникальную комбинацию символов (флаг). Далее участники отправляют флаг в специальную платформу и получают подтверждение, что задача решена верно или стоит попытаться дать ответ ещё раз [1].

Участие в CTF соревнованиях способствует хорошей профессиональной подготовке молодых ИБ-специалистов [2], а успешное участие в таких соревнованиях также помогает найти работу по специальности [3]. Чтобы успешно участвовать в CTF, необходимо иметь обширные знания по существующим уязвимостям в ПО и методам их эксплуатации, и наоборот, решение заданий помогает закрепить изученное на практике.

По интернету разбросано множество ресурсов, где студенты, которые интересуются CTF, могут найти тренировочные задания. Но, конечно, при этом не может не возникать мыслей, что было бы хорошо собрать и хранить в одном месте все задания и полезные материалы, и при этом хранить так, чтобы среди всего собранного легко отыскивалось то, что нужно.

Каждый студент или команда решают эту задачу по-своему. Самым простым решением будет просто добавлять все полезные ссылки в закладки браузера, однако, когда их становится слишком много, то проглядывать их становится неудобно. Можно создать список ссылок, где каждая ссылка подписана (например, в Google Документах или Notion) и потом пользоваться простым поиском на совпадение, но это неудобно, если искомая категория имеет несколько названий (человек ищет категорию заданий “Forensic”, а тот, кто подписывал ссылку, использовал русское “форенсика”). Тем более поиск на совпадение неудобен, если человек допустил ошибку в слове или ищет по смыслу связанное слово (ищет “RSA” вместо “криптография”). Современным решением проблемы является интеллектуальный поиск.

Интеллектуальный поиск (поиск на основе нейронной сети) подразумевает, что при достаточном количестве материала для обучения по запросу пользователя будет выдаваться релевантный ответ, даже если запрос не был точным: содержал ошибки, использовал синонимы или может быть только логически отнесён к какой-либо категории. В качестве интерфейса для запроса часто используется бот.

Бот – это специальная программа, выполняющая различные функции и упрощающая жизнь их пользователей. Главной задачей бота является автоматический ответ после введенной ему пользователем команды. В данном случае пользователь сможет написать,

какой вид заданий он хочет найти, а бот в ответ пришлёт ему искомые ссылки [4]. Также в отличие от веб-документа в бота можно добавить и дополнительные функции: искать дату следующего соревнования, объяснять термины.

Цель дипломной работы - разработать телеграм-бота, выполняющего запросы по категориям задач CTF.

Работа содержит 22 страницы. Состоит из введения, обзора существующих методов хранения и поиска, теоретических основ нейронных сетей, выбора инструментов реализации бота, описания реализации бота, заключения и приложения со ссылкой на репозиторий.

1 Обзор существующих систем хранения и поиска

1.1 Системы, которыми сейчас пользуются CTF команды

Самым простым способом хранить в одном месте полезные ссылки – это составить список ссылок и выложить его в открытый доступ. Например, обсуждение ВКонтакте (рисунок 1) [5].

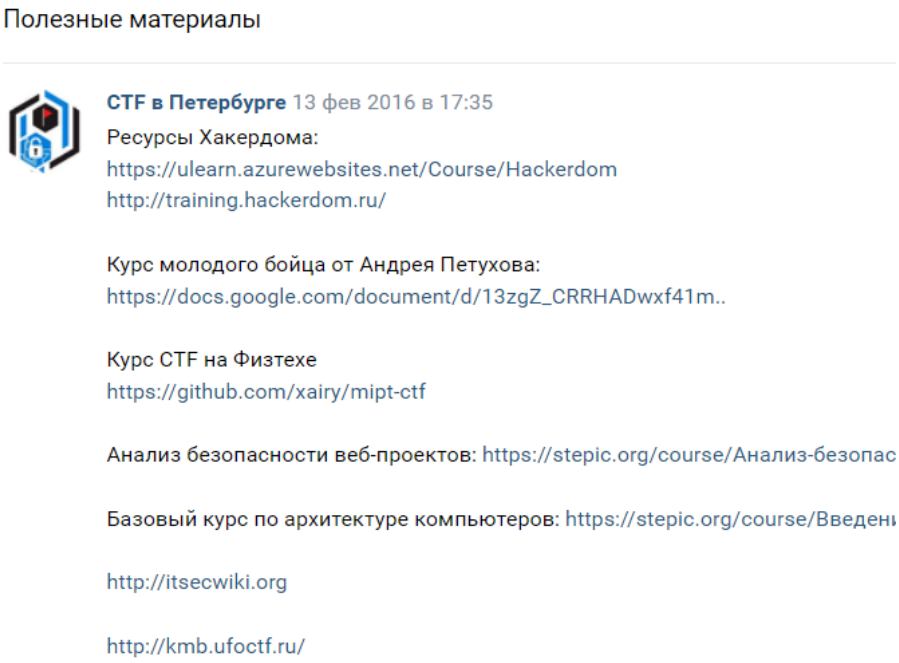


Рисунок 1 – Пример списка в обсуждениях ВКонтакте

Минусы такого подхода – чтобы найти что-то в таком списке, нужно или посмотреть его целиком, или помнить часть нужной ссылки или её описания (чтобы можно было найти поиском по странице).

Чуть более удобным вариантом является подход, когда ссылки сначала систематизируются и разбиваются по папкам (рисунок 2) [6].

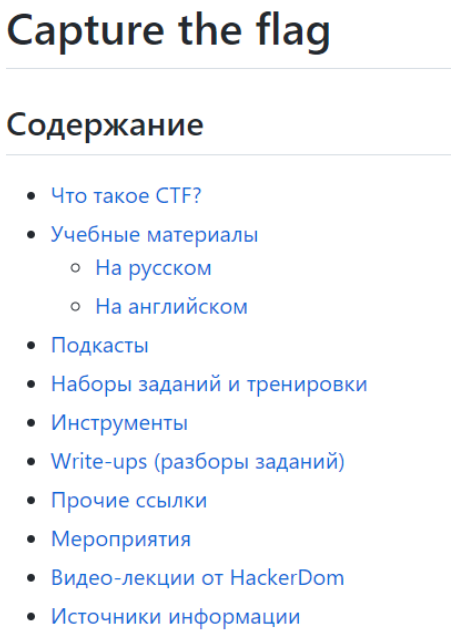


Рисунок 2 – Пример списка на GitHub, где ссылки разбиты по папкам

В таком случае пользователю сначала необходимо просмотреть список папок, а затем посмотреть список внутри, что упрощает навигацию. Стоит отметить, что в найденных в процессе обзора решениях задания являются отдельной папкой и не разбиты по категориям заданий CTF, что является недостатком.

Следующим примером системы хранения (правда, вовсе без заданий, только с теоретическими материалами) является wiki (рисунок 3) [7].

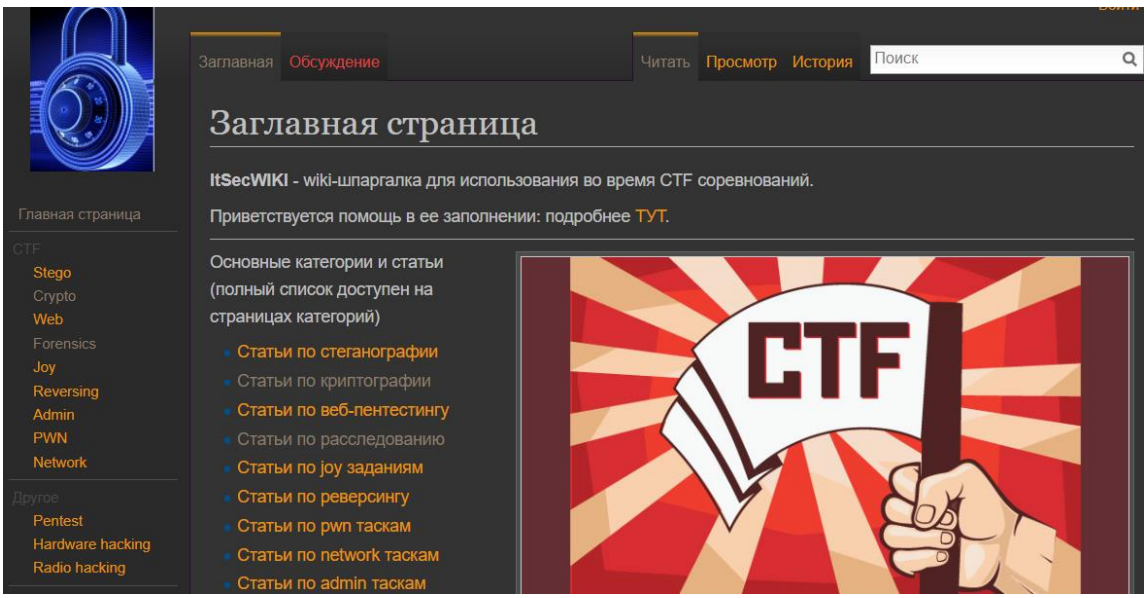


Рисунок 3 – Wiki по CTF

На сайтах, созданных с помощью wiki, как правило есть поиск по страницам, и по сути совмещает преимущества двух способов, описанных выше: можно искать самостоятельно ссылки по подразделам, можно искать глобально по во всех разделах сразу.

1.2 Решения, используемые в других предметных областях

Приложения банков постоянно совершенствуют интерфейс, но из-за обилия банковских продуктов и широких возможностей по управлению ими у пользователей приложения всё равно возникают проблемы с тем, чтобы найти нужную им функцию. Поэтому в приложения банков всё чаще добавляют бота, который осуществляет поиск по функциям приложения (рисунок 4). В данном случае уже используется интеллектуальный поиск – обученная на запросах пользователей нейросеть учитывает и возможные опечатки, и неверные термины, выдавая информацию, релевантную запросу. Проблемы возникают только в том случае, если пользователю нужно обратиться к оператору банка с проблемой, которая не входит в функционал приложения. Если использовать подобное решение для поиска по заданиям CTF, то подобный недостаток уходит, потому что пользователи не будут открывать чат с ботом для решения какой-то проблемы.

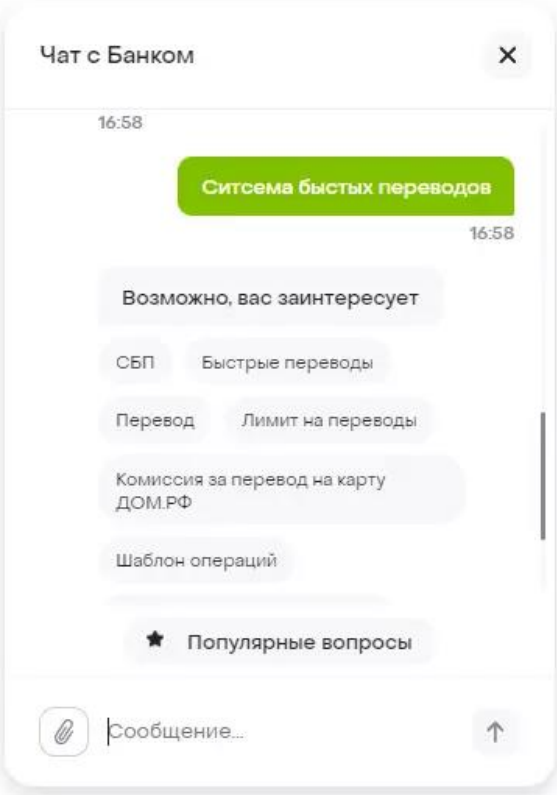


Рисунок 4 – Поисковый бот приложения банка ДОМ.РФ [8]

2 Нейронные сети

Машинное обучение (Machine Learning) — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться.
Машинное обучение находится на стыке математической статистики, методов оптимизации и классических математических дисциплин [9].

Нейронные сети — это вид моделей машинного обучения, основанные на принципе коннективизма — в них соединяется большое количество относительно простых элементов, а обучение сводится к построению оптимальной структуры связей и настройке параметров связей.

Самыми распространенными задачами нейронных сетей являются:

1. Классификация — на вход подаются данные, на выходе нужно получить категорию, к которой эти данные относятся. Например, по тексту статьи для журнала определить, к какой категории она относится - “технологии”, “обзор фильма” или “спорт”. Эту работу может сделать нейронная сеть, анализируя такую информацию, как встречаемость определённых слов или их сочетаний.
2. Регрессия — возможность предсказывать следующий шаг. Например, рост или падение акций, основываясь на ситуации на фондовом рынке.
3. Кластеризация — задача размещения входящих образов по категориям или, иначе говоря, кластерам, так чтобы близкие векторы (схожие образы) оказались в одной категории. Отличие задачи кластеризации от похожей на нее задачи классификации заключается в том, что набор категорий изначально не задан и определяется в процессе обучения нейронной сети. Примером задачи кластеризации служит задача сжатия информации путем уменьшения разнообразия данных [10]. Задача кластеризации относится к широкому классу задач обучения без учителя.

Нейрон — это вычислительный узел, который получает информацию, производит над ней простые вычисления и передает ее дальше. Они делятся на три основных типа: входной, скрытый и выходной. Каждый узел (нейрон) умножается на вес, т.е. имеет значение веса. В ходе обучения нейронная сеть регулирует эти показатели, чтобы произвести правильные выходные данные [11].

Есть три основных вида обучения нейронной сети [12]:

1. Обучение с учителем - когда есть полностью размеченный набор данных, т.е. каждому примеру в обучающем наборе соответствует ответ, который алгоритм и должен получить. Когда сеть получит новые данные, она сравнит их с примерами из обучающего набора данных, чтобы предсказать ответ.

2. Обучение без учителя - когда имеется набор данных, и нейронная сеть сама пытается найти корреляции в данных, извлекая полезные признаки и анализируя их.
3. Обучение с подкреплением - итеративный процесс, когда нейронная сеть принимает решение, получает обратную связь, положительную или отрицательную, и в зависимости от этого корректирует свою стратегию, чтобы максимизировать награду. Например, интеллектуальный робот учится передвигаться так, чтобы меньше врезаться в препятствия: за столкновение с препятствием его награда уменьшается, и он корректирует своё движение.

Обучение с учителем происходит в два этапа [13]:

1. Прямое распространение ошибки (рисунок 5);
2. Обратное распространение ошибки (рисунок 6).

Прямое распространение:

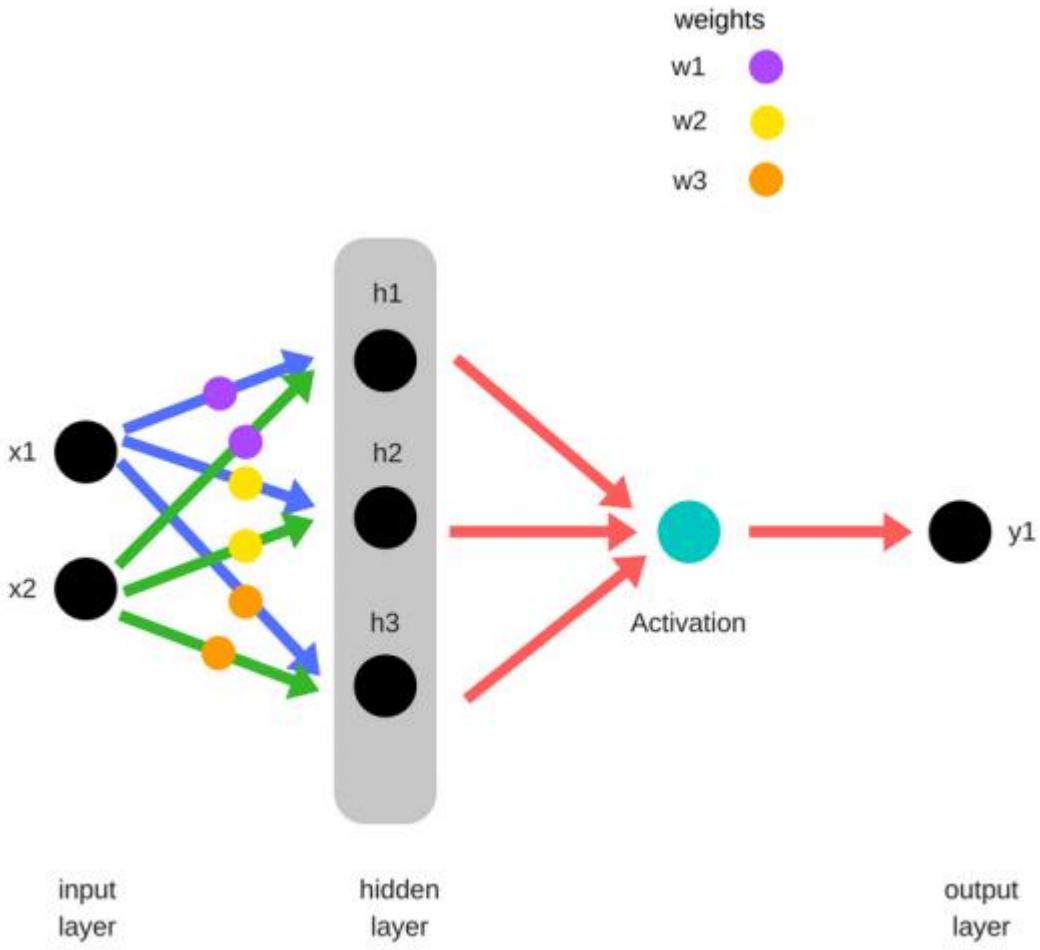


Рисунок 5 – Прямое распространение ошибки

Зададим начальные веса (w1, w2) случайным образом.

Умножим входные данные на веса для формирования скрытого слоя:

• h1 = (x1 * w1) + (x2 * w1)

• h2 = (x1 * w2) + (x2 * w2)

• h3 = (x1 * w3) + (x2 * w3)

Выходные данные из скрытого слоя передается через нелинейную функцию (функцию активации), для получения выхода сети:

• y_ = fn(h1 , h2, h3)

Обратное распространение:

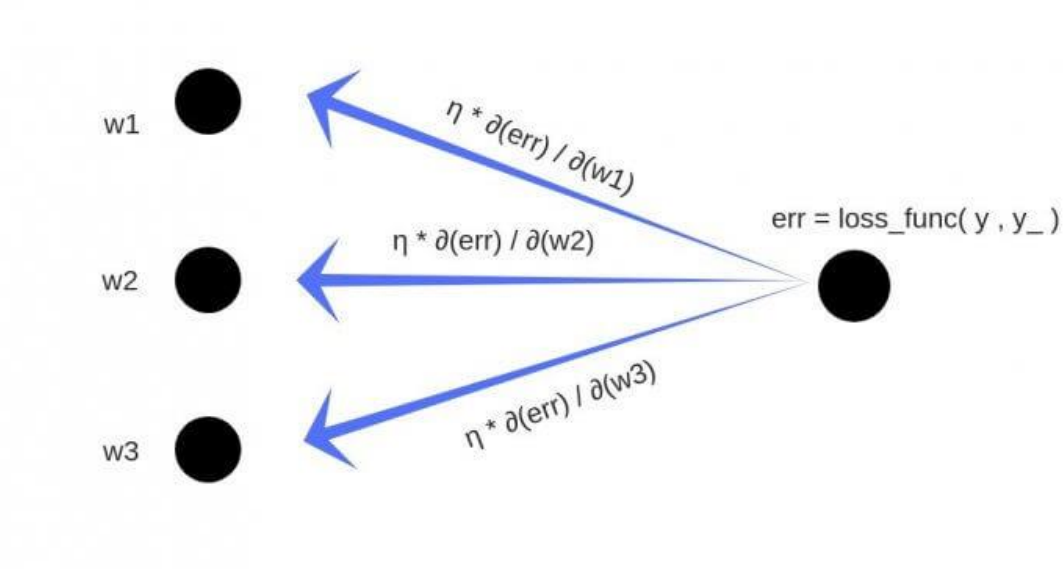


Рисунок 6 – Обратное распространение ошибки

Суммарная ошибка (total_error) вычисляется как разность между ожидаемым значением «y» (из обучающего набора) и полученным значением «y_» (посчитанное на этапе прямого распространения ошибки), проходящих через функцию потерь (cost function).

Частная производная ошибки вычисляется по каждому весу (эти частные дифференциалы отражают вклад каждого веса в общую ошибку (total_loss)).

Затем эти дифференциалы умножаются на число, называемое скорость обучения или learning rate (η).

Полученный результат затем вычитается из соответствующих весов.

В результате получатся следующие обновленные веса:

• w1 = w1 — (η * ∂(err) / ∂(w1))

• w2 = w2 — (η * ∂(err) / ∂(w2))

• w3 = w3 — (η * ∂(err) / ∂(w3))

То, что мы предполагаем и инициализируем веса случайным образом, и они будут давать точные ответы, звучит не вполне обоснованно, тем не менее, работает хорошо.

Градиентный спуск — это, по сути, и есть то, как обучаются модели.

Градиентный спуск — метод нахождения минимального значения функции потерь (существует множество видов этой функции). Минимизация любой функции означает поиск самой глубокой впадины в этой функции. Суть алгоритма – процесс получения наименьшего значения ошибки.

Кроме функции потерь градиентный спуск также требует градиент, который является dJ/dw (производная функции потерь относительно одного веса, выполненная для всех весов). dJ/dw зависит от выбора функции потерь. Наиболее распространена функция потерь среднеквадратичной ошибки (формула 1).

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 \quad (1)$$

3 Выбор инструментов реализации обучающей программы

3.1 Язык программирования Python

Python является одним из самых популярных языков программирования. Для него имеются множество библиотек под самые разные задачи, включая поставленные в данной работе. Python является самым популярным языком программирования для работы с нейронными сетями и анализа данных [14].

3.2 Telegram Bot API

Программы телеграм-ботов используют Telegram Bot API – набор методов для взаимодействия с сервером Telegram через простой HTTPS-интерфейс, который предоставляет простую версию Telegram API. Это позволяет разработчику одновременно иметь широкие возможности по предоставлению информации пользователю, но при этом для системной части программы пользоваться уже готовыми инструментами Telegram API [15]. В свою очередь пользователь видит привычный интерфейс с отправкой и получением сообщений, и ему не нужно осваивать новую программу, чтобы приступить к обучению. Данные свойства телеграм-бота позволяют сосредоточить усилия на решении такой задачи, как работа непосредственно с запросом пользователя, чтобы выдать максимально релевантную информацию. Также важно, что мессенджер Telegram является популярным среди молодёжи, что означает, что у пришедшего в CTF новичка он скорее всего уже есть.

3.3 Библиотека scikit-learn (sklearn)

Для решения задачи требуется обученная модель для классификации текста по категориям. Была выбрана sklearn – библиотека с открытым кодом для машинного обучения и анализа данных, одна из наиболее широко используемых в Python. Причиной этому в том числе является обширная документация. Она построена на базе других популярных Python-библиотек — NumPy, SciPy и matplotlib. Sklearn написана на языках Python, C, C++ и Cython. [16].

4 Создание интеллектуального бота

4.1 Схема работы интеллектуального бота

По принципу описанных выше примеров поисковых ботов, в данной работе бот будет иметь множество заготовленных вариантов ответа и использовать классифицирующую нейросеть для выбора одного из вариантов в соответствии с запросом пользователя. Общий принцип работы таков (рисунок 7):

1. Когда пользователь пишет сообщение боту, в программе бота вызывается функция, отвечающая за обработку сообщений.
2. Текст сообщения обрабатывается внутри функции: переводится в нижний регистр, разбивается на слова, переводится в начальную форму слова.
3. Подготовленный текст передаётся на вход обученной модели (подробнее в разделах 4.2 и 4.3). Модель определяет вероятность соответствия запроса каждой из имеющихся категорий. Если для какой-либо категории вероятность составляет более 75%, то возвращается её название. Если таких категорий нет, то возвращается строка “Fallback”.
4. Имеется файл или любое другое хранилище данных, где категории соответствует текстовый ответ. Также имеется ответ для строки “Fallback”. По строке, полученной на шаге 3, находится соответствующий ответ.
5. Полученный на шаге 4 ответ возвращается пользователю.

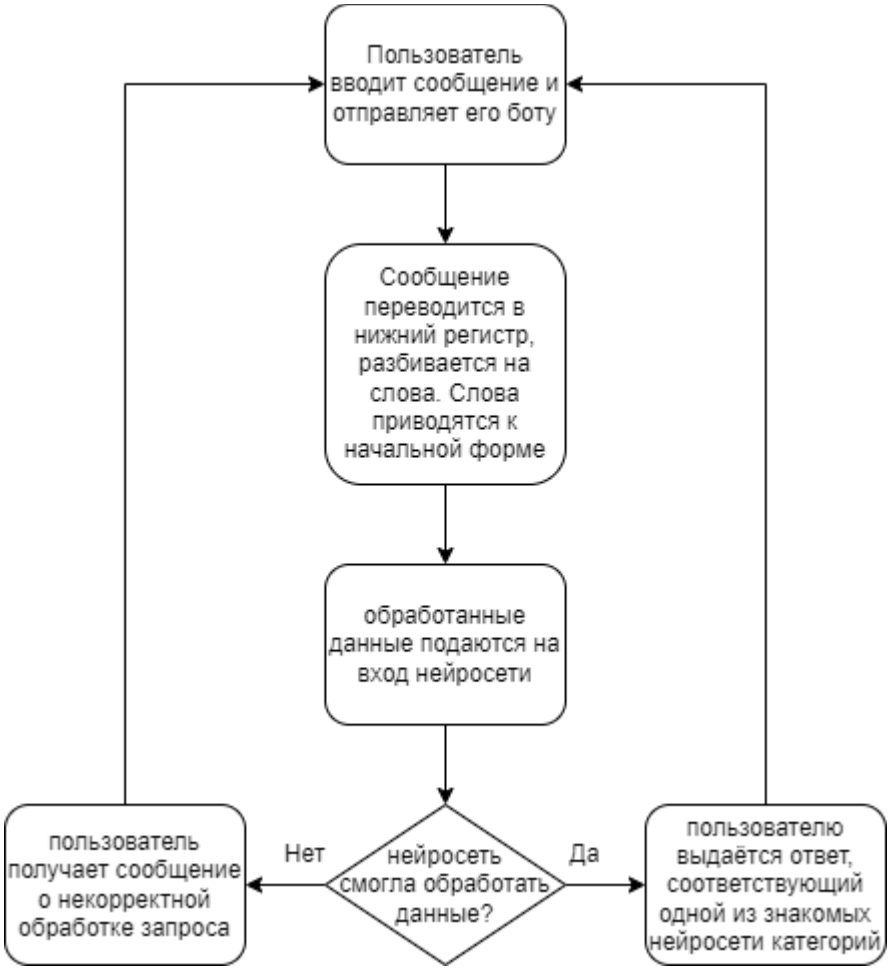


Рисунок 7 – Схема работы интеллектуального бота

Кратко функционал бота для пользователя и для администратора (человека, который осуществляет техническую поддержку) изображен на Use Case диаграммах (рисунки 8, 9)

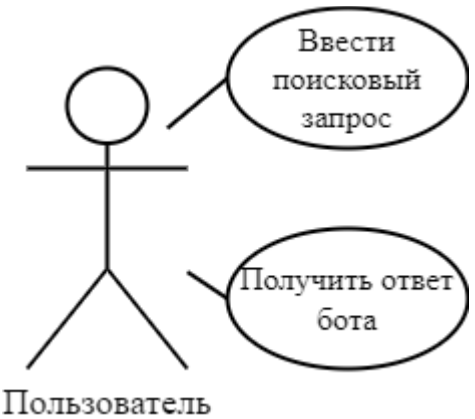


Рисунок 8 – Use Case диаграмма для роли пользователя

Рисунок 9 – Use Case диаграмма для роли администратора

Можно выделить четыре основных составляющих бота: платформа для бота (позволяющая отправлять и принимать сообщения), файл с категориями и текстовыми ответами для них, обученная классифицирующая нейросеть и сама программа, по которой работает бот. Выбор инструментов реализации описан в предыдущем разделе.

4.2 Подготовка набора данных для обучения нейронной сети

Так как на момент работы нет собранных данных о том, как пишут запросы по категориям CTF, придётся воспользоваться генерацией синтетических данных. Синтетические данные — это программно сгенерированные данные, которые по параметрам приближены к реальным.

Предположительно для категории заданий CTF должны присутствовать следующие данные:

- название категории – с разными видами ошибок (добавлена буква, пропущена буква, заменена буква), с различными синонимами и связанными по смыслу словами (подкатегориями)
- с различными синонимами слова “задание”: “упражнение”, “таск”, “тренировать”
- с разным количеством слов

Порядок слов и разные предлоги в данном случае не важны, так как во-первых при подготовке запроса короткие слова удаляются, во-вторых, используется самый простой способ преобразования текста в вектор для обучения нейросети: составляется словарь всех

возможных слов, и проверяется, есть ли то или иное слово в запросе. Это будет объяснено подробнее в следующем разделе.

4.3 Обучение нейронной сети

В качестве модели используется линейный классификатор на основе логистической регрессии для случая >2 классов (рисунок 10).

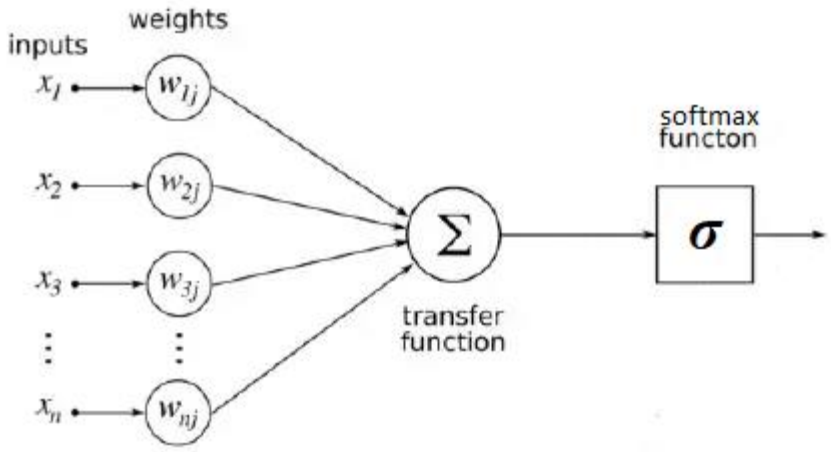


Рисунок 10 – Схема классификатора на основе логистической регрессии

Это однослойная нейросеть, где входные признаки, умноженные на матрицу весов, проходят через Softmax функцию [17]. Функция преобразует вектор **z** размерности ***K*** в вектор ***σ*** той же размерности, где каждая координата σ_i полученного вектора представлена вещественным числом в интервале ***[0,1]*** и сумма координат равна 1. Координаты σ_i вычисляются следующим образом (формула 2)

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}} \quad (2)$$

Обучение нейронной сети на классификацию текста происходит следующим образом: по обучающему набору данных строится словарь всех уникальных слов, которые присутствовали в наборе данных. Для каждого примера текста из набора вектор строится следующим образом: если слово из словаря есть в примере, то 1, если нет, то 0 [18].

Чтобы объяснить, как это работает, далее показан пример. Пусть имеется два примера текста: “Покажи задания на Forensic” и “Криптография задания”. После обработки эти тексты превратятся в списки слов: [“показать”, “задание”, “forensic”] и [“криптография”, “задание”]. Тогда словарь уникальных слов: 1-показать, 2-задание, 3-forensic, 4-

криптография. В виде вектора первый пример будет выглядеть как [1,1,1,0], второй пример – как [0,1,0,1]. Можно заметить, что оба вектора имеют 1 на слове “задание”, что показывает, что по этому слову их различить нельзя. В процессе обучения нейросеть уменьшит вес этого слова, и будет опираться на другие слова при принятии решения.

```
 тренировать с reverse@Реверс задания
 тренировать  реверс@Реверс задания
 тренировать  ревёрс@Реверс задания
 тренировать  обратный разработка@Реверс задания
 тренировать  реверс инжиниринг@Реверс задания
 тренировать  reverse@Реверс задания
показать задание по веб@Веб задания
показать задание по web@Веб задания
показать задание по http@Веб задания
показать задание по cors@Веб задания
показать задание по csrf@Веб задания
показать задание на веб@Веб задания
показать задание на web@Веб задания
показать задание на http@Веб задания
показать задание на cors@Веб задания
показать задание на csrf@Веб задания
показать задание с веб@Веб задания
показать задание с web@Веб задания
показать задание с http@Веб задания
показать задание с cors@Веб задания
```

Рисунок 11 – Часть набора данных, использованного для обучения

На рисунке 11 показана часть текстового файла. Слева от знака “@” идёт пример текста, справа – категория, которую надо присвоить этому тексту. Программа, генерирующая синтетические данные, записывает примеры в файл. Затем программа, обучающая нейросеть, читает файл, средствами библиотеки sklearn из примеров текста составляет словарь, переводит примеры в векторы и обучает нейросеть различать категории.

4.4 Примеры работы телеграм-бота

На первом примере (рисунок 12) пользователь запускает бота, затем делает два запроса. Первый запрос, “sql”, является корректным, и бот выдаёт задания на SQL-инъекции. Второй запрос написан с ошибкой (“стенография” вместо “стеганография”), но бот всё равно выдаёт задания на стеганографию.

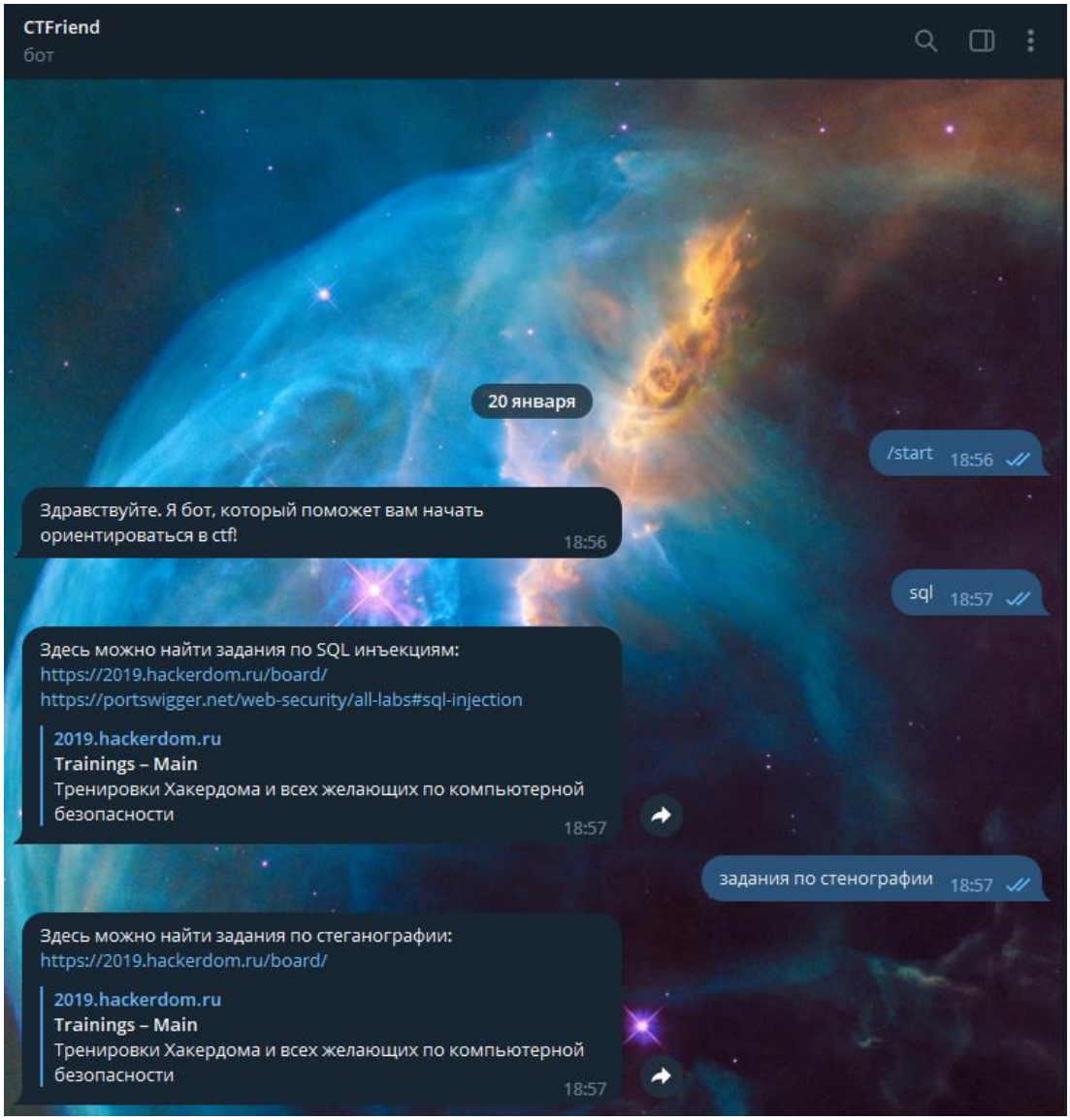


Рисунок 12 – Пример работы бота

На втором примере (рисунок 13) пользователь делает три корректных запроса, а затем пытается сделать запросы, на которые бот не может ответить, поэтому бот выдаёт ответ по умолчанию.

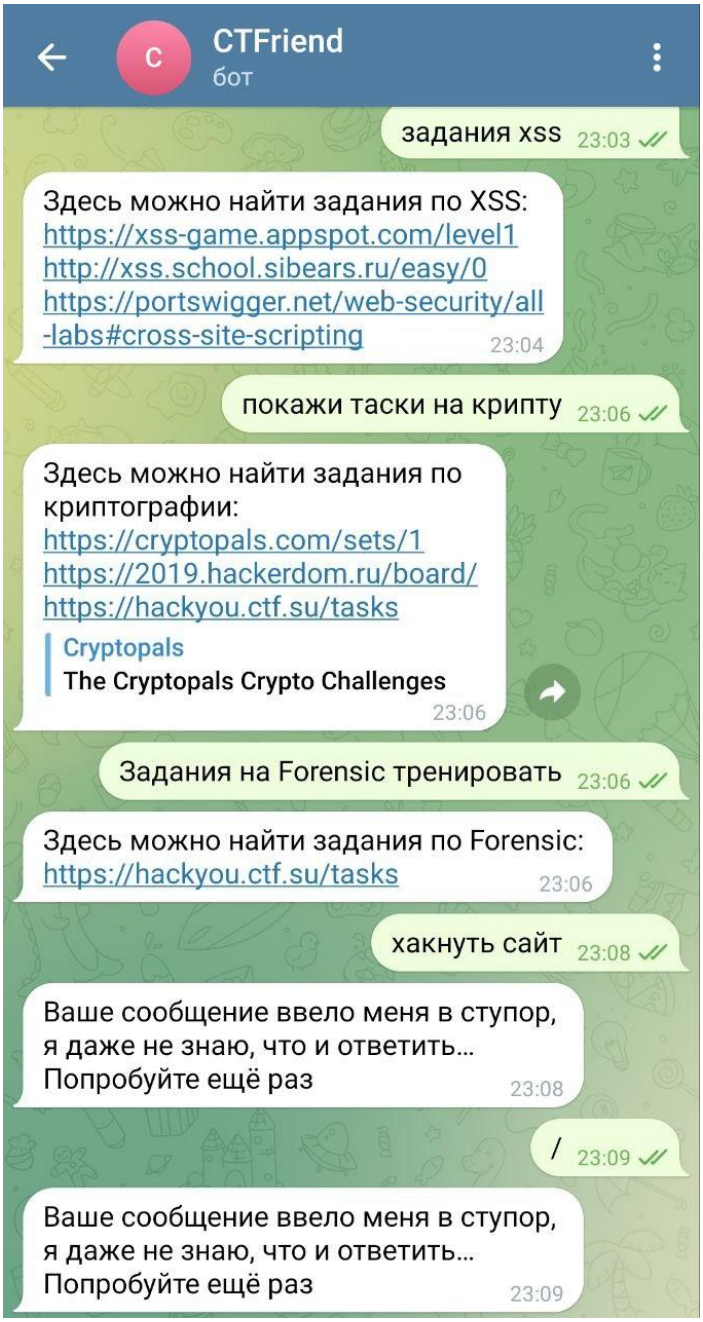


Рисунок 13 – Пример работы бота

ЗАКЛЮЧЕНИЕ

Результатом дипломной работы является поисковый телеграм-бот, предназначенного для помощи студентам, которые готовятся к CTF соревнованиям. Поисковый бот является современным решением проблемы хранения большого количества информации и эффективного поиска по ней. Принцип работы бота позволяет легко добавлять новую информацию в существующие категории. Также в бота при наличии запроса от пользователей можно будет добавить новые функции, при этом не теряя эффективности основной поисковой функции.

Ссылка на телеграм-бота: https://t.me/CTFriendBot

Исходный код в репозитории: https://github.com/GebMirFreiheit/CTFriend

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Что такое CTF [Электронный ресурс] / Всё о CTF в России. - URL: https://ctfnews.ru/what-is-ctf/ (дата обращения: 20.01.2023)
2. Что общего у CTF и тестов на проникновение [Электронный ресурс] / Омар Ганиев. - URL: https://www.slideshare.net/beched/ctf-25408688 (дата обращения: 20.01.2023)
3. Команды CTF [Электронный ресурс] / Факультет безопасности информационных технологий ИТМО. - URL: https://cit.itmo.ru/ru/page/17645/komandy-ctf.htm (дата обращения: 20.01.2023)
4. Боты в телеграм что это такое и как они работают [Электронный ресурс] / Shark Develop. - URL: https://sharkdevelop.com/boty-v-telegram/ (дата обращения: 20.01.2023)
5. Полезные материалы [Электронный ресурс] / CTF в Петербурге. – URL: https://vk.com/topic-114366489_33670323 (дата обращения: 20.01.2023)
6. Список материалов для обучения в ИТ [Электронный ресурс] / lgg. – URL: https://github.com/lgg/studying (дата обращения: 20.01.2023)
7. Заглавная страница [Электронный ресурс] / InformationSecurity WIKI. – URL: http://itsecwiki.org/index.php/Заглавная_страница (дата обращения: 20.01.2023)
8. Как воспитать виртуального ассистента: опыт ДОМ.РФ [Электронный ресурс] / Дом.рф. - URL: https://vc.ru/domrf/503064-kak-vospitat-virtualnogo-assistenta-opyt-dom-rf (дата обращения: 20.01.2023)
9. Машинное обучение [Электронный ресурс] / MachineLearning. - URL: http://www.machinelearning.ru/wiki/index.php?title=Машинное_обучение (дата обращения: 20.01.2023)
10. Юнусова Лилия Рафиковна, Магсумова Алия Рафиковна Кластеризация с помощью нейронных сетей и поиск зависимостей // Наука, образование и культура. 2019. №7 (41). URL: https://cyberleninka.ru/article/n/klasterizatsiya-s-pomoschyu-neyronnyh-setey-i-poisk-zavisimostey-1 (дата обращения: 20.01.2023)
11. Общий взгляд на машинное обучение: классификация текста с помощью нейронных сетей и TensorFlow [Электронный ресурс] / Tproger. - URL: https://tproger.ru/translations/text-classification-tensorflow-neural-networks/#1 (дата обращения: 20.01.2023)
12. Обучение нейросети с учителем, без учителя, с подкреплением — в чем отличие? Какой алгоритм лучше? [Электронный ресурс] / Neurohive. - URL:

https://neurohive.io/ru/osnovy-data-science/obuchenie-s-uchitelem-bez-uchitelja-s-podkrepleniem/ (дата обращения: 20.01.2023)

13. Как работает нейронная сеть: алгоритмы, обучение, функции активации и потери [Электронный ресурс] / Neurohive. - URL: https://neurohive.io/ru/osnovy-data-science/osnovy-nejronnyh-setej-algoritmy-obuchenie-funkcii-aktivacii-i-poteri/ (дата обращения: 20.01.2023)
14. Нейронные сети на Python: как всё устроено [Электронный ресурс] / GeekBrains. - URL: https://gb.ru/blog/nejronnye-seti-python/ (дата обращения: 20.01.2023)
15. Всё, о чём должен знать разработчик Телеграм-ботов [Электронный ресурс] / Артём Иванов. - URL: https://habr.com/ru/post/543676/ (дата обращения: 20.01.2023)
16. Scikit-learn [Электронный ресурс] / Блог SkillFactory. – URL: https://blog.skillfactory.ru/glossary/scikit-learn/ (дата обращения: 20.01.2023)
17. Softmax [Электронный ресурс] / Википедия. – URL: https://ru.wikipedia.org/wiki/Softmax (дата обращения: 20.01.2023)
18. Practical Text Classification With Python and Keras [Электронный ресурс] / Real Python. – URL: https://realpython.com/python-keras-text-classification/ (дата обращения: 20.01.2023)

АНТИПЛАГИАТ
ОБНАРУЖЕНИЕ ЗАИМСТВОВАНИЙ

Отчет о проверке на заимствования №1

Автор: Школьная Наталья
Проверяющий: Школьная Наталья

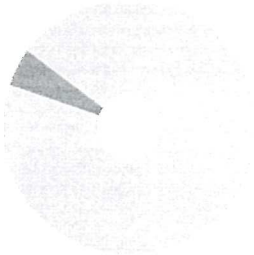
Отчет предоставлен сервисом «Антиплагиат» - http://users.antiplagiat.ru

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 4
Начало загрузки: 22.01.2023 07:58:37
Длительность загрузки: 00:00:00
Имя исходного файла: Диплом.pdf
Название документа: Диплом
Размер текста: 27 кБ
Символов в тексте: 27765
Слов в тексте: 3329
Число предложений: 186

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Начало проверки: 22.01.2023 04:58:38
Длительность проверки: 00:00:05
Комментарии: не указано
Модули поиска: Интернет Free



СОВПАДЕНИЯ	САМОЦИТИРОВАНИЯ	ЦИТИРОВАНИЯ	ОРИГИНАЛЬНОСТЬ
4,86%	0%	0%	95,14%

Совпадения — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.
Совпадения, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Актуален на	Модуль поиска
[01]	1,79%	ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБЕСПЕЧЕНИЯ АВТОНОМНОЙ НАВИГАЦИИ МОБИЛЬНОГО РОБОТА - Студенческий научный форум https://scienceforum.ru	24 Окт 2019	Интернет Free
[02]	1,71%	Разработка автоматизированного сервиса поддержки пользователей веб-сайта http://library.eltech.ru	19 Сен 2019	Интернет Free
[03]	1,36%	Кластеризация с помощью нейронных сетей http://bestreferat.su	26 Авг 2019	Интернет Free

Еще источников: 6
Еще совпадений: 0%

С результатами ознакомлена