

Configuration Manual

MSc Research Project
M.Sc Cybersecurity

Haroon Ali Mohamed Ibrahim Maraicar
22186549

School of Computing
National College of Ireland

Supervisor: Prof. Imran Khan

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Haroon Ali Mohamed Ibrahim Maraicar
Student ID: 22186549
Programme: Msc Cybersecurity **Year:** 2023-2024
Module: Msc Research Project (Practicum)
Supervisor: Prof. Imran Khan
Submission Due Date: 12-08-2024
Project Title: Enhancing Network Security by Detecting Rogue Access Points using Ensemble Machine Learning Algorithms
Word Count: 609 **Page Count:** 5

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: Haroon Ali

Date: 11-08-24

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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

Haroon Ali Mohamed Ibrahim Maraicar
22186549

1 Introduction

The following manual provides instructions on how to write and execute the code for detecting unauthorised access points using machine learning. The application was coded using the Python programming language. This document provides a detailed explanation of the necessary configurations and software tools needed to recreate the experimental arrangement for the purpose of development.

2 System Specifications

The model that identifies Rogue Access Points present in a wireless network was built on a computer that has the following specifications:

- ASUS TUF F15 Gaming Laptop
- Intel Core -i5 – 10500H
- 24GB RAM
- 1 TB SSD
- NVIDIA GeForce GTX Graphics card
- Windows 11 Operating System

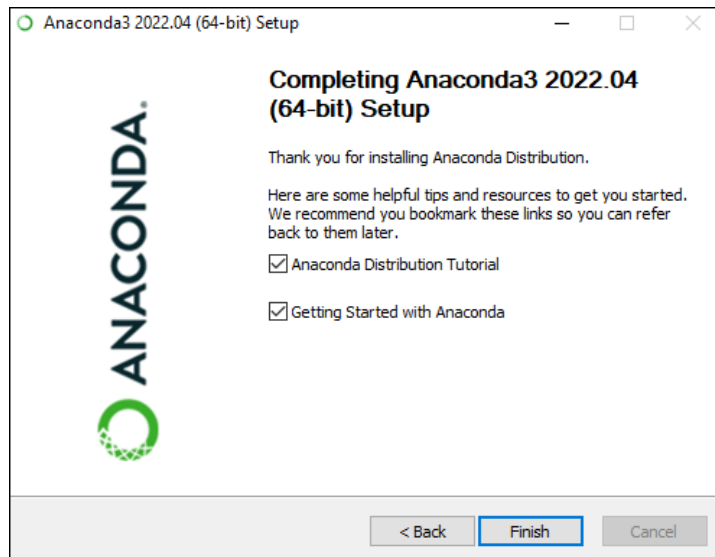
The softwares that were used in this research are:

- Jupyter Notebook version 7.08
- Google Colab 3.10
- VSCode version 1.92.0
- Python version 3.11.5

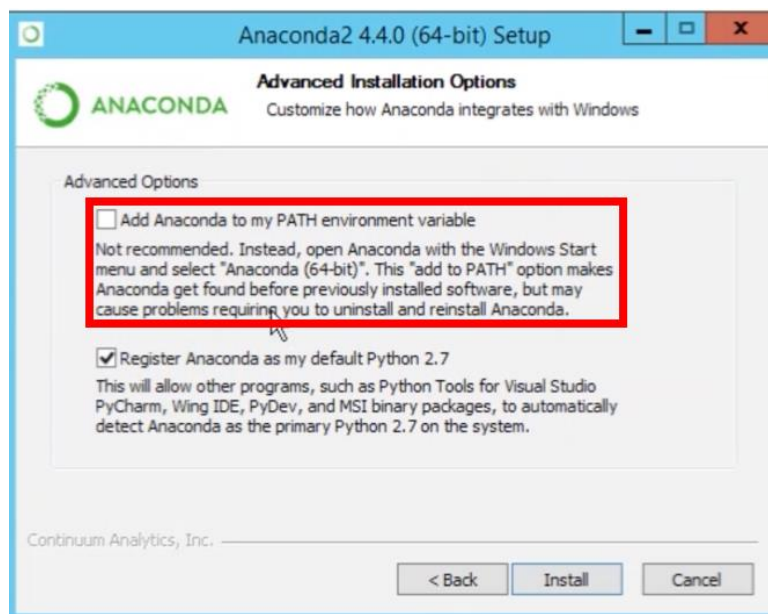
3 Software Setup

This section contains the instructions to download and install all the necessary software applications.

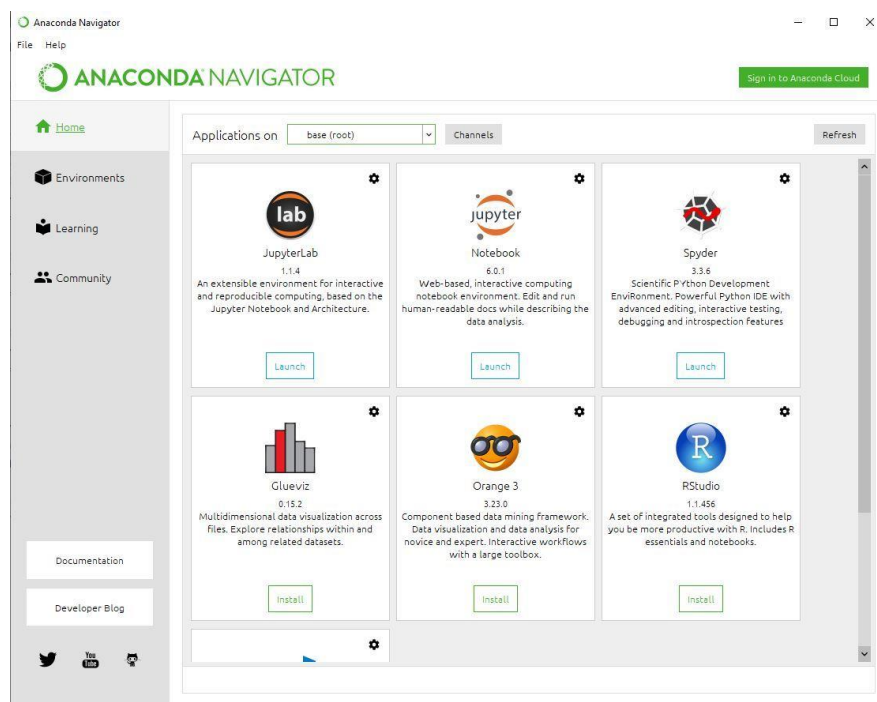
1. Python is a prerequisite to have this machine learning models trained. Jupyter notebook was used from Anaconda Navigator to run the codes. This software is available in the website <https://www.anaconda.com/download/success>



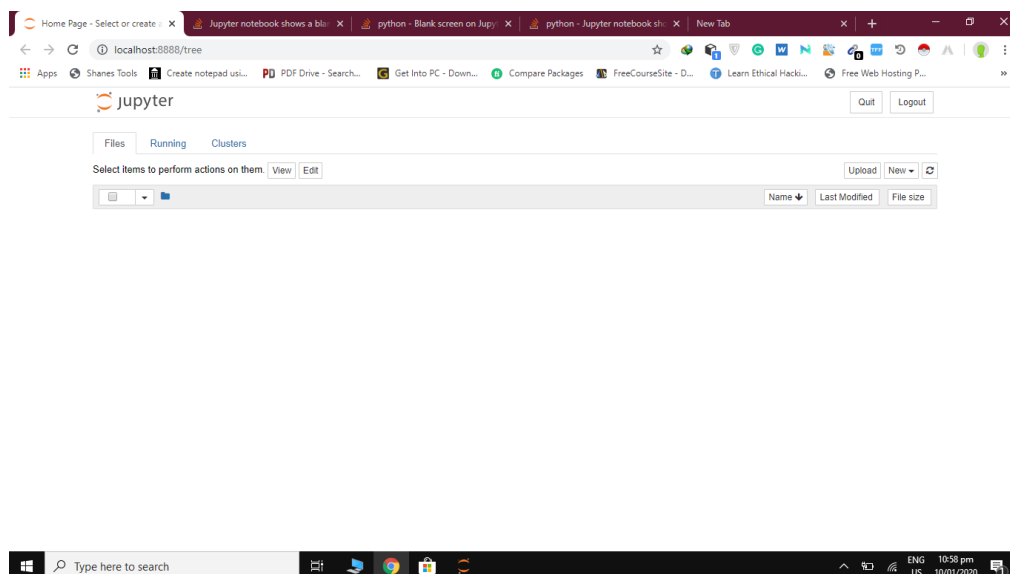
2. Add Anaconda to my Path environment variable



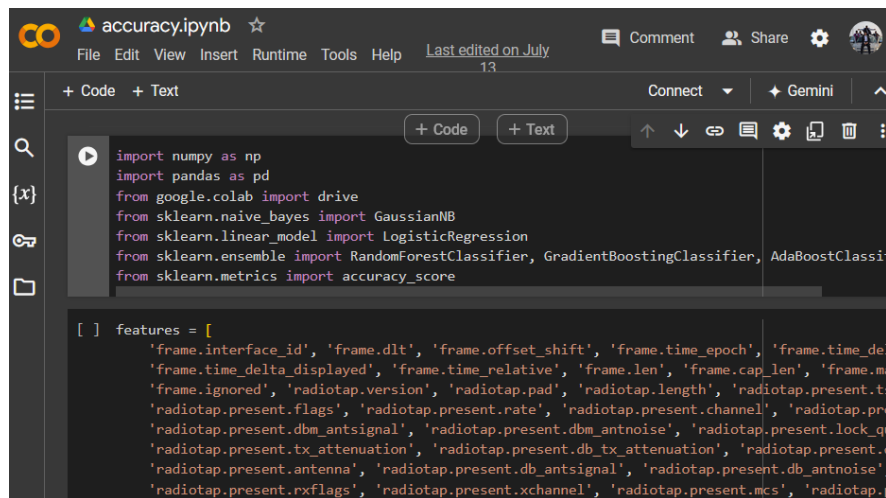
3. Once Anaconda has been installed, we can then launch Jupyter notebook from there.



4. Jupyter notebook to run the code



5. The code was also executed on Google Colab



```
import numpy as np
import pandas as pd
from google.colab import drive
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier
from sklearn.metrics import accuracy_score

[ ] features = [
    'frame.interface_id', 'frame.dlt', 'frame.offset_shift', 'frame.time_epoch', 'frame.time_delta',
    'frame.time_delta_displayed', 'frame.time_relative', 'frame.len', 'frame.cap_len', 'frame.mtu',
    'frame.ignored', 'radiotap.version', 'radiotap.pad', 'radiotap.length', 'radiotap.present.ts',
    'radiotap.present.flags', 'radiotap.present.rate', 'radiotap.present.channel', 'radiotap.present.dbm',
    'radiotap.present.dbm_antsignal', 'radiotap.present.dbm_antnoise', 'radiotap.present.lock_quality',
    'radiotap.present.tx_attenuation', 'radiotap.present.db_tx_attenuation', 'radiotap.present.db_tx_antnoise',
    'radiotap.present.antenna', 'radiotap.present.db_antsignal', 'radiotap.present.db_antnoise',
    'radiotap.present.rxflags', 'radiotap.present.xchannel', 'radiotap.present.mcs', 'radiotap.p
```

4 Python Packages used

- Pandas: The dataset is used to read the data set.
- Numpy: For array operations.
- Sklearn: for categorization, regression, clustering, and dimensionality reduction are all examples of statistical modelling.
- Matplotlib: Python Visualization using Matplotlib

5 Loading the Dataset

Aegean Wifi Intrusion Dataset (AWID) was used to train the models. Two versions of this dataset was used in this research. One had 10 class labels with no test and train split datasets. The other had only 4 class labels with test and train split datasets.

```
# Load and preprocess the training data
awid = pd.read_csv("1", header=None, names=features, encoding='latin-1')
```

```
# Load and preprocess the test data
awid_test = pd.read_csv("test/1", header=None, names=features)
```

6 Importing Libraries

The following libraries were imported and used.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
```

7 Model evaluation

This function was used to evaluate all the machine learning models.

```
# Function to evaluate a model
def evaluate_model(model, X_test, y_test):
    y_preds = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_preds)
    precision = precision_score(y_test, y_preds, average='weighted')
    recall = recall_score(y_test, y_preds, average='weighted')
    f1 = f1_score(y_test, y_preds, average='weighted', zero_division=0)
    conf_matrix = confusion_matrix(y_test, y_preds)
    return accuracy, precision, recall, f1, conf_matrix
```