National College of Ireland

Configuration Manual

Advancing Android Malware Detection with Machine Learning Techniques

MSc Research Project

Cyber Security

Joseph Mathew

Student ID: 22181741

School of Computing

National College of Ireland

Supervisor: Dr. Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Joseph Mathew
Student ID:	22181741
Programme:	MSc. Cyber Security **Year:** 2024
Module:	Practicum
Lecturer:	Dr. Imran Khan
Submission Due Date:	02/12/2024
Project Title:	Advancing Android Malware Detection with Machine Learning Techniques
Word Count:	614 **Page Count:** 6

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:	Joseph Mathew
Date:	02/12/2024

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

Joseph Mathew
22181741

1 Introduction

In this configuration manual contain the information regards the applications, tools, algorithm and software used for creating a Random Forest algorithm for detecting malware from a dataset. In this model, we use a data set of application behaviour features to classify each sample as "malware" or 'safe'. Feature engineering, hyper parameter tuning, model evaluation and feature importance analysis are combined to make a strong, interpretable malware detection tool. On the 2 parts shows the system specification used for this project. And on 3 part the software tools and library used for creating this model. 4-part Dataset requirements. 5th Code configuration.

2 System Specifications

In this project the setup was done in a notebook PC.

- ASUS FX506LHB 2022
- RAM: 8 GB
- WINDOWS 11 64-bit OS, intel Corei5 processor.
- WINDOWS 11 PRO
- Software setup: Anaconda navigator [1], Jupyter notebook, Python 3.13.0[2].

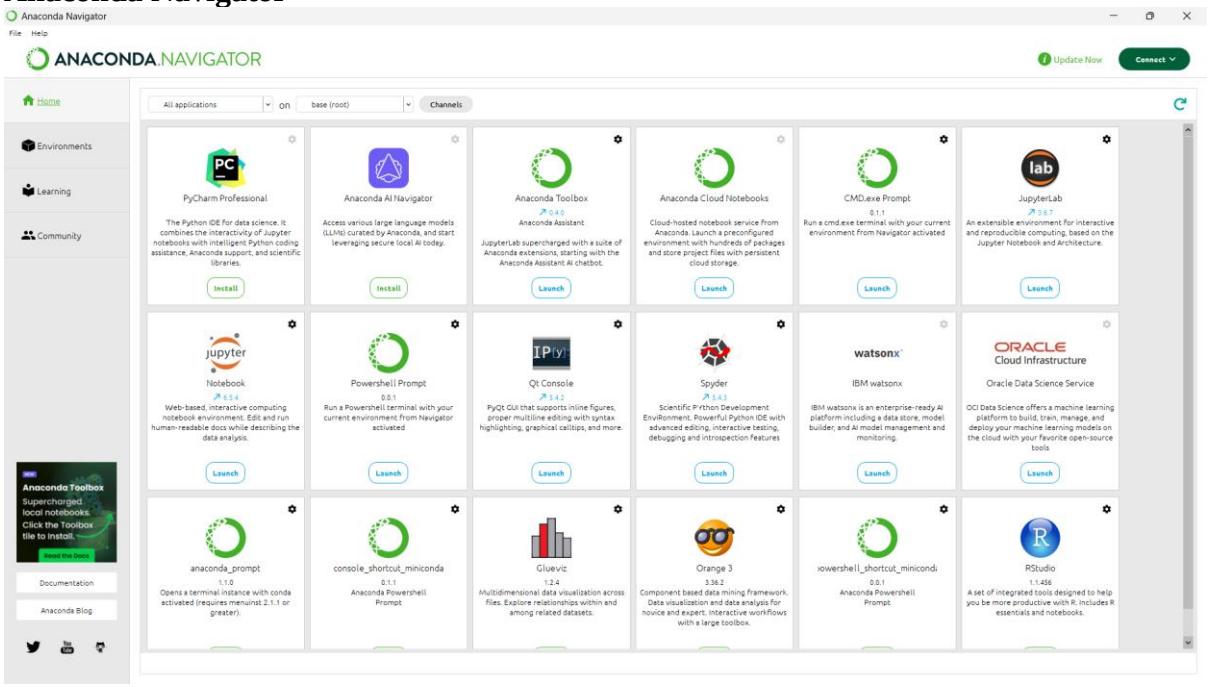
3 Software Specifications

- Jupyter notebook
- Python 3.13.0
- Pandas
- NumPy
- Matplotlib.pyplot
- RandomForestClassifer
- LabelEncoder

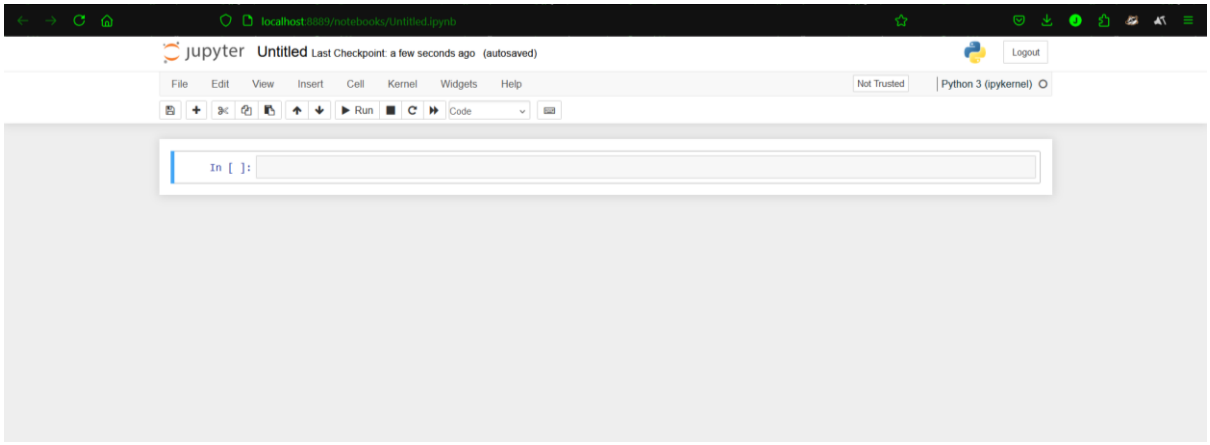
I. Configuration of software and tools

1. Download and Install Anaconda 2.6.0
2. Launch Jupyter notebook 6.5.4 and create a new python3 (ipykernal)

Anaconda Navigator



Jupyter Notebook



4 Dataset

Dataset: Make sure it's a CSV file with the malware labels and the features for the model.
Path: Replace the file path in the code with the local or server file's CSV file path:

```
In [10]: file_path = '5561.csv'
```

The dataset should contain a column called class which determine the data is malware or safe.

5 Software Specifications

- Import Libraries

```
In [3]: import pandas as pd
In [4]: from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
In [5]: from sklearn.preprocessing import StandardScaler, LabelEncoder
In [6]: from sklearn.ensemble import RandomForestClassifier
In [7]: from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score
In [8]: import matplotlib.pyplot as plt
In [9]: import numpy as np
```

- Find path: give path to the dataset

```
In [10]: file_path = '5561.csv'
```

- Data processing: The code automatically encodes the target variable (class) into binary form (malware vs. safe).

- Model configuration: The Random Forest model is initialized with a random state for reproducibility.

```
: rf_model = RandomForestClassifier(random_state=42)
```

- Model training and evaluation: train the model using the parameters and the model also provides other evaluation results such as ROC AUC, confusion matrix and feature importance levels. This gives an overall of the model in detecting malware.

```
In [21]: param_grid = {
             'n_estimators': [100, 200, 300],
             'max_depth': [None, 10, 20, 30],
             'min_samples_split': [2, 5, 10],
             'min_samples_leaf': [1, 2, 4]
         }
```

```
grid_search = GridSearchCV(estimator=rf_model, param_grid=param_grid, cv=5, n_jobs=-1, scoring='roc_auc')
grid_search.fit(X_train, y_train)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(random_state=42), n_jobs=-1,
             param_grid={'max_depth': [None, 10, 20, 30],
                         'min_samples_leaf': [1, 2, 4],
                         'min_samples_split': [2, 5, 10],
                         'n_estimators': [100, 200, 300]},
             scoring='roc_auc')
```

```
estimator: RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
RandomForestClassifier
RandomForestClassifier(random_state=42)
```

```
best_params = grid_search.best_params_
```

```
best_score = grid_search.best_score_
```

```
print("Best Parameters:", best_params)
print("Best Cross-Validation ROC AUC Score:", best_score)
```

```
Best Parameters: {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Best Cross-Validation ROC AUC Score: 0.9979906499707261
```

References

1) **Anaconda Navigator** https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Windows-x86_64.exe
2) **Python 3.13.0** https://www.python.org/ftp/python/3.13.0/python-3.13.0-amd64.exe