National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cybersecurity

Devika Rajiv Galinde
Student ID: 20177283

School of Computing
National College of Ireland

Supervisor: Prof. Arghir Nicolae Moldovan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Devika Rajiv Galinde

Student ID: 20177283

Programme: MSc in Cyber Security **Year:** 2023

Module: MSc Research Project

Lecturer: Prof. Arghir Nicolae Moldovan

Submission Due Date: 29/05/2023

Project Title: Effective approach for Malware Detection using Machine Learning and Deep Learning for IoT-Devices.

Word Count: 1016 **Page Count:** 7

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: Devika Rajiv Galinde

Date: 29/05/2023

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):	

Configuration Manual

Devika Rajiv Galinde
Student ID: 20177283

1 Introduction

The configuration manual consists of all the information regarding the tools as well as the technologies used for the entire research implementation. The Experimental Setup is as discussed in section 2. Technologies and Software tools for the implementations are discussed in section 3. The section 4 , shows the entire setup for the software tool and the implementation steps explained from importing libraries, data pre-processing , training and testing method for ML, DL models and then output as the Classification Report. Furthermore section 5, highlights the references for the software guide.

2 Experimental Setup

The experiment was conducted on the personal system where the experimental setup was made to further conduct the implementation.

- Hardware Specification: Processor of AMD Ryzen 7 @5700U with Radeon Graphics @1.80 GHz , 16GB of RAM.
- Windows Specification : Windows 10, 22H2v.
- Experimental Setup : Windows 10 , Anaconda 3.1, Jupyter NoteBook v6.4.12, with Python 3.9.13 and the Tensorflow of the 2.4 environments.

3 Technologies and Software used for Implementation

- Software used : Anaconda 3.1, Jupyter NoteBook v6.4.12, with Python 3.9.13 and the Tensorflow of the 2.4 environments.
- Anaconda is basically a distribution of the programming languages like Python and R, for the purpose of the scientific computing which is in the field of machine learning application, data science , data processing in the large-scale , predictive analysis, etc. It basically aims to simplify the package management and also the deployment [1] .
- Jupyter is an open source software which has open standards , free software to use , and have web services for the interactive computing for all the programming languages [2].

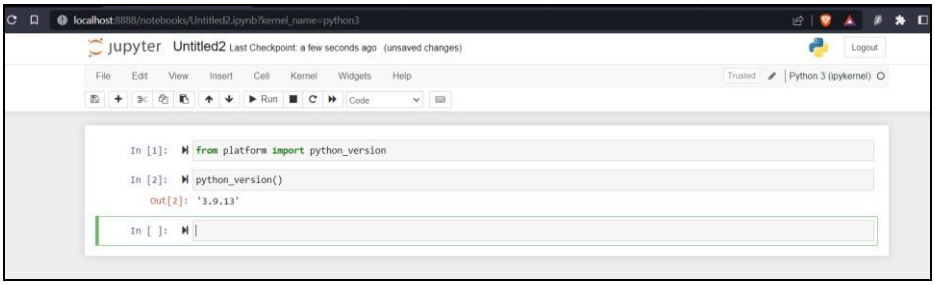

Figure 1: Python Version used in Jupyter Notebook.

4 Implementation

Step 1 : Anaconda was downloaded and installed.

Step 2 : Jupyter Notebook was installed and launched.

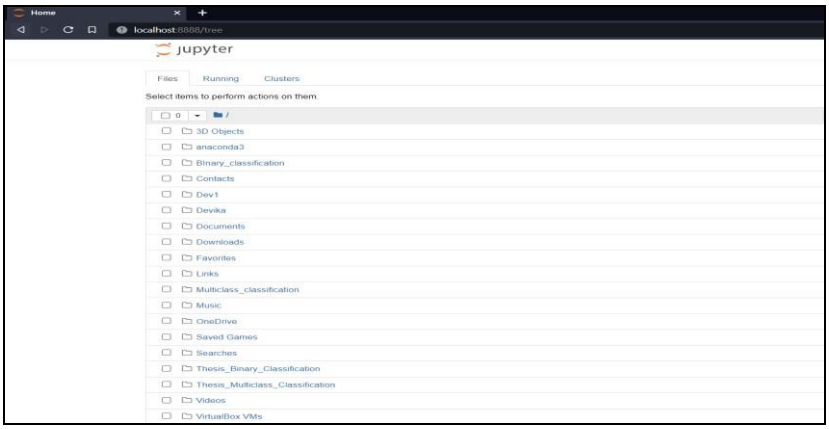

Figure 2 : Jupyter Notebook Home page.

Step 3 : The dataset should be downloaded initially.

Step 4 : The important libraries for the Jupyter Notebook should be imported for further any file execution such as pandas, Tenserflow , numpy, time , etc.

Step 5 : Importing the libraries for Machine Learning algorithms.

```
import pandas as pd
import numpy as np
import os
import re
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

Figure 3 : Importing basic libraries for ML algorithms

Step 6 : To perform the data pre-processing the dataset should be loaded as well as its proper file path should be given.

```
path = r'C:/Project/IOT23/opt/Malware-Project/BigDataset/IoTScenarios'
```

Figure 4 : File path for Dataset

```
all_files_path = []
for folder in os.listdir(path):
    subpath = path + '/' + folder
    # print(return_final_folder(subpath))
    all_files_path.append(return_final_folder(subpath))
```

Figure 5 : All file path list

Step 7 : All 20 malware captures are logs are read separately.

```
df34 = pd.read_table(filepath_or_buffer=all_files_path[6], skiprows=10, nrows=100000)
df34.columns=['ts',
              'uid',
              'id.orig_h',
              'id.orig_p',
              'id.resp_h',
              'id.resp_p',
              'proto',
              'service',
              'duration',
              'orig_bytes',
              'resp_bytes',
              'conn_state',
              'local_orig',
              'local_resp',
              'missed_bytes',
              'history',
              'orig_pkts',
              'orig_ip_bytes',
              'resp_pkts',
              'resp_ip_bytes',
              'label']
df34.drop(df34.tail(1).index,inplace=True)
```

Figure 6 : Reading log files separately

Step 8 : All the frames are then concatenated.

```
frames=[df1, df17, df20, df21, df3, df33, df34, df35, df36, df39, df42, df43, df44, df48, df49, df52, df60, df7, df8, df9]

df_c=pd.concat(frames)
```

Figure 7 : All frames concatenated

Step 9 : Generating the separate csv files (train, test and iot23_combines_14-features) as pre-processed data for model training and testing. (Note: only iot23_combines_14-features.csv is

used, also instead of this, the train and test csv files can also be used for the training and testing of models.)

```
# Shuffle the dataset
df = df_c.sample(frac=1).reset_index(drop=True)

# Split the dataset into train and test sets
train_size = int(0.8 * len(df))
train_df = df.iloc[:train_size, :]
test_df = df.iloc[train_size:, :]

# Save the train and test datasets as CSV files
train_df.to_csv('train.csv',index=False)
test_df.to_csv('test.csv',index=False)

df_c.to_csv('iot23_combined_14_features.csv',index=False)
```

Figure 8 : Separate csv files generated

Step 10 : The preprocessed data is then fed to the Machine Learning and Deep Learning Algorithms. The pre-processed file is then read.

```
filepath = r"C:\Users\devik\Devika\iot23_combined_14_features.csv"

df = pd.read_csv(filepath)
```

Figure 9 : Reading file path for all models

Step 11 : The models are trained and tested with the 80:20 ratio. Where X and Y are the total 14 data features which are pre-processed.

```
X = df[['id.orig_h', 'id.orig_p', 'id.resp_h', 'id.resp_p', 'proto', 'duration',
       'orig_bytes', 'resp_bytes', 'missed_bytes', 'orig_pkts',
       'orig_ip_bytes', 'resp_pkts', 'resp_ip_bytes']]
Y = df['label']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, random_state=10, test_size=0.2)
```

Figure 10 : Training and Testing data

Step 12 : The Classification report for all the algorithms is printed.

```
print("Classifiction Report :")
print(classification_report(Y_test, y_pred))
```

Figure 11 : Output in form of Classification Report

5 References

[1] “Anaconda (Python distribution),” *Wikipedia*. Mar. 29, 2023. Accessed: May 01, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Anaconda_(Python_distribution)&oldid=1147257152

[2] “Project Jupyter.” https://jupyter.org (accessed May 01, 2023).