# Configuration Manual

MSc Research Project

Cyber Security


Elizabeth Mughogho

Student ID: x22224343

School of Computing

National College of Ireland


Supervisor: Ross Spelman

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | |
|---|---|
| **Student Name:** | Elizabeth Mughogho ……………………………………………………………………………………………… |
| **Student ID:** | x22224343 ……………………………………………………………………………………………….. |
| **Programme:** | MSc Cyber Security ……………………………………………………………… **Year:** 2025 ……………………….. |
| **Module:** | MSc Research Project ………………………………………………………………………………………….… |
| **Lecturer:** | Ross Spelman ………………………………………………………………………………………….… |
| **Submission Due Date:** | 24/04/2025 ………………………………………………………………………………………….…… |
| **Project Title:** | Enhancing IoT Network Security Through Intrusion Detection Using Machine Learning ………………………………………………………………………………………….…… |
| **Word Count:** | ……………………………………… **Page Count:** |

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:** Elizabeth Mughogho

**Date:** 24/04/2025 …………………………………………………………………………………………

**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

Elizabeth Mughogho
x22224343

## 1. Introduction

To conduct this study, I used the CIC-IoT2023 dataset to set up the experiment. This manual will walk you through setting up your system to support the project, preventing problems from arising during implementation, and producing a better result. The manual will cover the necessary steps and tools for the project, including the necessary hardware and software used to carry out the experiment.

## 2. System Specification

The configuration of the system used in the project is:

- Operating System:Windows 10/11
- RAM: minimum 8GB
- Processor: 2.5 GHz and above
- Hard drive: 256GB

## 3. Software Tools

The project was implemented using the following tools:

- Python 3.8 above
- Jupyter Notebook
- Google Colab( https://colab.google/)
- Anaconda(https://www.anaconda.com/download)
- Visual studio

## 3.1 Software Installation

Description of the steps taken in installation the tools

- Python v3.11 can be downloaded from: https://www.python.org/downloads/windows/

## 4. Implementation

For implementation the below libraries were used for the project:

- Numpy
- Seaborn
- Pandas
- Matplotlib

# Implementation on the Network Intrusion Detection Dataset

1. Imported libraries for data visualization and models and installation of missing package

C: > Users > Eli > Desktop > Project > Thesis > Ensemble_IDS_IotID2023.ipynb > from google.colab import drive

Generate + Code + Markdown | Run All | Outline ···

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn import feature_extraction, linear_model, model_selection, preprocessing
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings("ignore")
```

```
(base) C:\Users\Eli\.anaconda>conda install -c conda-forge xgboost
```

Fig 1: Importing the libraries

2. Data was loaded on the notebook

```
# Load the dataset from a CSV file
df1 = pd.read_csv('C:/Users/Eli/Desktop/Project/Thesis/CIC IoT2023.csv')
```

MS 86 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER

Fig 2: Importing the dataset

3. Instance for label Encoding

```
from sklearn.preprocessing import LabelEncoder
# Create an instance of LabelEncoder
labelencoder = LabelEncoder()
df['label'] = labelencoder.fit_transform(df['label'])
✓ 0.1s
```

Fig3: Label encoding

```
y = df['label'].values.reshape(-1,1)
df.drop(["label"],axis=1,inplace=True)
features = df.dtypes[df.dtypes != 'object'].index
df[features] = df[features].apply(
    lambda x: (x - x.mean()) / (x.std()))
# Fill empty values by 0
df_pre = df.fillna(0)
print('=====Data Preprocessing=========')
print(df_pre)
X = df_pre
y=np.ravel(y)
```

Fig 4:Checking for null values in the dataset

```
X_train, X_test, Y_train, Y_test = train_test_split(X_fs, y, test_size =0.2)
AX_train=X_train
AX_test=X_test
AY_train=Y_train
AY_test=Y_test
✓ 0.1s
```

Fig 5: Data splitting

4. The below shows the different machine learning algorithms used in my model selection with calibrated probabilities to help improve the accuracy and reliability of the model prediction

```
xg = xgb.XGBClassifier(n_estimators = 10)
xg.fit(X_train,Y_train)
from sklearn.calibration import CalibratedClassifierCV
calibrated_classifier_xg = CalibratedClassifierCV(xg, method='sigmoid', cv='prefit')
calibrated_classifier_xg.fit(X_train, Y_train)
y_predict=calibrated_classifier_xg.predict(X_test)
y_true=Y_test
```

```
Accuracy of xgboost: 0.9901755414973397
Precision of xgboost: 0.989647531986721
Recall of xgboost: 0.9901755414973397
F1-score of xgboost: 0.9898492735038285
```

Fig 6: XGBoost Classifier

```
rf = RandomForestClassifier(random_state = 0)
rf.fit(X_train,Y_train)
# fit classifier to training set
from sklearn.calibration import CalibratedClassifierCV
calibrated_classifier_RF = CalibratedClassifierCV(rf, method='sigmoid', cv='prefit')
calibrated_classifier_RF.fit(X_train, Y_train)
y_predict=calibrated_classifier_RF.predict(X_test)
y_true=Y_test
from sklearn.metrics import accuracy_score,precision_recall_fscore_support
test_accuracy=accuracy_score(y_true, y_predict)
print('Accuracy of RF: '+ str(test accuracy))
```

```
Accuracy of RF: 0.9907201809878923
Precision of RF: 0.9900228234117687
Recall of RF: 0.9907201809878923
F1-score of RF: 0.9900420604908385
```

Fig 7. Random Forest Classifier

```
# fit classifier to training set
from sklearn.calibration import CalibratedClassifierCV
calibrated_classifier_DT = CalibratedClassifierCV(DT, method='sigmoid', cv='prefit')
calibrated_classifier_DT.fit(X_train, Y_train)
y_predict=calibrated_classifier_DT.predict(X_test)
y_true=Y_test
from sklearn.metrics import accuracy_score,precision_recall_fscore_support
test_accuracy=accuracy_score(y_true, y_predict)
print('Accuracy of DT: '+ str(test_accuracy))
```

```
Accuracy of DT: 0.9910762914240228
Precision of DT: 0.9912262888219049
Recall of DT: 0.9910762914240228
F1-score of DT: 0.9911313679263002
```

Fig 8. Decision Tree

```
# define the voting ensemble
ensemble = VotingClassifier(estimators=models, voting='soft')
ensemble.fit(X_train,Y_train)
ensemble_score=ensemble.score(X_test,Y_test)
y_predict=ensemble.predict(X_test)
y_true=Y_test
```

```
Accuracy of ensemble: 0.9925216808412586
Precision of ensemble: 0.9921574795979001
Recall of ensemble: 0.9925216808412586
F1-score of ensemble: 0.9922477691653331
```

Fig 9 . Ensemble

The code below aims to simplify the classification of network traffic by grouping all traffic into two categories: BenignTraffic and MaliciousTraffic

```
df1['label'] = df1['label'].apply(lambda x: 'BenignTraffic' if x == 'BenignTraffic' else 'MaliciousTraffic')
```

Fig 10. models that predict one of two outcomes

```
X_train, X_test, y_train, y_test = train_test_split(X,y, train_size = 0.8, test_size = 0.2, random_state = 0,stratify
from sklearn.feature_selection import mutual_info_classif
importances = mutual_info_classif(X_train, y_train)
```

```
=====Network Traffic Data=========
        flow_duration  Header_Length  Protocol Type  Duration         Rate  \
0            0.000000          54.00           6.00     64.00     0.329807
1            0.000000          57.04           6.33     64.00     4.290556
2            0.000000           0.00           1.00     64.00    33.396799
3            0.328175       76175.00          17.00     64.00  4642.133010
4            0.117320         101.73           6.11     65.91     6.202211
...               ...            ...            ...       ...          ...
238682       0.000000          54.00           6.00     64.00     3.049186
238683       0.000000          54.00           6.00     64.00   183.433732
238684       0.000785          56.29           6.11     64.00   306.952216
238685       0.000901          72.09           6.11     64.64   158.475986
238686       0.000000           0.00           1.00     64.00     1.291274

              Srate  Drate  fin_flag_number  syn_flag_number  rst_flag_number  \
0          0.329807    0.0              1.0              0.0              1.0
1          4.290556    0.0              0.0              0.0              0.0
2         33.396799    0.0              0.0              0.0              0.0
3       4642.133010    0.0              0.0              0.0              0.0
4          6.202211    0.0              0.0              1.0              0.0
...             ...    ...              ...              ...              ...
238682     3.049186    0.0              1.0              0.0              1.0
238683   183.433732    0.0              0.0              0.0              0.0
238684   306.952216    0.0              0.0              1.0              0.0
```

Fig 11. Data splitting

**Alternatively follow the below steps for setting up the environment in Google colab.**

- One google colab in google chrome navigate to(https://colab.google/).
- Select to upload the .ipynb file from your computer or from google drive
- Once the file is uploaded, connect to the python server.
- If connected status will show on the top right hand of the screen and the bottom as connected.

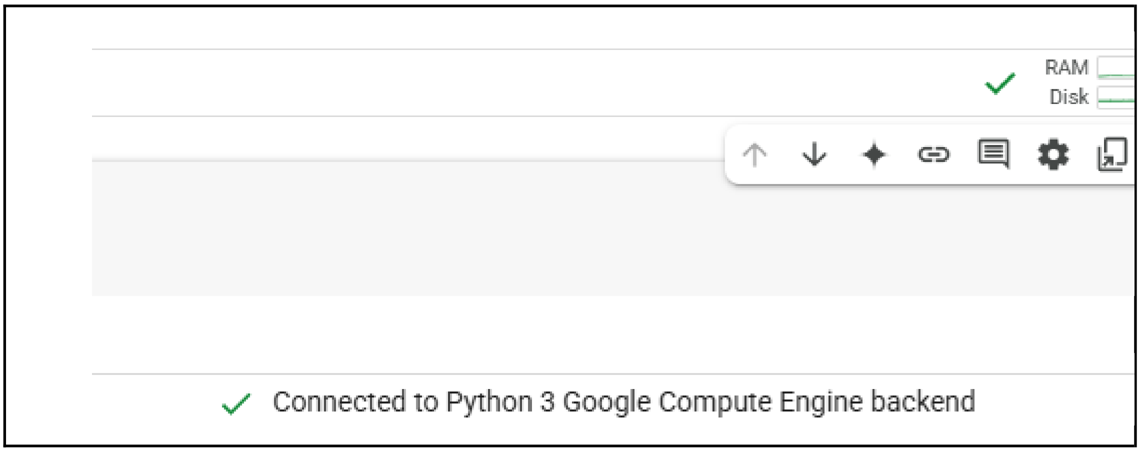


Fig 7 google colab connection status

- Then upload the CIC IoT2023.csv file into the colab

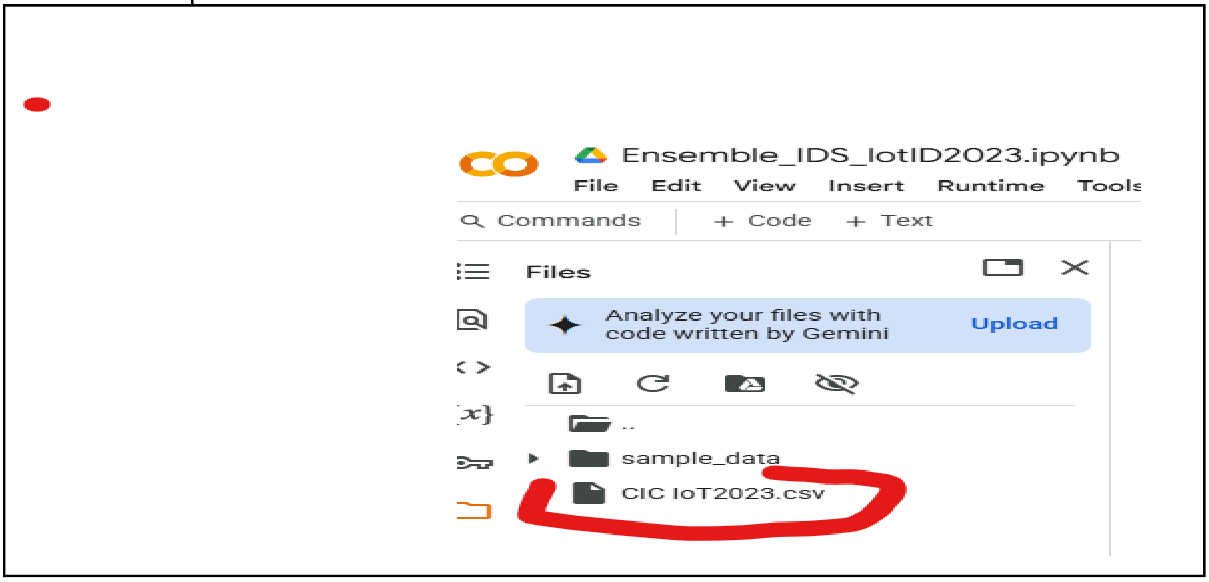


Fig 8 csv file upload in google colab

- Once the file has been uploaded make sure you run the below code to give access to the google drive where your csv file and the Ensemble_IDS_IoTD2023.ipynb is located

```
from google.colab import drive
drive.mount('/content/gdrive')
import os
os.chdir("/content/gdrive/MyDrive")
```

Fig 5 colab google importer

It will ask for permission to access your gdrive, make sure the access is granted for the code to run correctly.

**2. Dataset**

This study made use of the CIC-IoT2023 dataset, a comprehensive and labeled dataset specifically designed for IoT-based network intrusion detection. The dataset can be downloaded  from kaggle: ([https://www.kaggle.com/datasets/madhavmalhotra/unb-cic-iot-dataset](https://www.kaggle.com/datasets/madhavmalhotra/unb-cic-iot-dataset))

Table 1. Data description

| Description | Licence |
|---|---|
| This dataset is from the University of New Brunswick Centre for Cybersecurity.<br>It has extracted CSV features on network traffic across 105 Internet of Things (IoT) devices with 33 cyberattacks run on them. 7 types of attacks were run: distributed denial of service (DDoS), denial of service (DoS), reconnaissance, web-based, brute-force, spoofing, and the Mirai botnet. | The data can be republished and it’s allowed to mirror it with the datasets although anyone wishing to republish the data must put a citation of the data and the research paper listed on the webpage. |

## 6. References

[1] Malhotra, M. (2023) UNB CIC IOT 2023 dataset, Kaggle. Available at: https://www.kaggle.com/datasets/madhavmalhotra/unb-cic-iot-dataset (Accessed: 24 January 2025).

[2] Nickjeffrey (no date) Ensemble_learning/cic_iot_dataset2023.ipynb at main · Nickjeffrey/ensemble_learning, GitHub. Available at: https://github.com/nickjeffrey/ensemble_learning/blob/main/CIC_IOT_Dataset2023.ipynb (Accessed: 16 January 2025).