National College of Ireland

Security Vulnerability Detection with Enhanced Privacy Preservation for Edge Computing Using Hybrid Machine Learning Approach - Configuration Manual

MSc Research Project
Cloud Computing

Shubham Prashant Patil
Student ID: 21139261

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shubham Prashant Patil
Student ID:	21139261
Programme:	Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	15/12/2022
Project Title:	Security Vulnerability Detection with Enhanced Privacy Preservation for Edge Computing Using Hybrid Machine Learning Approach - Configuration Manual
Word Count:	573
Page Count:	8

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	14th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Security Vulnerability Detection with Enhanced Privacy Preservation for Edge Computing Using Hybrid Machine Learning Approach - Configuration Manual

Shubham Prashant Patil
21139261

1 Introduction

This configuration manual/handbook outlines the process that can be used to recreate the development of a hybrid machine-learning algorithm with infinite feature selection and PCA. In addition to this, it includes the basic configuration that is required to set up the tools that are essential for the investigation. The overarching goal of this paper is to provide the instructions and background information necessary to successfully run the code that was included as part of the research project submission. Jupyter notebook is used throughout the whole of the project in order to carry out the coding.

2 System configuration requirements

This section refers to a technical document that serves as a guideline for creating software. It describes the program or application in general, such as its purpose, its parameters, how it will communicate with its environment as well as its users, and its necessary hardware and software.

2.1 Hardware Requirements

2.1.1 The experiment was carried out on the following hardware:

- Processor: AMD Ryzen 5 5600U with Radeon Graphics 2.30 GHz
- RAM: 16.0 GB
- System type: 64-bit operating system, x64-based processor

2.1.2 Minimum hardware requirements are:

- Modern Operating System:
    1. Windows 7 or 10
    2. Mac OS X 10.11 or higher 64-bit

3. Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu)

- x86 64-bit CPU (Intel / AMD architecture)
- 4 GB RAM
- 5 GB free disk space

2.2 Software Requirements

- Google Colaboratory: cloud-based jupyter notebook, python version 3.8.
- Email: Gmail account needed for accessing the drive.
- Browser: Any web browser.
- Other Software: Word.

3 Project Recreation

Within this area, instructions are given on how to construct the infrastructure and where to get information.

3.1 Environment Setup

Google Collaboratory is used for the process of developing models. It is necessary to have a working Gmail account in order to access the Google Collaboratory. Python version 3.10.5 was used during the whole Model creation process.

- Step 1: Launch the Google Collaboratory by clicking on the link. Link: `https://research.google.com/colaboratory/`
- Step 2: Navigate to the file area, and then launch a new notebook.

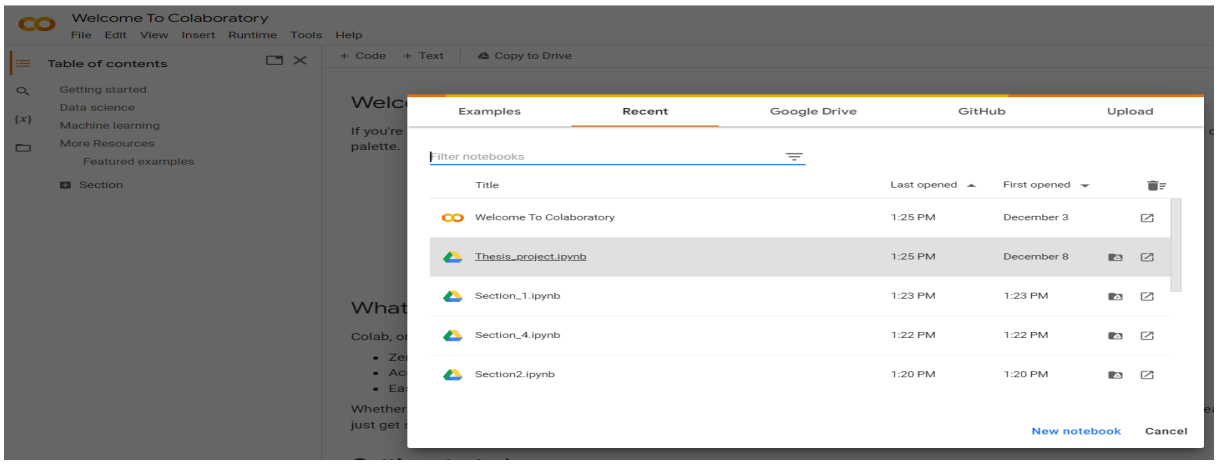

Figure 1: Creation of new notebook

- Step 3: Navigate to the area labeled "upload," then choose the file to upload.

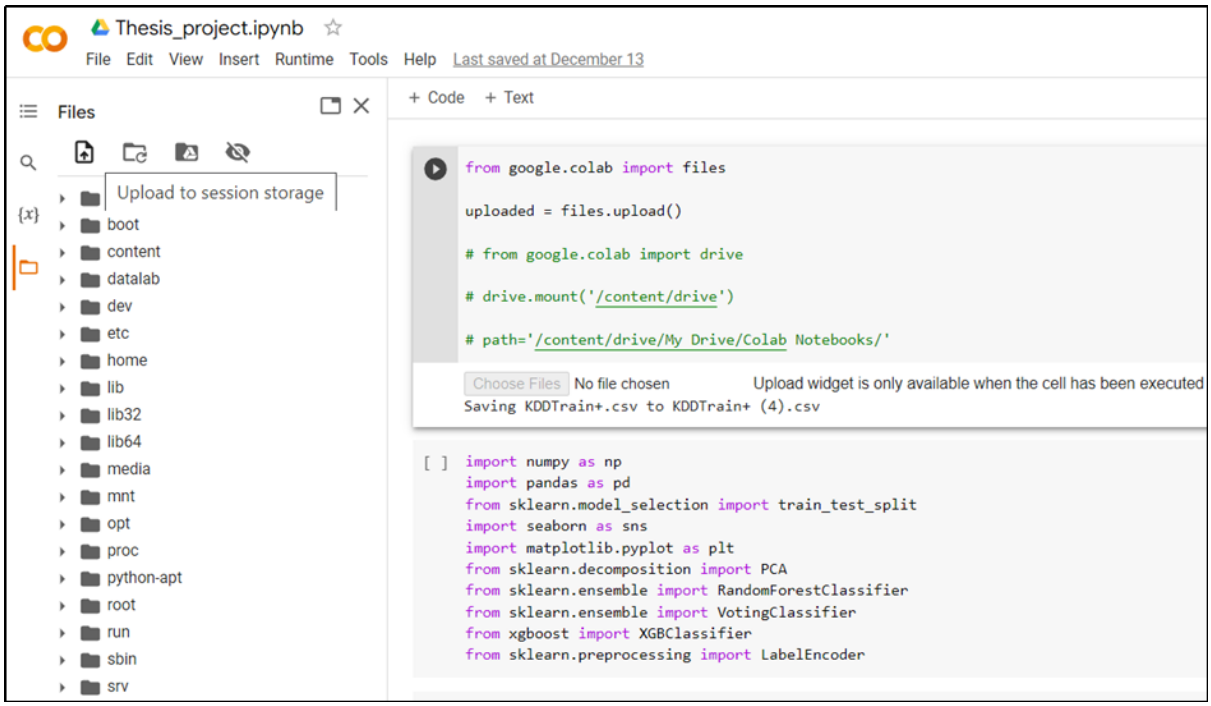

Figure 2: Uploading the files

- Step 4: There are two options to import the dataset.

  1. Importing dataset file via Drive.

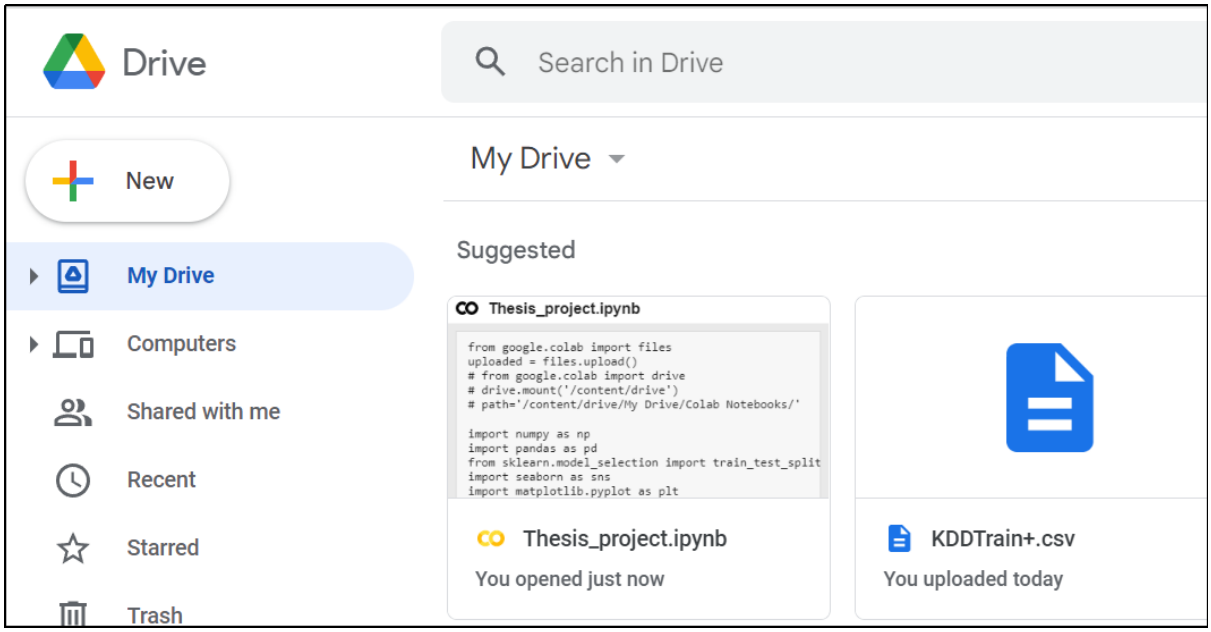

Figure 3: Importing dataset

  2. Importing dataset file via Code.

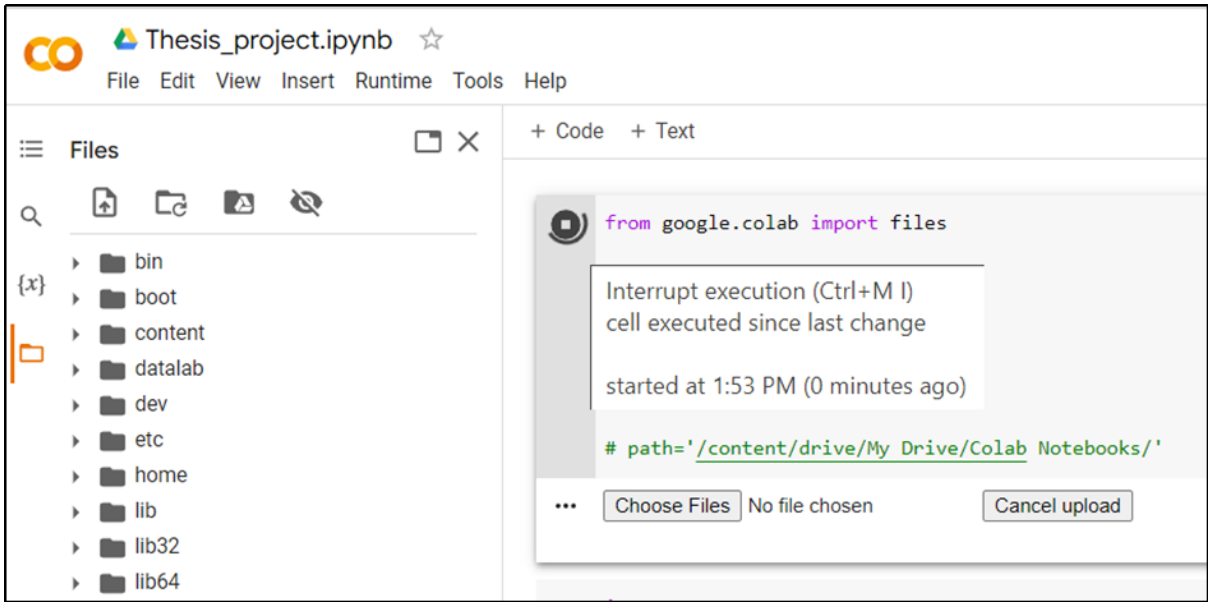

Figure 4: Imported via browse option

3.2 Packages and libraries

After the dataset has been successfully uploaded, check that the following libraries have been imported before continuing with the code implementation.

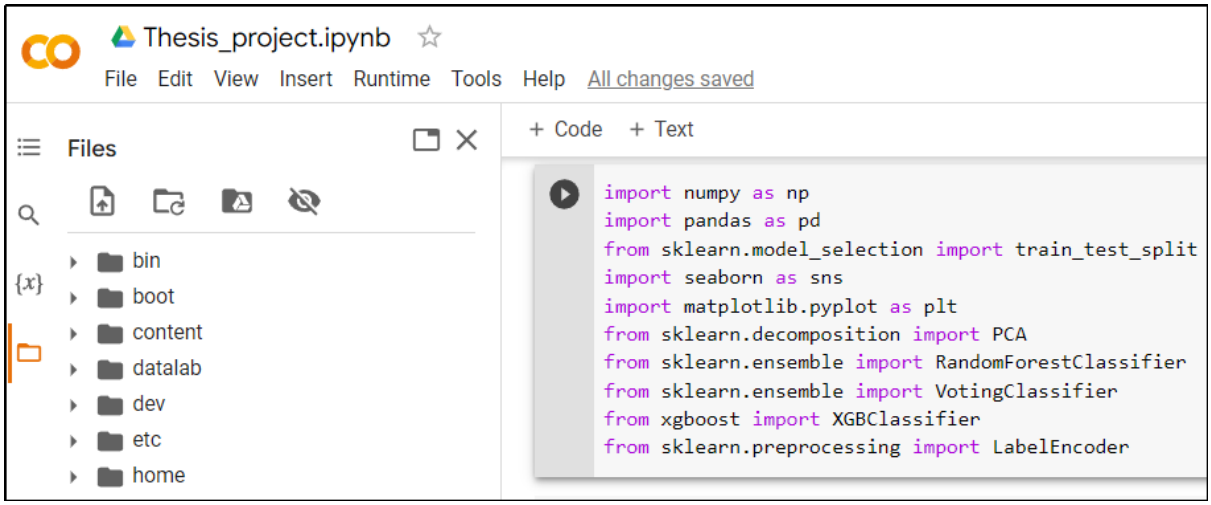

Figure 5: Required Libraries

The following is a list of the needed libraries for execution:

- Matplotlib
- Scipy
- Numpy
- Pandas

- Sci-kit learn
- Seaborn
- Plotly
- Xgboost
- Math
- Stats

4 Phases

The complete methodology of the research process, including data collecting, Exploratory Data Analysis is presented in the following paragraphs.

4.1 Data gathering

- Reading dataset using pandas.

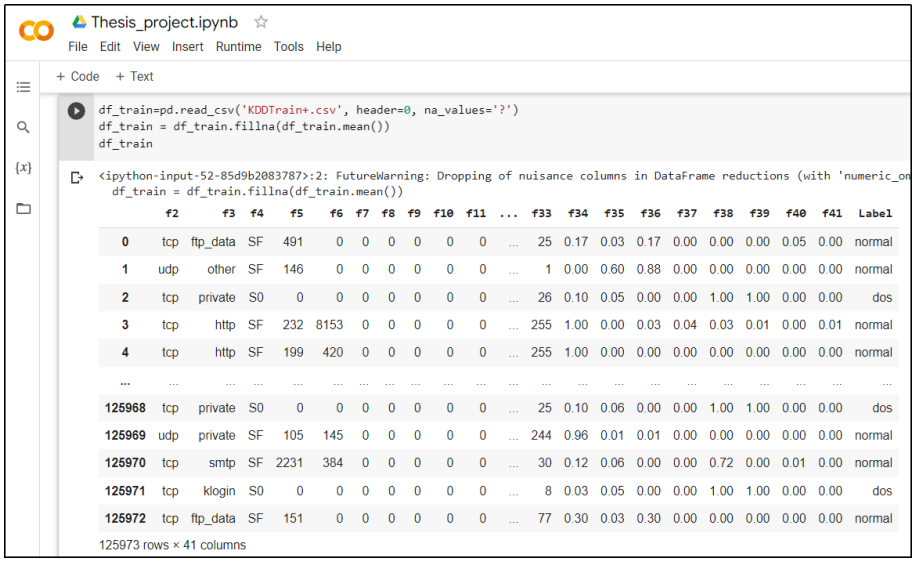

Figure 6: Reading

- Renaming the column names.

Figure 7: Renaming

- Display information of the dataset.

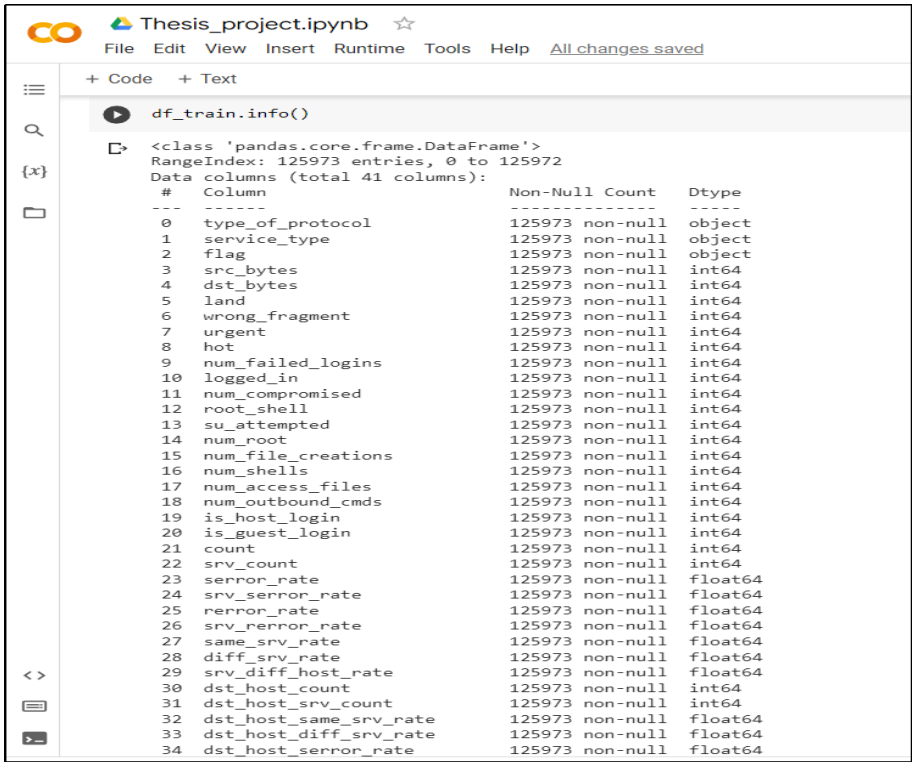

Figure 8: Content of dataset

4.2 Exploratory Data Analysis

- Displaying attack categorization bar graph:

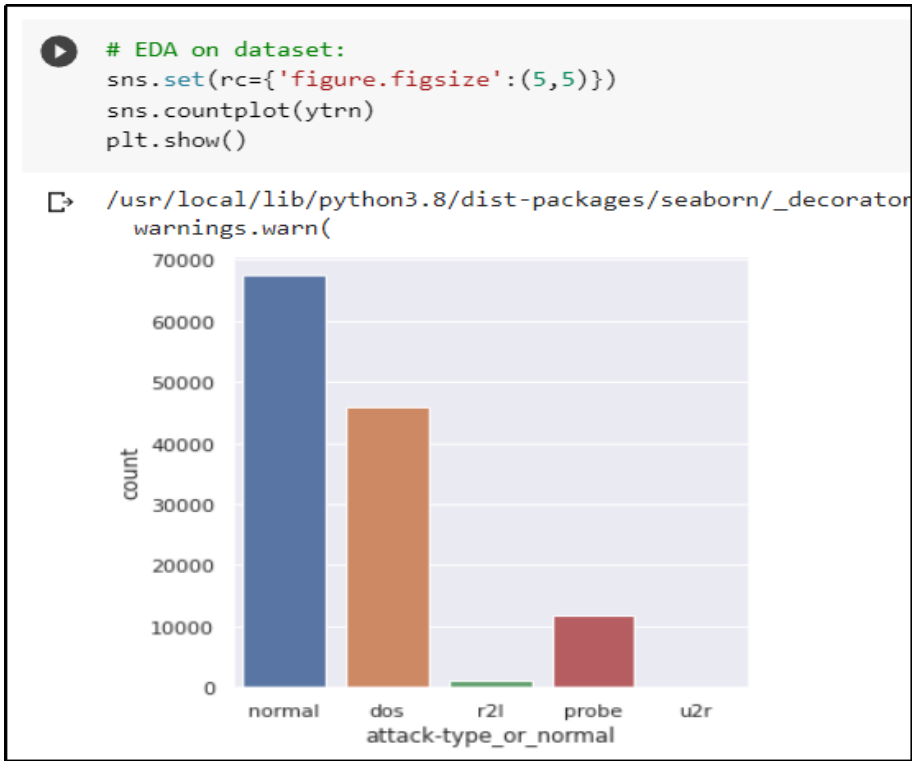

Figure 9: Attack Count

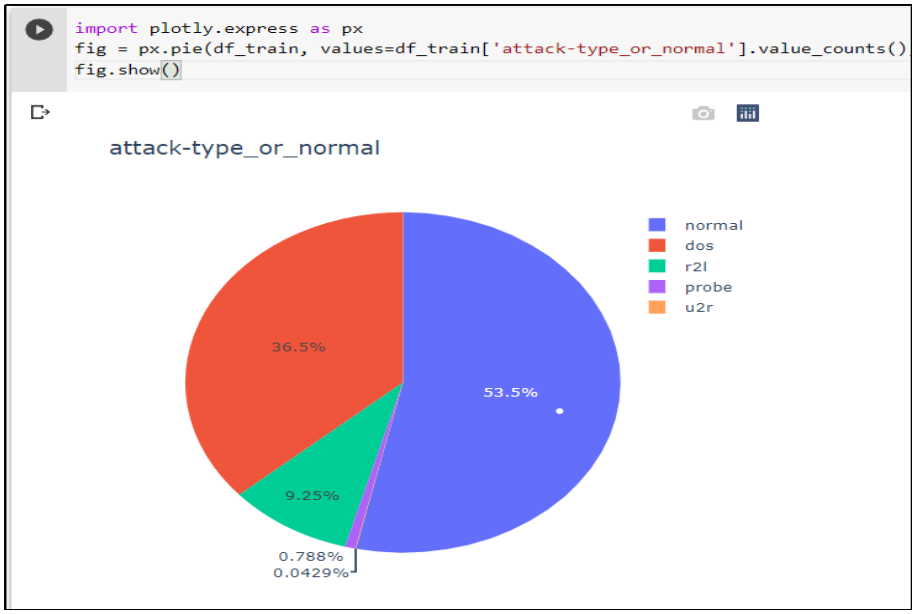

Figure 10: Attack percentage

- Displaying types of a protocol bar graph:

```
temp = df_train['type_of_protocol'].value_counts()
copy = temp.index
interchange = temp.values
zipped = list(zip(copy, interchange))
display = pd.DataFrame(zipped, columns=['type_of_protocol', 'Number_of_occurances'])
sns.barplot(data=display, x='type_of_protocol', y='Number_of_occurances')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcf88e9ee20>
```

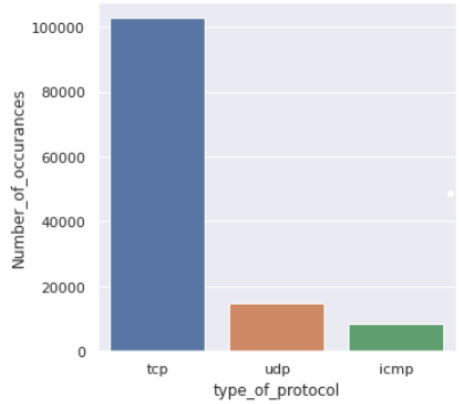

Figure 11: Protocol Type

5 Data Pre-processing

We have used label encoding for the pre-processing stage.

```
# Pre-processing Dataset
lbEn = LabelEncoder()
x['type_of_protocol']=lbEn.fit_transform(x['type_of_protocol'])
x['service_type']=lbEn.fit_transform(x['service_type'])
x['flag']=lbEn.fit_transform(x['flag'])
x['attack-type_or_normal']=lbEn.fit_transform(x['attack-type_or_normal'])

ytrn=lbEn.fit_transform(ytrn)
```

Figure 12: Label Encoding

6 Results

The accuracy score for the test data is 99.33 percent.

References

Radovanovic, I. (2022), 'Google colab - a step-by-step guide - algotrading101 blog'.
URL: *https://algotrading101.com/learn/google-colab-guide/*

The python tutorial (n.d.).
URL: *https://docs.python.org/3/tutorial/*