УДК 004.852, 004.056.53

ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ОБЛАСТИ ИНЖЕНЕРНО-ТЕХНИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ

Короткова Ангелина Александровна[1], Бобылева София Вадимовна[2]

[1]Студент;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;
e-mail: koraa.19@uni-dubna.ru.

[2]Старший преподаватель;
Государственный университет «Дубна»;
Россия, 141980, Московская обл., г. Дубна, ул. Университетская, д. 19;
e-mail: sbobylova94@gmail.com.

В статье исследуется возможность применения методов машинного обучения в области инженерно-технической защиты информации. Рассматриваются основные методы машинного обучения, такие как нейронные сети, решающие деревья, метод опорных векторов и другие. Результатом работы является разработка прототипа, основанного на методах машинного обучения, который позволяет обнаруживать нарушения защиты информации, путем классификации радиочастот. Результаты работы могут быть использованы в учебном процесс и могут послужить источником для дальнейшего развития данной области.

Ключевые слова: информационная безопасность, инженерно-техническая защита информации, машинное обучение, нейронная сеть, ПАК «Кассандра К6», задача классификации.

Для цитирования:

Короткова А. А. Бобылева С. В. Применение методов машинного обучения в области инженерно-технической защиты информации // Системный анализ в науке и образовании: сетевое научное издание. 2023. № 2. С. 45-55. EDN: GYUROT. URL: https://sanse.ru/index.php/sanse/article/view/578.

APPLICATION OF MACHINE LEARNING METHODS IN THE FIELD OF ENGINEERING AND TECHNICAL INFORMATION SECURITY

Korotkova Angelina A.[1], Bobyleva Sofia V.[2]

[1]Student;
Dubna State University;
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;
e-mail: koraa.19@uni-dubna.ru.

[2]Senior teacher;
Dubna State University;
19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;
e-mail: sbobylova94@gmail.com

The article explores the possibility of using machine learning methods in the field of engineering and technical information security. The main methods of machine learning are considered, such as neural networks, decision trees, support vector machine and others. The result of the work is the development of a prototype based on machine learning methods, which allows you to detect information security violations by classifying radio frequencies. The results of the work can be used in the educational process and can serve as a source for further development of this area.

Keywords: information security, engineering and technical protection of information, machine learning, neural network, Cassandra K6, classification problem.

For citation:
Korotkova A. A., Bobyleva S. V. Application of machine learning methods in the field of engineering and technical information security. *System analysis in science and education*, 2023;(2):45-55 (in Russ). EDN: GYUROT. Available from: https://sanse.ru/index.php/sanse/article/view/578.

Введение

В настоящее время информационные технологии охватывают все сферы нашей жизни. Вместе с тем, в силу недостаточной защищенности информации и роста угроз со стороны злоумышленников, очень важно обеспечить надежную защиту информации, особенно в области инженерно-технической защиты информации.

СМИ полны сообщениями о хищении данных с использованием уязвимостей информационных систем, но технические каналы передачи побочных электромагнитных, акустических или оптических импульсов используются уже 50-60 лет и продолжают оставаться популярными, при этом технологии съема данных развиваются быстрыми темпами. Электромагнитные колебания, возникающие при вводе данных с клавиатуры или при выводе информации на монитор с легкостью, преобразуются в информацию – изображение на экране или введенный текст станут доступны злоумышленникам. Акустические и виброакустические импульсы превращаются в речь, тайна переговоров в результате считывания лазерным лучом колебаний оконного стекла станет известна конкурентам [1]. В этом и заключается актуальность данной темы.

Одним из актуальных направлений в области инженерно-технической защиты информации является применение методов машинного обучения, которые позволяют быстро и эффективно определять, и анализировать потенциально угрожающие ситуации. С помощью машинного обучения можно производить классификацию данных и выявлять аномальные ситуации.

1. Инженерно-техническая защита информации

Инженерно-техническая защита (ИТЗ) – это совокупность технических средств и мероприятий, нацеленных на предотвращение утечек, разглашения информации, и несанкционированного доступа в сетевые ресурсы организации [2].

Основными задачами ИТЗИ являются:

- обеспечение высокой степени защиты для критичной информации, такой как конфиденциальная, коммерческая и другая важная информация;
- предотвращение получения доступа к информации без разрешения, т.е. несанкционированного доступа к ней;
- обеспечение целостности и доступности информации.

Защита информации с помощью технических средств позволяет компаниям не только более глубоко и тщательно прорабатывать новые разработки и технологии, но и обезопасить их от возможной кражи интеллектуальной собственности. Использование инженерно-технических средств обеспечивает надежную защиту компьютерной информации и уменьшает риск промышленного шпионажа, а также нелегального использования новых продуктов, поэтому это является важнейшей задачей для любой компании.

2. Обнаружение технического канала утечки информации

Технические каналы утечки информации – это способы раскрытия информации, которые связаны с контролируемыми техническими характеристиками информационных систем и средств обработки данных. Некоторые из таких каналов могут быть созданы специально, другие же могут возникать случайно.

Технические средства для разведки и обезвреживания каналов утечки информации можно определить следующим образом:

1. Активный тип поисковых работ;
2. Пассивный способ обнаружения утечки.

Обнаружить технический канал утечки информации можно с помощью программно-аппаратного-комплекса «Кассандра К6».

Программно-аппаратный комплекс радиомониторинга «Кассандра К6» – это программно-аппаратный комплекс радиомониторинга, который включает в себя аппаратные средства и специализированное программное обеспечение. Он используется для проведения мониторинга радиочастотного диапазона и анализа работоспособности радиоэлектронных средств (РЭС), таких как радиостанции, радиорелейные линии, сети передачи данных, спутниковые системы связи и др.

Основные возможности системы «Кассандра К6» включают:

1. Мониторинг радиоэлектронных средств – система позволяет получать информацию о работе РЭС и выявлять неисправности или нарушения в их работе.
2. Анализ качества связи – система проводит анализ качества связи в радиочастотном диапазоне, что позволяет выявлять проблемы с сигналом, перегрузки и препятствия на пути сигнала.
3. Анализ и прогнозирование надежности системы связи – система использует методы машинного обучения для анализа данных о работе РЭС и построения прогнозов отказов.
4. Информационная поддержка принятия решений – система предоставляет точную и своевременную информацию о работе РЭС, что помогает операторам мониторинга принимать взвешенные решения при управлении системами связи.

3. Машинное обучение в защите информации

Машинное обучение (МО) – это область искусственного интеллекта, которая позволяет компьютерам определять закономерности в данных, на основе которых они могут создавать модели и делать предсказания.

1. Применение машинного обучения в защите информации может быть использовано для следующих задач:
2. Анализ логов: МО может быть использовано для анализа больших объемов системных журналов и логов, чтобы выявить необычное поведение и предотвратить возможные атаки на систему.
3. Обнаружение вторжений: машинное обучение может помочь прогнозировать и обнаруживать вторжение в систему путем анализа подозрительной активности и поведения в режиме реального времени.
4. Определение аномалий: МО может использоваться для анализа поведения пользователей в системе и определения аномалий, таких как необычные входы в систему, необычные запросы и т.д.
5. Классификация угроз: МО может использоваться для классификации типов угроз на основе анализа данных, связанных с ранее возникшими событиями на защищаемой системе.
6. Улучшение процесса управления правами доступа: МО можно использовать для анализа поведения пользователей в системе, что поможет повысить эффективность управления правами доступа и сократить риски утечки конфиденциальной информации [3].

Основные классы задач, решаемых с помощью машинного обучения, включают в себя:

1. Классификация: задача отнесения объекта к определенному классу на основе представленных данных [4].
2. Регрессия: задача предсказания численного значения на основе представленных данных.
3. Кластеризация: задача группировки объектов на основе их характеристик без заранее известных метки классов.

4. Снижение размерности: задача уменьшения количества признаков в представленных данных без потери информации.

Для решения задачи, поставленной в данной работе, был использован метод решения задачи классификации, такой как нейронные сети.

4. Нейронные сети

Нейронная сеть – это информационная модель, построенная по принципу функционирования биологических нейронных систем, которая позволяет решать различные задачи, такие как распознавание образов, классификация, прогнозирование и т.д.

- Существует несколько типов нейронных сетей, которые широко применяются для задач классификации:

- Многослойный персептрон (*MLP*) – это наиболее распространенный тип нейронной сети для классификации. Он состоит из нескольких слоев, каждый из которых содержит нейроны, связанные с предыдущим и следующим слоями. Входной слой принимает данные, а выходной слой выполняет классификацию.

- Сверточная нейронная сеть (*CNN*) – это тип нейронной сети, который используется для классификации изображений. Она работает на основе метода свертки, который позволяет выделять признаки на изображении.

- Рекуррентная нейронная сеть (*RNN*) – это тип нейронной сети, который используется для обработки последовательных данных, таких как текст или звук. Она направлена на сохранение состояния сети между шагами времени, что позволяет ей запоминать предыдущие значения и использовать их для классификации.

- Нейронная сеть преобразований (*NTN*) – это тип нейронной сети, который используется для классификации, когда данные представлены в виде графов и объектов, а не в виде обычных таблиц.

В работе был использован такой тип нейронной сети, как многослойный персептрон.

5. Проектирование нейронной сети

Для разработки нейронной сети были определены следующие функциональные требования:

1. разработанная нейронная сеть должна классифицировать частоты, которые уловила «Кассандра К6» с приемлемой точностью;
2. разработанная система должна сохранять обученную модель для её дальнейшего использования.

Для реализации модели нейронной сети был выбран язык *Python*, а для написания кода был выбран веб-интерфейс *Google Colab*.

Для прототипа классификатора радиочастот была сформирована обучающая выборка, в которой будут находиться следующие данные: частота в МГц и название предполагаемого устройства. Фрагмент обучающей базы знаний представлен на рисунке ниже (см. рис. 1). Тестовая выборка была сформирована автоматически, путем выгрузки базы данных частот из ПАК «Кассандра К6», используя ПО «*RadioInspector*». Фрагмент обучающей базы знаний представлен на рисунке ниже (см. рис. 2).

Частота, МГц	Название
0	связь с подводными лодками
0,01	связь с подводными лодками
0,02	связь с подводными лодками
0,03	связь с подводными лодками
0,03	компьютер
0,04	компьютер
0,05	компьютер
925	сотовые телефоны
926	сотовые телефоны
927	сотовые телефоны
928	сотовые телефоны
929	сотовые телефоны
930	сотовые телефоны
931	сотовые телефоны
932	сотовые телефоны
933	сотовые телефоны
934	сотовые телефоны
935	сотовые телефоны
936	сотовые телефоны
937	сотовые телефоны
938	сотовые телефоны
939	сотовые телефоны

Рис. 1. Фрагмент обучающей выборки

№		Частота, МГц	Полоса частот кГц	Время обнаружения	Период обнаружения	Мощность над порогом	Уровень сигнала
1	0	955,85165	192,868	2021.03.04 13:16:19	12679/14770	0,41	54,2
2	0	943,8102	197,701	2018.10.13 15:39:46	8457/14254	0,4	54,6
3	0	956,38907	221,248	2021.03.04 13:16:21	7659/14765	0,78	61
4	0	125,03213	117,252	2018.10.13 15:39:39	6388/6386	0,07	49,9
5	0	927,05791	144,504	2018.10.13 15:09:00	2506/14934	0,1	50,4
6	0	880,75155	1782,956	2021.03.04 13:16:21	2421/3733	2,94	52,9
7	0	926,69175	142,583	2018.10.13 15:09:06	2372/15667	0,1	47
8	0	927,65957	149,549	2018.10.13 15:08:13	2354/15588	0,12	47,9
9	0	929,14054	144,086	2018.10.13 15:08:11	2292/16175	0,1	46,9
10	0	928,29105	147,819	2018.10.13 15:08:56	2188/15679	0,11	50,6
11	0	927,17594	140,481	2018.10.13 15:39:40	2132/14877	0,1	46,9
12	0	929,25936	142,969	2018.10.13 15:08:08	2104/16119	0,1	49,4
13	0	929,02728	139,908	2018.10.13 15:08:12	2092/16153	0,1	46,7
14	0	1827,2418	116,292	2018.10.13 15:40:00	2060/12649	0,07	47
15	0	926,93993	141,693	2018.10.13 15:09:05	2042/15491	0,1	50
16	0	941,21367	200,122	2021.03.04 13:16:32	1985/4225	0,37	57,9
17	0	928,40788	145,475	2018.10.13 15:10:04	1977/15456	0,11	47
18	0	927,45733	151,156	2018.10.13 15:08:11	1965/15910	0,12	48,4
19	0	927,86691	148,434	2018.10.13 15:08:25	1949/15940	0,11	47,2
20	0	869,89912	812,376	2018.10.13 15:40:11	1948/3749	0,88	59,5

Рис. 2. Фрагмент тренировочной выборки

6. Реализация нейронной сети

На рис. 3 представлены все необходимые для реализации алгоритма библиотеки Python.

```
from keras.utils import np_utils
import cv2
import numpy as np
import tensorflow as tf
import pandas as pd
from sklearn import preprocessing
from keras.utils import to_categorical
from keras.utils import plot_model
from IPython.display import SVG
from keras.utils.vis_utils import model_to_dot
from keras.optimizers import SGD
from tensorflow import keras
from keras import layers,models,utils
from IPython.display import SVG
#from tensorflow.python.keras.utils.vis_utils import model_to_dot
from keras.utils.vis_utils import model_to_dot
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split
from keras.layers import Dropout, Dense , Flatten
import matplotlib.pyplot as plt
```

Рис. 3. Импорт библиотек

Загружаем, читаем и преобразуем в нужный вид базу знаний и базу частот.

```
df = pd.read_csv('База-знаний-для-диплома (2).csv')
df=df.drop(columns = ['Unnamed: 3','Unnamed: 4','Unnamed: 5','Unnamed: 6','Unnamed: 7','Unnamed: 8','Unnamed: 9'],axis = 1)
df
```

	Частота, МГц	Название	Unnamed: 2
0	0.01	связь с подводными лодками	0
1	0.02	связь с подводными лодками	0
2	0.03	связь с подводными лодками	0
3	0.03	компьютер	1
4	0.04	компьютер	1
...	...	...	...
1860	468.00	Прослушивающие радиоустройства	16
1861	468.00	Прослушивающие радиоустройства	16
1862	469.00	Прослушивающие радиоустройства	16
1863	469.00	Прослушивающие радиоустройства	16
1864	470.00	Прослушивающие радиоустройства	16

1865 rows × 3 columns

Рис. 4. Загрузка базы знаний

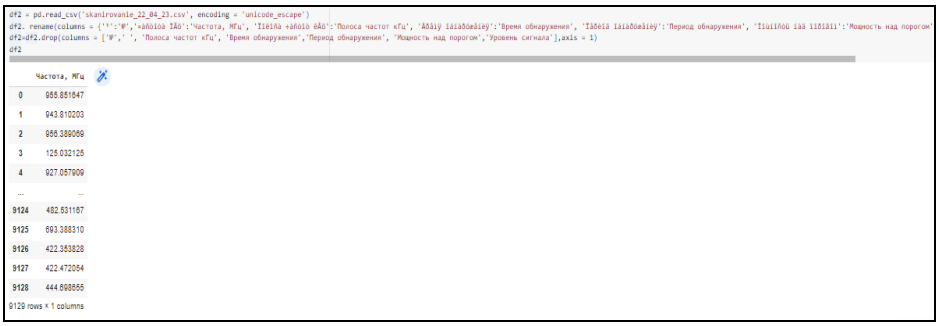

Рис. 5. Загрузка баз данных для тестирования

Затем необходимо было создать массив с обозначениями классификаций.

```
t={0:'связь с подводными лодками', 1:'компьютер', 2:'Сотовые телефоны',
      3:'GSM-жучки', 4:'Мобильная связь',
      5:'Мобильный интернет', 6:'Прослушивающие устройства',
      7:'Мобильные сети 4G  (4G mobile|CDMA «МТС» и «ТЕЛЕ2»)\xa0.',
      8:'4G mobile «Билайн» и «Мегафон»',
      9:'Беспроводная передача данных, передача аудио и видеоданных и управления квадрокоптерами. ',
      10:'Канал передачи информации во многих устройствах.  (WI-FI 802.11 a/ac)',
      11:'Спутниковая навигация.', 12:'Скрытые микрофоны',
      13:'Беспроводные камеры', 14:'Автомобильные радиостанции',
      15:'Портативные\xa0радиостанции (рации)',
      16:'Прослушивающие радиоустройства'}
```

Рис. 6. Массив с классификациями

Задаем значения обучающей и тестовой выборок. В переменную *X* записываем значения столбца «Частота, МГц» из обучающей базы знаний, а в *Y* – значения столбца «Название» из той же базы знаний.

```
X=df['Частота, МГц']
Y=df.select_dtypes(include=[object])
Y.Название.unique()
```

```
array(['связь с подводными лодками', 'компьютер', 'Сотовые телефоны',
       'GSM-жучки', 'Мобильная связь', 'Мобильный интернет',
       'Прослушивающие устройства',
       'Мобильные сети 4G  (4G mobile|CDMA «МТС» и «ТЕЛЕ2»)\xa0.',
       '4G mobile «Билайн» и «Мегафон»',
       'Беспроводная передача данных, передача аудио и видеоданных и управления квадрокоптерами. ',
       'Канал передачи информации во многих устройствах.  (WI-FI 802.11 a/ac)',
       'Спутниковая навигация.', 'Скрытые микрофоны',
       'Беспроводные камеры', 'Автомобильные радиостанции',
       'Портативные\xa0радиостанции (рации)',
       'Прослушивающие радиоустройства'], dtype=object)
```

Рис. 7. Присвоение значений переменным X и Y

Затем используя функцию «*train_test_split*», которая позволяет разделить датасет на обучающую и тестовую выборку, делим базу знаний на две части.

```
X_train , x_test , Y_train , y_test = train_test_split(X , Y ,test_size = 0.3)
```

Рис. 8. Разделение обучающего датасета на две выборки

Преобразовываем данные для *Keras*. Преобразуем метки классов (названия устройств), представленные в виде строк, в набор чисел.

```
le = preprocessing.LabelEncoder()
train_lables2=Y_train.apply(le.fit_transform)
test_lables2=y_test.apply(le.fit_transform)
```

Рис. 9. Преобразование меток классов в набор чисел

Преобразуем метки в тензорный объект.

```
train_lables=to_categorical (train_lables2)
train_lables
test_lables=to_categorical (test_lables2)
test_lables
```

Рис. 10. Преобразование меток классов в тензорный объект

Выводим размерности обучающих и тестовых переменных, чтобы проверить что все будет работать, так как есть размерности разные, например, значений *x_train* больше или меньше, чем *train_lables*, то программа выдаст ошибку и обучить нейронную сеть не получится.

Следующим этапом работы было создание самой модели. На рис. 11 представлена модель.

```
mlp1 = Sequential()
#добавляем уровни сети:
mlp1.add(Dense(800, input_dim=1, kernel_initializer='normal', activation='relu'))
mlp1.add(Dense(17, kernel_initializer="normal", activation="softmax"))

opt = SGD(learning_rate=0.01, momentum=0.9)
mlp1.compile(optimizer=opt, loss='categorical_crossentropy', metrics=['accuracy'])
mlp1.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 800)               1600

 dense_1 (Dense)             (None, 17)                13617

=================================================================
Total params: 15,217
Trainable params: 15,217
Non-trainable params: 0
_________________________________________________________________
```

Рис. 11. Создание модели

Sequential позволяет создавать последовательную группировку линейного стека слоев. Следующими действиями в созданную модель добавляются следующие параметры:

Dense – добавляет в нейронную сеть еще один «плотный» слой. Плотным слоем называют слой начального уровня, предоставляемый надстройкой *Keras*, который принимает в число нейроном или единиц в качестве своего требуемого параметра. *Input_dim* = 1 – количество признаков, у нас это один признак. *kernel_initializer='normal'* – это опция, используемая в библиотеке глубинного обучения Keras в Python для инициализации начальных весов ядер (значений внутри нейронов) в нейронных сетях. В этом случае normal означает, что начальные веса ядер будут выбраны из нормального распределения, со средним значением равным 0 и стандартным отклонением 0.05. Использование *kernel_initializer='normal'* может помочь улучшить производительность нейронной сети, ускорить

сходимость обучения и повысить качество результатов. *Activation = 'relu' и Activation = 'softmax'* – функция активации. Функция активации необходима для вычисления выходного сигнала нейрона. *Relu* – является выпрямленной линейной функцией активации. Ее формула выглядит так: F(s) = max(0,s). Функция активации *softmax* применяется на выходе и представляет собой обобщение сигмоидной функции для случая, когда выходы являются множеством классов (многоклассовая классификация). Она принимает на вход вектор значений и вычисляет вероятность принадлежности каждого элемента к определенному классу. Таким образом, значения выходного слоя будут в интервале [0,1] и сумма всех выходов будет равна 1. Применение активации *softmax* имеет преимущества в том, что она позволяет получить вероятностную оценку для каждого класса, что удобно при классификации.

Compile – данная функция позволяет настроить функции потерь, оптимизации и метрики нейронной сети. *Optimizer="SGD"* – это опция, используемая для задания метода оптимизации градиентного спуска во время обучения нейронных сетей. *SGD (Stochastic Gradient Descent)* – это простой и наиболее распространенный метод оптимизации градиентного спуска. Метод *SGD* выполняет обновление весов модели на основе градиента функции потерь, вычисленного на каждом примере входных данных. Во время обучения SGD обрабатывает обучающие данные, один за другим, каждый раз записывая градиент функции потерь. После прохода по всему обучающему набору, веса обновляются на основе среднего значения градиентов во всем наборе данных. *loss='categorical_crossentropy'* – это опция, используемая для задания функции потерь во время обучения нейронной сети. *Categorical cross-entropy* (категориальная перекрестная энтропия) – это функция потерь, которая вычисляет ошибку между настоящими и предсказанными значениями и показывает насколько точно модель предсказывает правильный класс для каждого примера. *metrics=['accuracy']* - это опция, используемая для задания метрики оценки качества работы модели во время обучения.'*accuracy*' – вычисляет долю правильных ответов, которые модель предсказала из общего количества примеров.

Просматриваем общую информацию о созданной модели.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 800)               1600

 dense_1 (Dense)             (None, 17)                13617

=================================================================
Total params: 15,217
Trainable params: 15,217
Non-trainable params: 0
_________________________________________________________________
```

Рис. 16. Информация о модели

Приступаем к обучению созданной модели на предоставленных данных *X_train, x_test* и метках *train_labels, test_lables* за 400 эпох с размером пакета (*batch size*) равным 1. Результат обучения сохраняется в переменной a, которая содержит историю потерь и точности на каждой эпохе обучения (рис.12, рис.13).

```
a = mlp1.fit(X_train, train_lables, validation_data=(x_test,test_lables), epochs = 400, batch_size = 1)
```

Рис. 12. Запуск обучения

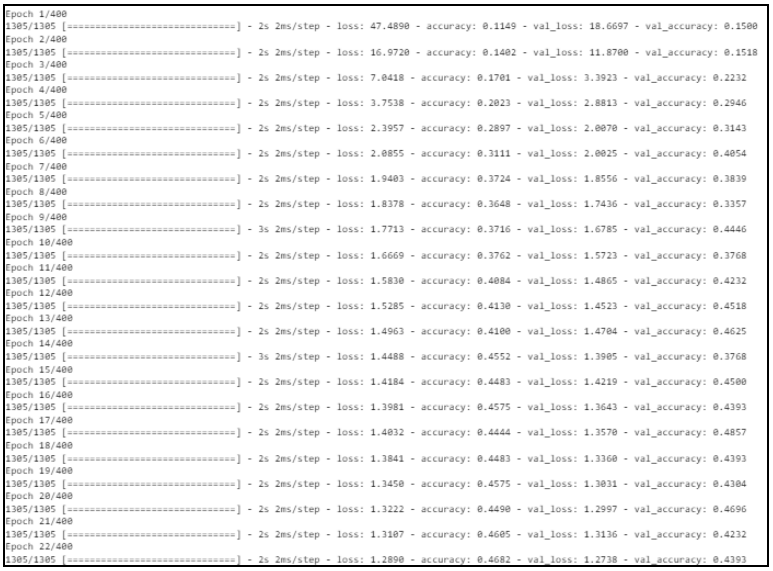

Рис. 13. Обучение модели

Модель обучилась и точность того, что она классифицирует частоту правильно равна 56%.

Рис. 14. Точность модели

Для проверки работоспособности программы, ей на вход было отправлено сначала число из тренировочной выборки.

Запуск распознавания набора данных, на котором обучалась сеть.

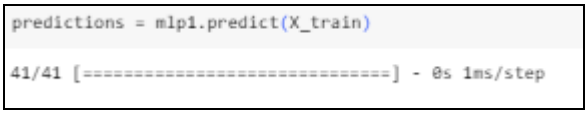

Рис. 21. Запуск распознавания

Проверка одной любой частоты из тренировочной выборки.

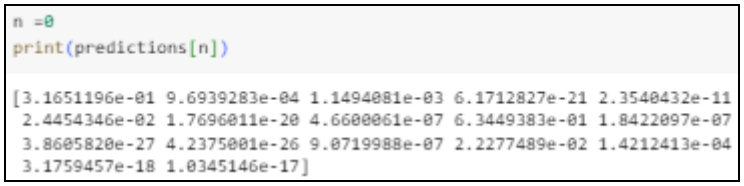

Рис. 22. Проверка элемента из тренировочной выборки

Определение номера класса частоты, который предлагает сеть.

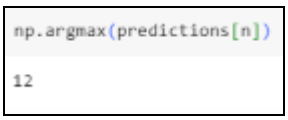

Рисунок 23. Определение номера класса

Печать названия класса.

```
t[np.argmax(predictions[n])]

'Скрытые микрофоны'
```

Рис. 24. Название класса

Печать номера класса правильного ответа.

```
np.argmax(train_lables[n])

12
```

Рис. 25. Номер класса правильного ответа

Печать названия класса правильного ответа.

```
t[np.argmax(train_lables[n])]

'Скрытые микрофоны'
```

Рис. 26. Название класса правильно ответа

Распознавание набора данных «Кассандры К6».

```
x_test2=df2['Частота, МГц']

"""делаем предсказания по всем тестовым данным"""
predictions = mlp1.predict(x_test2)
"""извлекаем номера предсказаний с максимальными вероятностями по всем объектам тестового набора"""
predictions = np.argmax(predictions, axis=1)
predictions
```

Рис. 27. Распознавание тестового набора

Запись результатов предсказаний в базу данных с частотами.

```
df2['Предсказание'] = predictions
df2
```

Рис. 28. Запись результатов классификации

Проверка того, что получилось.

	Частота, МГц	Предсказание
0	955	5
1	943	0
2	956	5
3	125	12
4	927	0
...	...	...
9124	482	10
9125	693	10
9126	422	10
9127	422	10
9128	444	10

9129 rows x 2 columns

Рис. 29. Проверка

Таким образом, в данном разделе было проведено тестирование алгоритма классификации радиочастот из базы данных, которые обнаруживает программно-аппаратный комплекс радиомонито-

ринга «Кассандра К6». В процессе тестирования программа использовала входные данные, представленные базой данных с различными частотами. Один из основных параметров обучения и тестирования был связан с единственным признаком – частотой в МГц. Точность обученной модели составляет 56%.

Тестирование разработанного алгоритма дало возможность получить данные о его работе и выявить недостатки. Однако, на основе этих данных были предложены варианты улучшения алгоритма.

Заключение

Анализ результатов работы показал, что реализованная нейронная сеть допускает ошибки, что может объясняться недостаточным количеством признаков. Чтобы улучшить качество распознавания, необходимо:

- переобучить модель с использованием большего количества признаков, например, полосы частот, которую также распознает ПАК «Кассандра К6»;
- добавить новые диапазоны частот;
- доработать нормализацию данных;
- применить другие функции потерь;
- провести более детальный анализ данных.

Данная работа показала, что методы машинного обучения могут быть эффективно применены в области инженерно-технической защиты информации. Была выявлена возможность использования нейронных сетей для классификации частот, которые могли быть угрозой утечки информации. Результаты данной работы могут быть использованы для разработки систем защиты информации на основе методов машинного обучения.

Список источников

1. Угрозы реализации технических каналов утечки информации. – ООО «СёрчИнформ», 2022. — URL: https://searchinform.ru/analitika-v-oblasti-ib/utechki-informatsii/sluchai-utechki-informatsii/tekhnicheskie-kanaly-utechki-informatsii/ugrozy-realizatsii-tekhnicheskikh-kanalov-utechki-informatsii/ (дата обращения: 13.12.2022).
2. Инженерно-техническая защита информации. – ООО «СёрчИнформ», 2022. — URL: https://searchinform.ru/services/outsource-ib/zaschita-informatsii/tekhnicheskaya/inzhenerno-tekhnicheskaya/ (дата обращения: 21.05.2023).
3. Машинное обучение и ИИ в области информационной безопасности: возможности,ограничения и риски / AB Софт // Кибербезопасность, информационная безопасность: cisoclub.ru. — CISOCLUB, 2020-2023. — Дата публикации: 25.05.2023. — URL: https://cisoclub.ru/mashinnoe-obuchenie-i-ii-v-oblasti-informacionnoj-bezopasnosti-vozmozhnosti-ogranichenija-i-riski/.
4. Задача классификации (Classification problem) // A Loginom Wiki. — Loginom, 2023. — URL: https://wiki.loginom.ru/articles/classification-problem.html (дата обращения: 13.04.2023).