National College of Ireland

Configuration Manual

MSc Research Project
Cyber Security

Aksa Anna Shajan
Student ID: 20178875

School of Computing
National College of Ireland

Supervisor: Dr. Vanessa Ayala-Rivera

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……. Aksa Anna Shajan……

Student ID: ……20178875……..……

Programme: ……Cyber Security…… **Year:** …2021-2022

Module: …MSc Research Project……..……

Lecturer: ……Dr. Vanessa Ayala-Rivera……..……

Submission Due Date: …16 Dec 2021……..……

Project Title: Intrusion detection in IoT devices using zero bias DNN……..……

Word Count: …591…… **Page Count:** ……4……..……..……

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: Aksa Anna Shajan……

Date: 16 Dec 2021……

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

Aksa Anna Shajan
Student ID: 20178875

1 Introduction

This configuration manual aims to explain the implementation procedure including software and hardware requirement of the project – Intrusion detection in IoT devices using Zero-Bias DNN.

2 Hardware - Software requirements

A system with good specifications results in better performance. The following are the hardware requirements:

- A processor type of 11th gen Intel core i7 processor.
- A processor speed up to 4.2 GHz
- A 16 GB of RAM
- A 1 TB of SSD
- An OS of windows 10 (64 bit)

The following are the software requirements:
The code of the project is written in the python programming language. Anaconda is free and open-source software with many packages for machine learning, was downloaded and installed from the official website. Jupyter notebook IDE was used to write the python codes.

3 Implementation procedures

Figure 1 shows the main imported packages and libraries.

```
import pandas as pd
import numpy as np
import sys
import sklearn
import io
import random
from sklearn.feature_selection import RFE
from sklearn.ensemble import RandomForestClassifier
import pickle

from sklearn.metrics import accuracy_score,classification_report
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt
import itertools
```

Figure 1

Versions of Libraries:
Python version of 3.8.5 is used in our model.
TensorFlow and Keras version is 2.5.0. The NumPy version is 1.19.5.

The MQTT-IOT-IDS2020 dataset was used for the model. The dataset contains normal and abnormal cases of MQTT protocol. The dataset was available on the IEEEport public platform. The dataset was cleaned as the preprocessing step.
The code below is for the creation of DNN architecture(Fig 2).

```
import tensorflow as tf
from tensorflow.keras import initializers


def model1(n):

    # bias_initializer = initializers.Zeros()
    input_shape=(1,n)
    model = tf.keras.Sequential(
        [
            # tf.keras.Input(shape=input_shape),
            tf.keras.layers.Conv1D(6,activation="relu",kernel_size=1,use_bias=True,bias_initializer='zeros',input_shape=input_shape
            tf.keras.layers.Flatten(),
            tf.keras.layers.Dense(10, activation="relu",use_bias=True,bias_initializer='zeros'),
            tf.keras.layers.Dense(1, activation="sigmoid"),
        ]
    )
    model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
    return model
```

Figure 2

The model was trained using the MQTT-IoT-IDS2020 dataset.

4 Execution procedures

Activate the environment and run the code for GUI.

```
Anaconda Prompt (project 3 sem) - python  gui.py

(base) C:\Users\Aksa Anna Shajan>activate env_torch

(env_torch) C:\Users\Aksa Anna Shajan>cd Desktop\Intrusion_aksa

(env_torch) C:\Users\Aksa Anna Shajan\Desktop\Intrusion_aksa>python gui.py
2021-12-08 19:13:58.826678: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that thi
s TensorFlow binary was not compiled to use: AVX2
Model loaded from disk
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 1, 13)             182
_________________________________________________________________
flatten (Flatten)            (None, 13)                0
_________________________________________________________________
dense (Dense)                (None, 1000)              14000
_________________________________________________________________
dense_1 (Dense)              (None, 100)               100100
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 101
=================================================================
Total params: 114,383
Trainable params: 114,383
Non-trainable params: 0
_________________________________________________________________
['Unnamed: 0', 'ip_len', 'ip_flag_rb', 'src_port', 'dst_port', 'mqtt_messagelength', 'mqtt_flag_uname', 'mqtt_flag_passw
d', 'mqtt_flag_retain', 'mqtt_flag_qos', 'mqtt_flag_willflag', 'mqtt_flag_clean', 'mqtt_flag_reserved']
```

Figure 3

Running the code will display a screen as shown below.

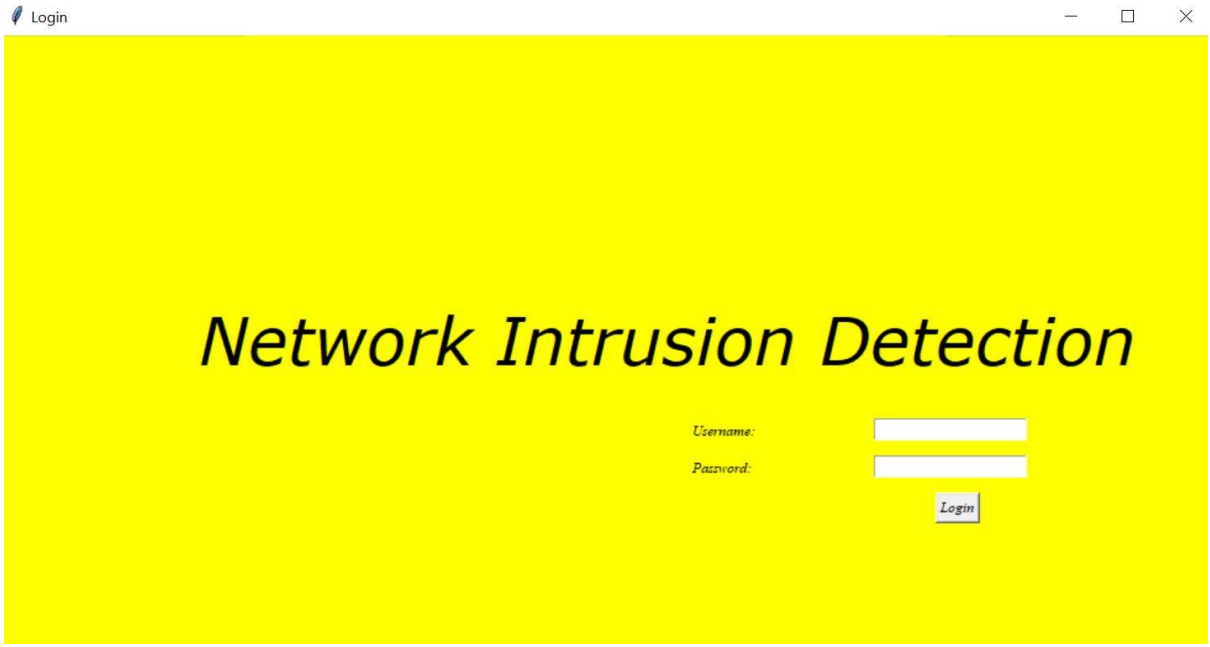

Figure 4

Logging into the page provides a box to enter the data, and it will display the result as intrusion detected or the normal condition depending upon the data we have entered.

The figure 5 shows the result of entering data of an attack pattern.

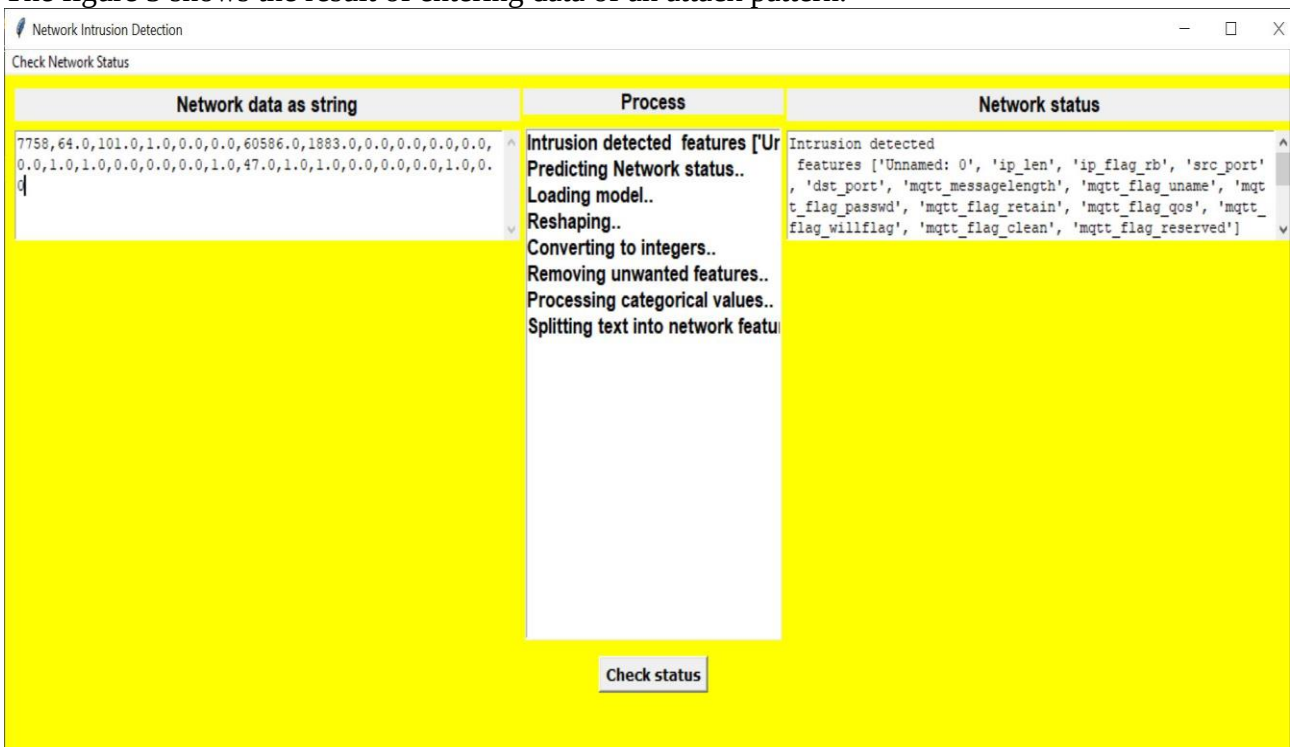

Figure 5

The figure 6 shows the result of a normal condition.

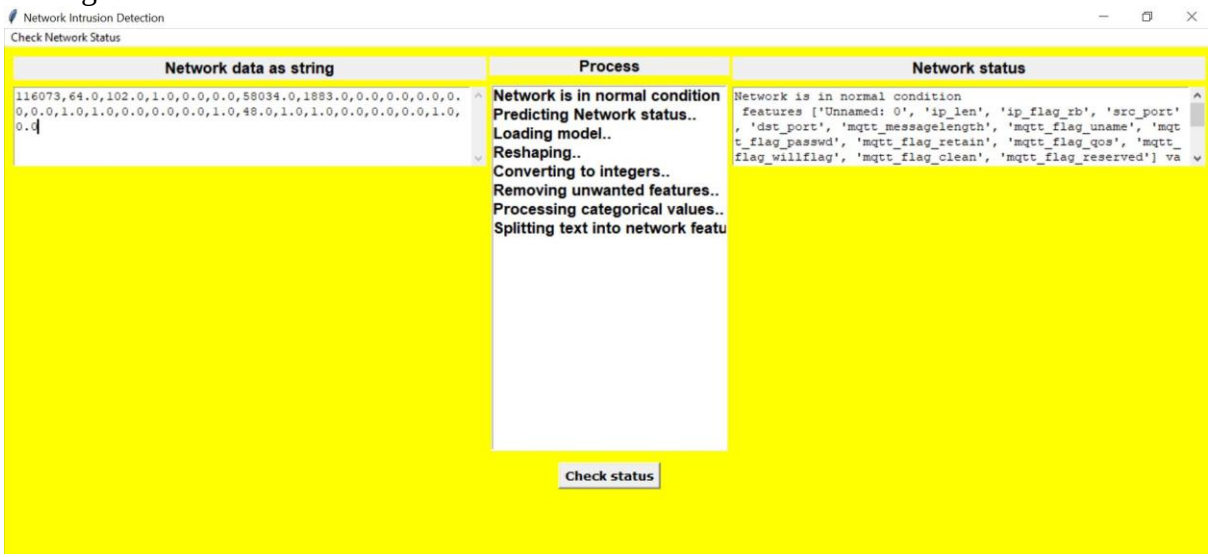

Figure 6

References

[1] *Anaconda Documentation — Anaconda documentation* (no date). Available at: https://docs.anaconda.com/ (Accessed: 8 December 2021).

[2] Hindy, H. (2020) 'MQTT-IoT-IDS2020: MQTT Internet of Things Intrusion Detection Dataset'. IEEE. Available at: https://ieee-dataport.org/open-access/mqtt-iot-ids2020-mqtt-internet-things-intrusion-detection-dataset (Accessed: 8 December 2021).