Configuration Manual

MSc Research Project
Cyber Security

Tushar Sanjaykumar Vaidya
X20254083

School of Computing
National College of Ireland

Supervisor: Prof. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Tushar Sanjaykumar Vaidya

Student ID: X20254083

Programme: MSc Cybersecurity **Year:** 2022 -2023

Module: MSc Research Project

Lecturer: Imran Khan

Submission Due Date: 01/02/2023

Project Title: Identifying inappropriate access points using machine learning algorithms RandomForest and KNN

Word Count: 942 **Page Count:** 10

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: Tushar Sanjaykumar Vaidya

Date: 01/02/2023

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

Tushar Sanjaykumar Vaidya
Student ID: X20254083
MSc in Cybersecurity
National College of Ireland

1 Introduction

This article describes how to develop and run the Identifying improper access points using machine learning code. The application was written throughout the programming language named Python. It explains each of the settings and software tools required to duplicate the development's experimental configuration.

2 System Specification

The system that detects the unauthorized access points in a wireless was created on:

- Processor: i7 processor
- Operating System:
- RAM: 8GB RAM
- Hard Drive: 512 GB

3 Software Tools

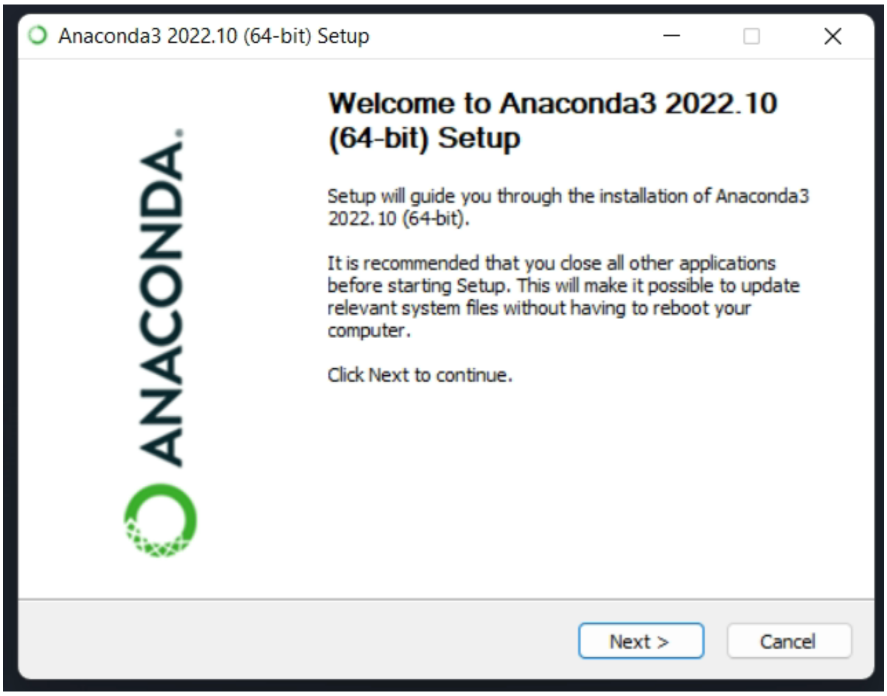
The following software tools were utilized to carry out this study:

- Anaconda
- Python
- Tkinter is the Python library is used for creating designing the desktop application
- Sublime text editor

3.1 Setup of Software

This section describes the steps necessary to install the tools.

1. On the machine, the Anaconda program has been downloaded and installed. It is available for download from the official website listed here. https://www.anaconda.com/products/individual.
2. Once you downloaded then follow the basic steps to installed it.



Anaconda3 2022.10 (64-bit) Setup

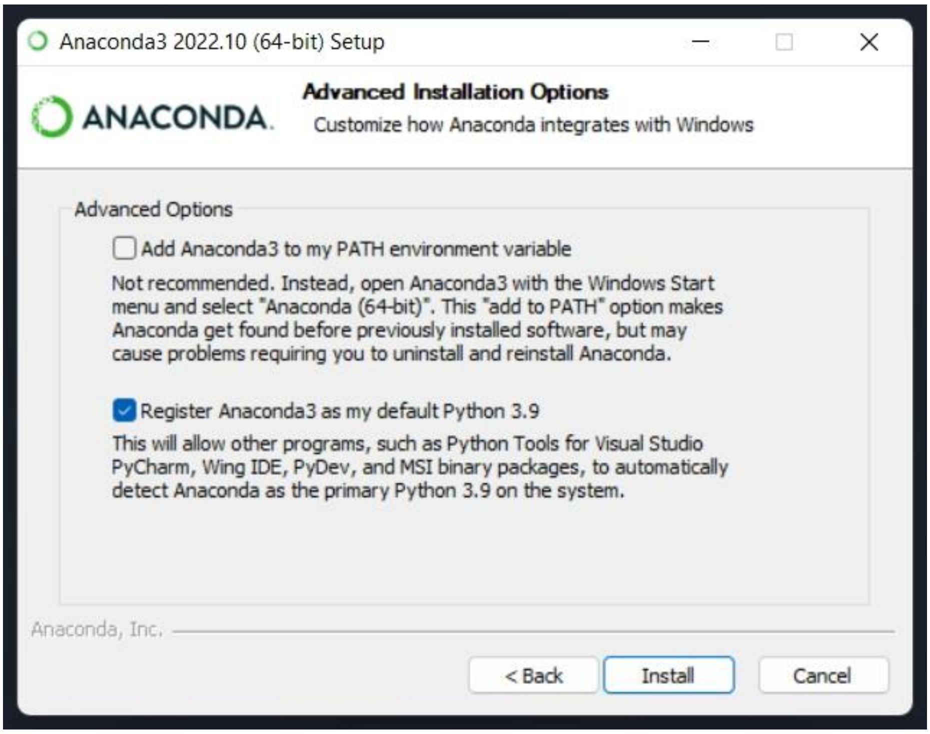
Welcome to Anaconda3 2022.10 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 2022.10 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

3. Check the selections just at the stage depicted in the figure below.



Anaconda3 2022.10 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

☐ Add Anaconda3 to my PATH environment variable

Not recommended. Instead, open Anaconda3 with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda3 as my default Python 3.9

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.9 on the system.

4. The sublime utility downloaded from this link: https://www.sublimetext.com/3

3.2 Package details

The environment "int_detect" is created in anaconda. It is made up of custom-installed machine learning python libraries that aid in the execution of the entire program. The following libraries are among those found in the environment:

- Matplotlib: Python Visualization using Matplotlib
- Pickling: Used in the serialization and deserialization of Python object structures
- Pandas: The dataset is used to read the data set.
- Numpy: For array operations.
- Sklearn: It is used for categorization, regression, clustering, and dimensionality reduction are all examples of statistical modeling.

4 Setting up the environment

- Extract IAP_Fullcode.rar compress file
- Extract the int_detect.rar folder and paste it into C:\ProgramData\Anaconda3\envs
- In the project folder, run anaconda prompt.
- Enter the command "activate int_detect" at the prompt.

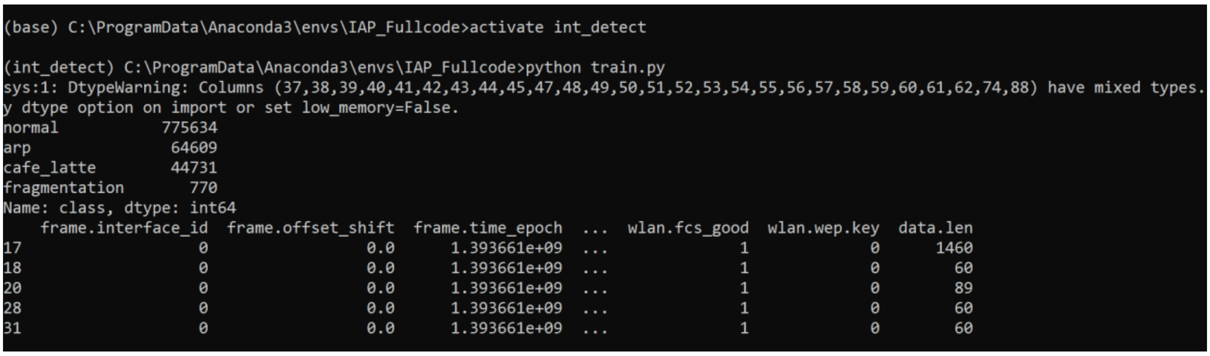
5 Dataset Source

The dataset for this research topic was obtained from github, an open platform that allows users to view and download multiple dataset collections. The sample was obtained and saved as Project_Dataset in the IAP_Fullcode folder.

6 Code Execution

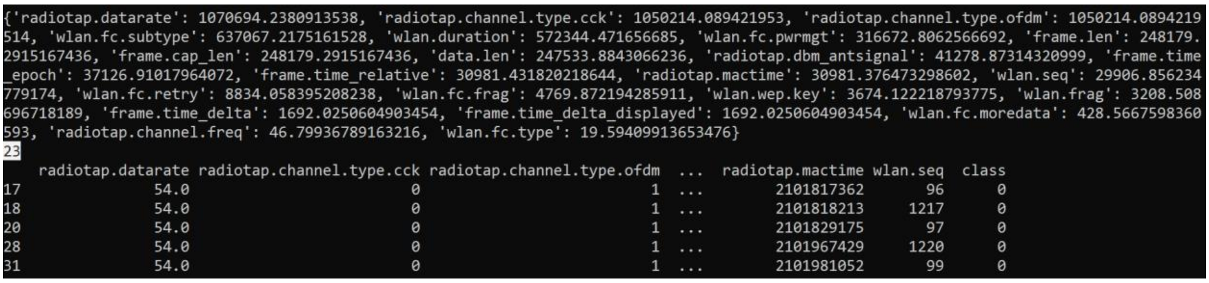
The Anaconda prompt has been launched. Please execute the following instructions:

- python train.py to model is trained.



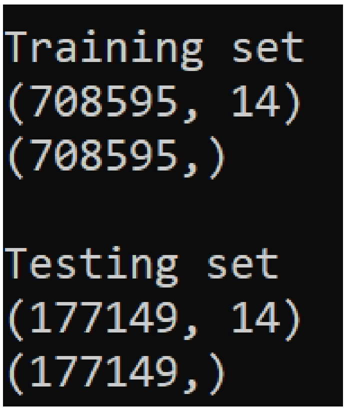
```
(base) C:\ProgramData\Anaconda3\envs\IAP_Fullcode>activate int_detect

(int_detect) C:\ProgramData\Anaconda3\envs\IAP_Fullcode>python train.py
sys:1: DtypeWarning: Columns (37,38,39,40,41,42,43,44,45,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,74,88) have mixed types.
y dtype option on import or set low_memory=False.
normal           775634
arp               64609
cafe_latte        44731
fragmentation       770
Name: class, dtype: int64
    frame.interface_id  frame.offset_shift  frame.time_epoch  ...  wlan.fcs_good  wlan.wep.key  data.len
17                   0                 0.0      1.393661e+09  ...              1             0      1460
18                   0                 0.0      1.393661e+09  ...              1             0        60
20                   0                 0.0      1.393661e+09  ...              1             0        89
28                   0                 0.0      1.393661e+09  ...              1             0        60
31                   0                 0.0      1.393661e+09  ...              1             0        60
```

- Feature Selection show below:



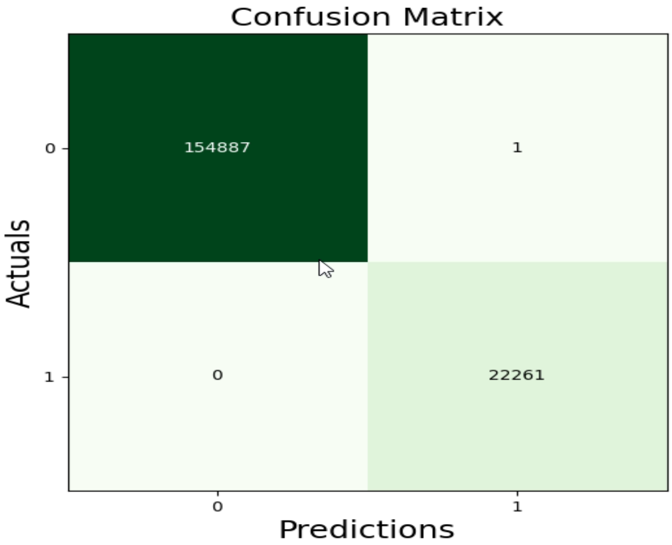
```
{'radiotap.datarate': 1070694.2380913538, 'radiotap.channel.type.cck': 1050214.089421953, 'radiotap.channel.type.ofdm': 1050214.0894219
514, 'wlan.fc.subtype': 637067.2175161528, 'wlan.duration': 572344.471656685, 'wlan.fc.pwrmgt': 316672.8062566692, 'frame.len': 248179.
2915167436, 'frame.cap_len': 248179.2915167436, 'data.len': 247533.8843066236, 'radiotap.dbm_antsignal': 41278.87314320999, 'frame.time
_epoch': 37126.91017964072, 'frame.time_relative': 30981.431820218644, 'radiotap.mactime': 30981.376473298602, 'wlan.seq': 29906.856234
779174, 'wlan.fc.retry': 8834.058395208238, 'wlan.fc.frag': 4769.872194285911, 'wlan.wep.key': 3674.122218793775, 'wlan.frag': 3208.508
696718189, 'frame.time_delta': 1692.0250604903454, 'frame.time_delta_displayed': 1692.0250604903454, 'wlan.fc.moredata': 428.5667598360
593, 'radiotap.channel.freq': 46.79936789163216, 'wlan.fc.type': 19.59409913653476}
23
    radiotap.datarate radiotap.channel.type.cck radiotap.channel.type.ofdm  ...  radiotap.mactime wlan.seq  class
17               54.0                         0                          1  ...        2101817362       96      0
18               54.0                         0                          1  ...        2101818213     1217      0
20               54.0                         0                          1  ...        2101829175       97      0
28               54.0                         0                          1  ...        2101967429     1220      0
31               54.0                         0                          1  ...        2101981052       99      0
```

- When you require to quickly evaluate quality of the model, models that are expensive to train, or very huge datasets, this strategy is ideal. The data will be divided into two tests: training and testing. 20% of the data will be applied to test or evaluate the performance of the machine classification methods, while the remaining 80% will be utilized to train the machine-learning classifiers.

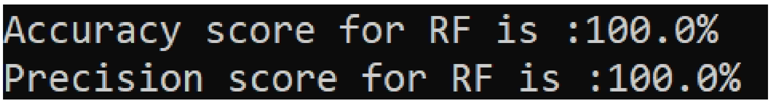


```
Training set
(708595, 14)
(708595,)

Testing set
(177149, 14)
(177149,)
```

- The effectiveness of the machine-learning model would then be evaluated using a confusion matrix. Here, 0 means Normal and 1 means Melicious found. Below confusion matrix for rendom forest. 154887 of these packets were accurately classed as Normal. 22261 of these packets were appropriately identified as Melicious. One packet was wrongly categorised among them.

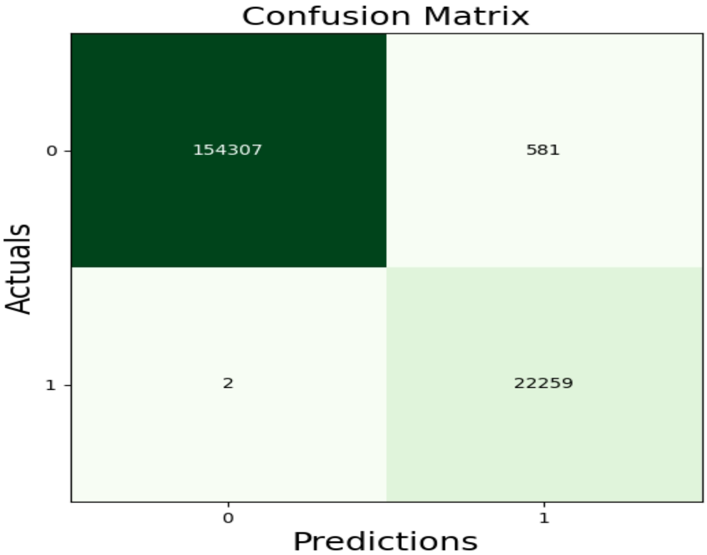


- After that our accuracy is 100% base on above confusion matrix but here limitation is that output base on dataset and systems so if we change the dataset then maybe output will be different.

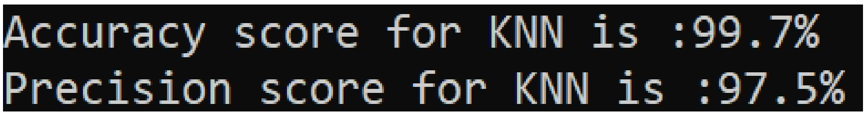


```
Accuracy score for RF is :100.0%
Precision score for RF is :100.0%
```

- Confusion matrix for KNN.

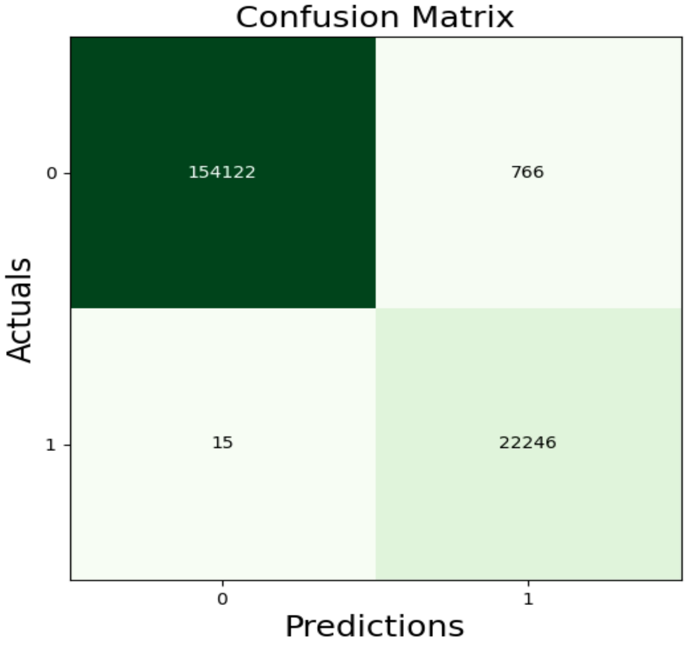


- Accuracy for KNN is 99.7 %.

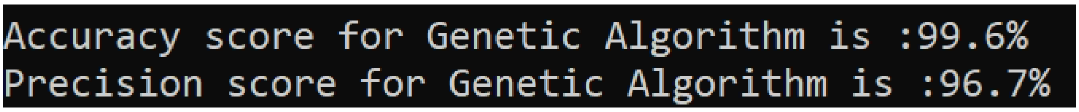


```
Accuracy score for KNN is :99.7%
Precision score for KNN is :97.5%
```

- Confusion matrix for Genetic Algorithm.

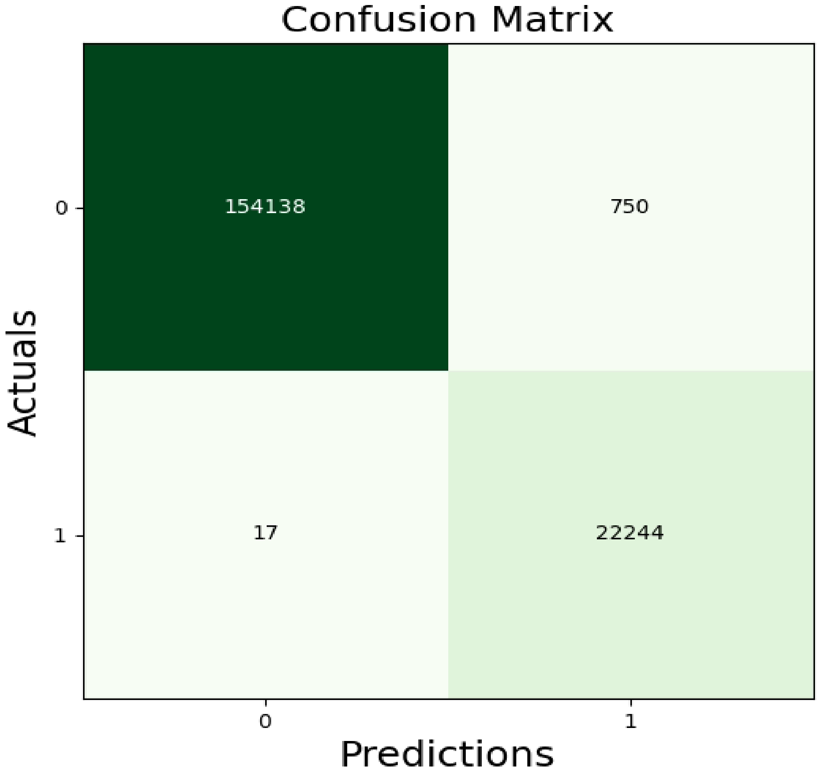


- Accuracy for Genetic Algorithm is 99.6 %.

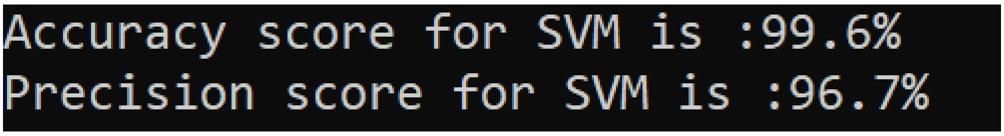


```
Accuracy score for Genetic Algorithm is :99.6%
Precision score for Genetic Algorithm is :96.7%
```

- Confusion matrix for SVM.

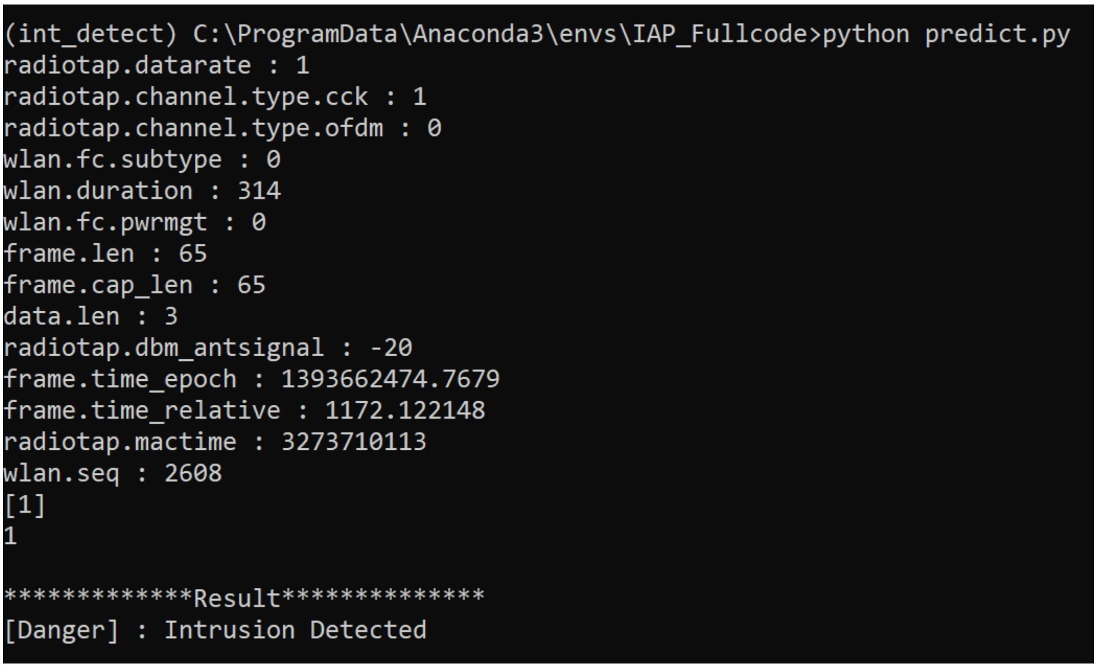


- Accuracy for SVM is 99.6 %.



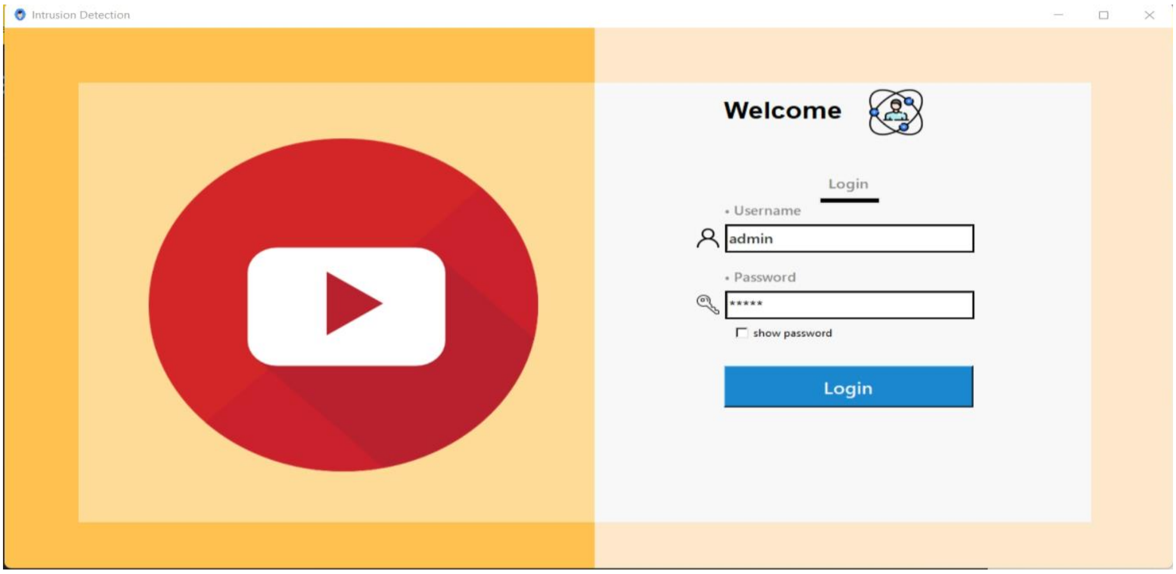
```
Accuracy score for SVM is :99.6%
Precision score for SVM is :96.7%
```

- Command: “python predict.py” to utilizing the training model, forecast if the packet is malicious or not. For the data presented below and in the GUI, there are the attributes of the 14 features.



```
(int_detect) C:\ProgramData\Anaconda3\envs\IAP_Fullcode>python predict.py
radiotap.datarate : 1
radiotap.channel.type.cck : 1
radiotap.channel.type.ofdm : 0
wlan.fc.subtype : 0
wlan.duration : 314
wlan.fc.pwrmgt : 0
frame.len : 65
frame.cap_len : 65
data.len : 3
radiotap.dbm_antsignal : -20
frame.time_epoch : 1393662474.7679
frame.time_relative : 1172.122148
radiotap.mactime : 3273710113
wlan.seq : 2608
[1]
1

************Result*************
[Danger] : Intrusion Detected
```

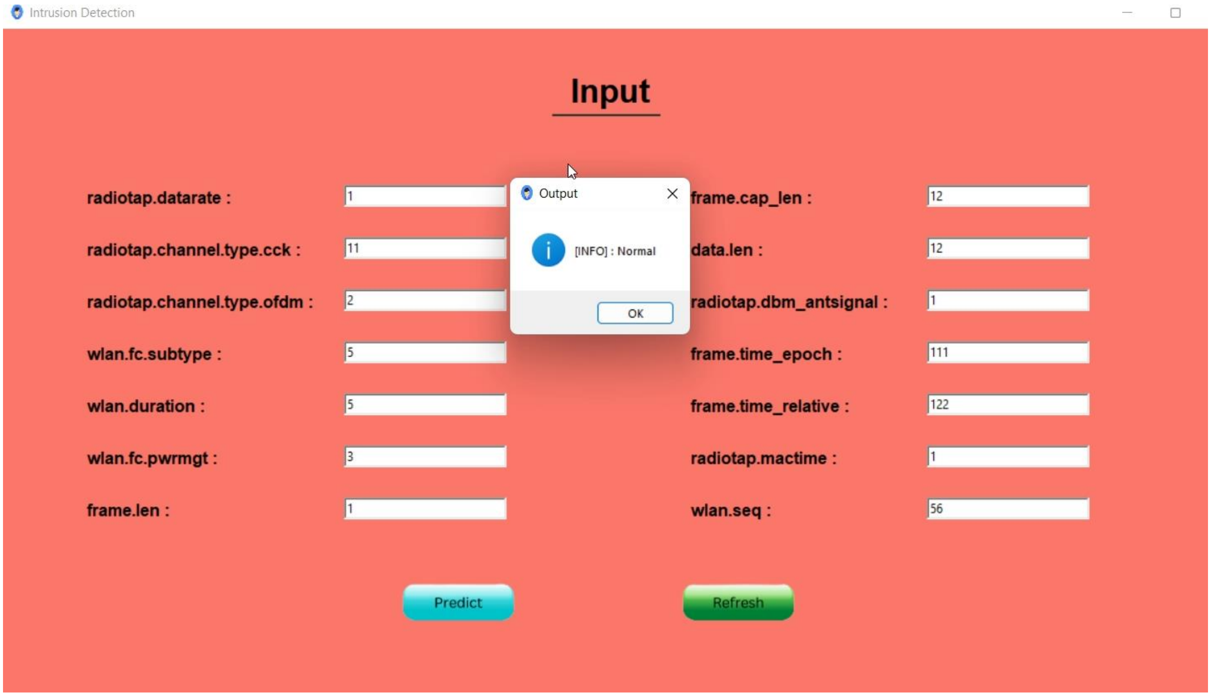
- Command: “python gui.py” to demonstrate the Graphical User Interface (GUI), which accepts user input and presents the outcome.



Username: admin

Password: *****

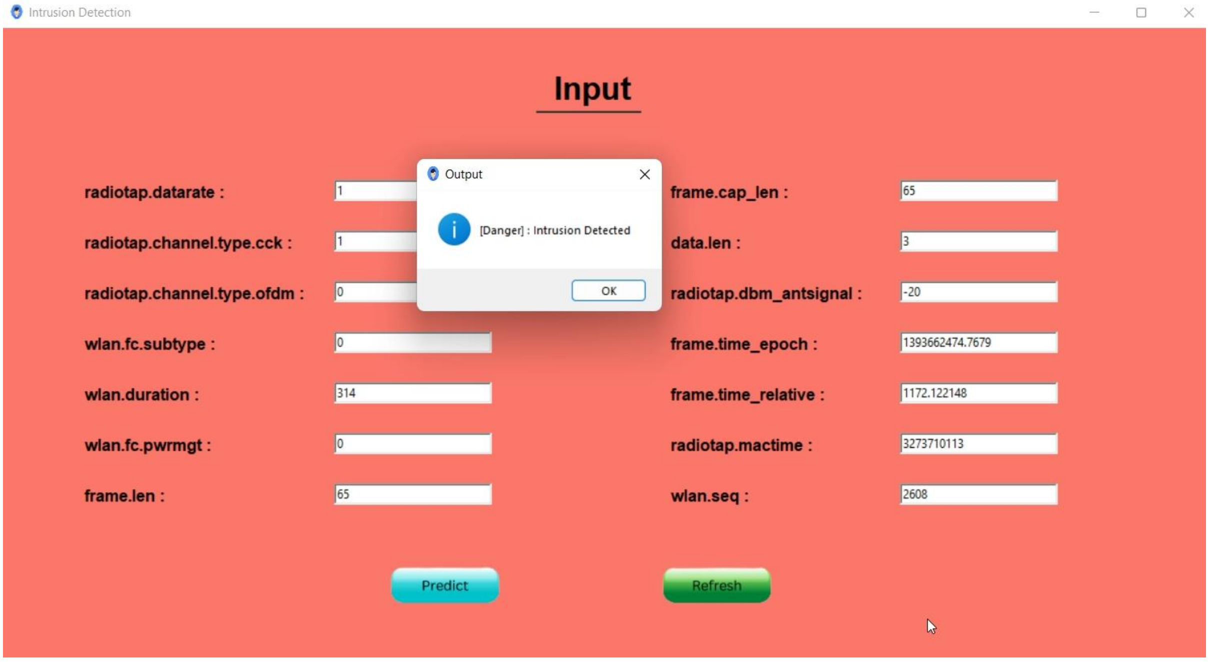
Figure 1: Login page with “admin” username and “admin” passwords.



radiotap.datarate : 1
radiotap.channel.type.cck : 11
radiotap.channel.type.ofdm : 2
wlan.fc.subtype : 5
wlan.duration : 5
wlan.fc.pwrmgt : 3
frame.len : 1
frame.cap_len : 12
data.len : 12
radiotap.dbm_antsignal : 1
frame.time_epoch : 111
frame.time_relative : 122
radiotap.mactime : 1
wlan.seq : 56

[INFO] : Normal

Figure 2: Normal output



radiotap.datarate : 1
radiotap.channel.type.cck : 1
radiotap.channel.type.ofdm : 0
wlan.fc.subtype : 0
wlan.duration : 314
wlan.fc.pwrmgt : 0
frame.len : 65
frame.cap_len : 65
data.len : 3
radiotap.dbm_antsignal : -20
frame.time_epoch : 1393662474.7679
frame.time_relative : 1172.122148
radiotap.mactime : 3273710113
wlan.seq : 2608

[Danger] : Intrusion Detected

Figure 3: Intrusion Detection

7 References

Anaconda | anaconda distribution Anaconda. Available at: https://www.anaconda.com/products/distribution (Accessed: 14 December 2022).

Download - sublime text (no date). Available at: https://www.sublimetext.com/3 (Accessed: 14 December 2022).