Configuration Manual

MSc Research Project

LEVERAGING DATA ANALYTICS TO ENHANCE CYBER SECURITY THREAT DETECTION

Anantha Padmanabha Rajendran

Student ID: x22242520

School of Computing

National College of Ireland

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	R Ananatha Padmanabha Naidu
Student ID:	22242520
Programme:	MSc Data Analytics **Year:** 2024-25
Module:	Research Project
Lecturer:	Dr. Anu Sahni
Submission Due Date:	24/04/2025
Project Title:	LEVERAGING DATA ANALYTICS TO ENHANCE CYBERSECURITY THREAT DETECTION
Word Count:	**903** **Page Count: 9**

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: .....

Date: .....

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.

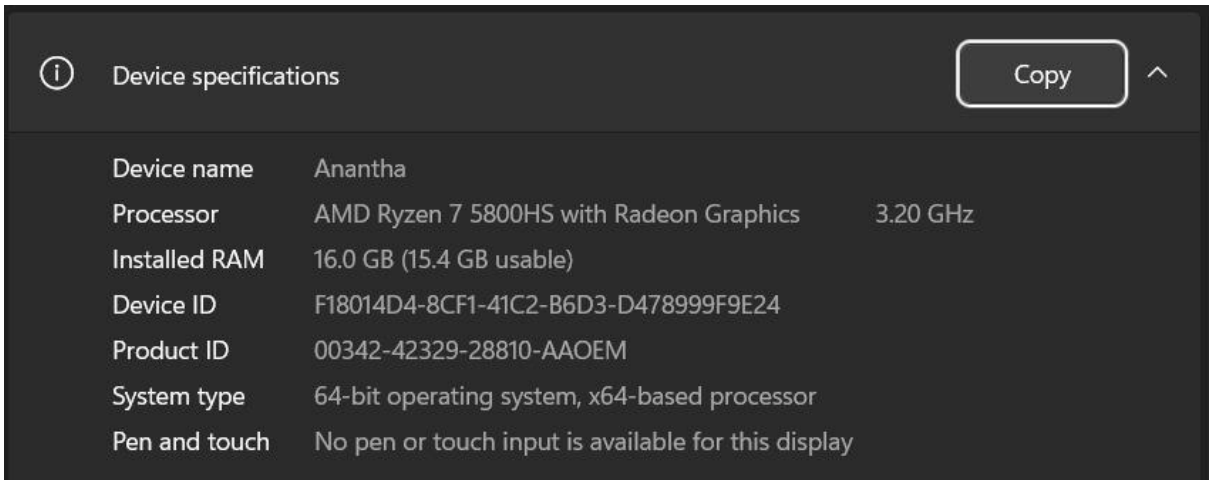
Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

1. Introduction

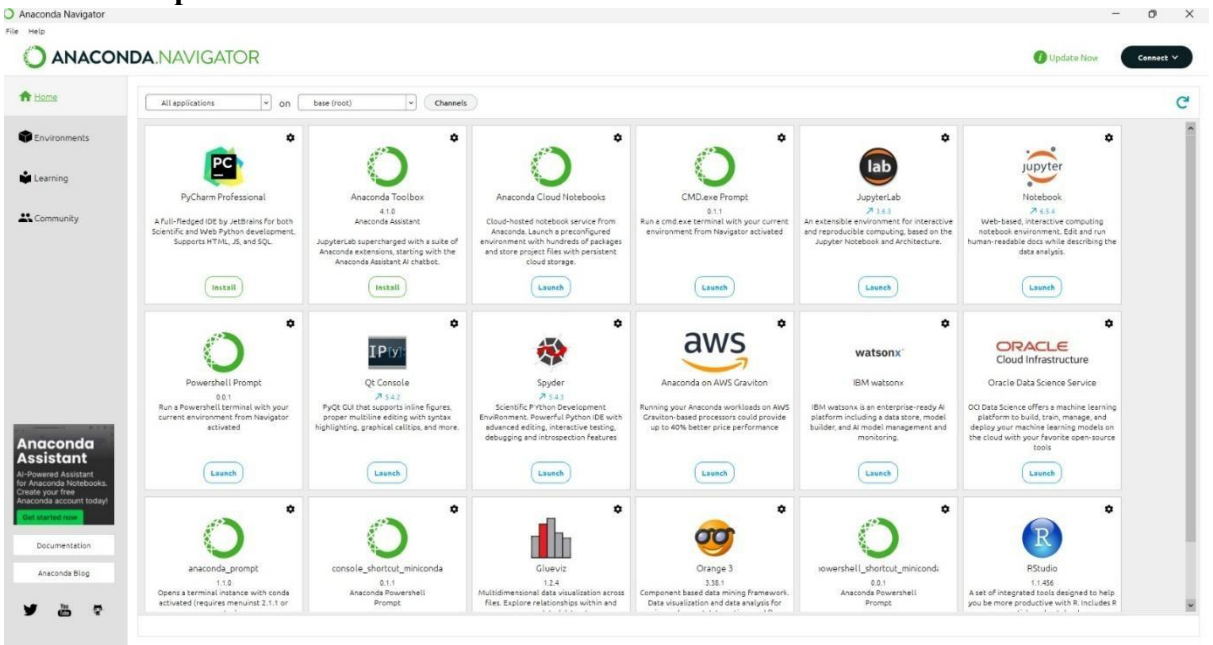
This configuration manual provides complete setup guidelines to successfully run the thesis project. The implementation involves machine learning (ML) and deep learning (DL) models to enhance cybersecurity threat detection using Python-based data analytics techniques. It includes hardware requirements, software environments, library setup, and additional tools for diagram generation.

2. Device Hardware Specification

Component	Specification
Device Name	Anantha
Processor	AMD Ryzen 7 5800HS with Radeon Graphics @ 3.20 GHz
RAM	16.0 GB (15.4 GB usable)
System Type	64-bit Operating System, x64-based processor
Graphics	Integrated Radeon Graphics
Storage	Recommended ≥ 512 GB SSD (assumed, for performance)



3. Software Specification



Software	Details
OS	Windows 10 / 11 (x64)
Python	Version 3.8 or later
IDE	Jupyter Notebook (via Anaconda) or Google Colab
ML/DL Frameworks	TensorFlow 2.x (with Keras), scikit-learn
Jupyter Version	7.0.8 (if using Anaconda distribution)

4. Report Diagram Generation

All project architecture diagrams, workflow charts, and model pipeline visuals were generated using(Miro, 2024):

Miro: https://miro.com

5. Official Library Setup

Below is a list of required Python libraries along with their usage:

Library Installation Command	Purpose
pip install pandas	Data manipulation and cleaning
pip install numpy	Numerical operations
pip install scikit-learn	Model building (Decision Tree, K-Means)
pip install tensorflow	Deep learning framework (Neural Network)
pip install keras	High-level deep learning API
pip install seaborn	Statistical data visualization
pip install matplotlib	Graph plotting and heatmaps
pip install imbalanced-learn	Handling class imbalance (e.g., SMOTE)
pip install uuid	Creating unique identifiers for records

Import Required Libraries

In [12]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt

from sklearn.cluster import KMeans
from tensorflow.keras.models import Sequential
from sklearn.tree import DecisionTreeClassifier
from tensorflow.keras.layers import Dense, Dropout
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, silhouette_score
```

4. Dataset Used

- **Name:** thesis dataset1.csv and thesis dataset.csv
- **Source:** Kaggle
- **Description:** Contains labeled network traffic data for training and testing various cybersecurity threat detection models.(Kaggle, 2024).

6. Dataset Preprocessing

```
In [2]: df1 = pd.read_csv('thesis dataset1.csv')
        print("Dataset Head:")
        display(df1.head())

Dataset Head:
```

Fig 6.1: Load Dataset

```
Dataset Preview:
   payload_byte_1  payload_byte_2  payload_byte_3  payload_byte_4  \
0              66              80              83              33
1              66              80              83              33
2              66              80              83              33
3              66              80              83              33
4              66              80              83              33

   payload_byte_5  payload_byte_6  payload_byte_7  payload_byte_8  \
0              32              55              98              32
1              32              53              52              32
2              32              57              32              66
3              32              56              51              32
4              32              57              32              66

   payload_byte_9  payload_byte_10  payload_byte_11  payload_byte_12  \
0              66               80               83               33
1              66               80               83               33
2              80               83               33               32
3              66               80               83               33
4              80               83               33               32

   payload_byte_13  payload_byte_14  payload_byte_15  total_len protocol  \
0               32               55               98        100   others
1               32               53               52        100   others
2               57               32               66         90   others
3               32               56               51        100     pipe
4               57               32               66         90   others
```

	Event ID	Timestamp	Source IP	Destination IP	User Agent	Attack Type	Attack Severity	Data Exfiltrated	Threat Intelligence	Response Action
0	2019969e-ecfa-41c4-b681-9b684bc3b3bf	07-02-2020 23:46	219.80.193.15	44.155.75.24	Mozilla/5.0 (Macintosh; PPC Mac OS X 10_7_8 rv...	Ransomware	Critical	False	Crime low this behind option tax product.	Eradicated
1	1668e954-781f-4731-94dc-24218b983ba1	25-05-2021 19:03	110.155.68.245	178.123.150.38	Mozilla/5.0 (Windows 95) AppleWebKit/534.2 (KH...	Malware	Critical	True	Responsibility work way effect.	Eradicated
2	0ef24a20-1d25-41fa-81b8-e19fb63e9e4c	04-01-2022 09:08	171.153.115.83	76.187.142.133	Mozilla/5.0 (X11; Linux x86_64; rv:1.9.7.20) G...	Ransomware	High	False	Artist though type imagine food push.	Eradicated
3	073b6225-0998-488c-aa1c-23e49814b6ff	12-10-2022 19:48	29.49.228.195	89.39.7.177	Mozilla/5.0 (Linux; Android 7.1.1) AppleWebKit...	DDoS	Critical	False	In still military despite TV look.	Contained
4	783fd153-6b88-44c1-8db5-d882300088cc	24-11-2021 02:04	120.43.64.52	113.82.34.164	Mozilla/5.0 (iPad; CPU iPad OS 9_3_6 like Mac ...	Malware	Medium	False	Push always least police it range either.	Eradicated

```
In [3]: df1.dropna(inplace=True)
        df1.drop_duplicates(inplace=True)

In [4]: df1.drop(columns=["Event ID", "User Agent"], inplace=True, errors='ignore')
```

Fig 6.1: Data Cleaning

```
In [5]: encoding_columns = ["Response Action", "Attack Type", "Attack Severity", "Data Exfiltrated"]
        for col in encoding_columns:
            df1[col] = df1[col].astype('category').cat.codes + 1
```

Fig 6.2: Convert Categorical to Numeric

7. Exploratory Data Analysis (EDA)

EDA was used to understand class imbalances and data distributions.

```
In [7]: plt.figure(figsize=(12, 6))
        sns.histplot(df1["Attack Severity"], bins=10, kde=True)
        plt.title("Distribution of Attack Severity")
        plt.xlabel("Severity Level")
        plt.ylabel("Count")
        plt.show()
```

Fig 7.1: Attack Severity Distribution

```
In [8]: plt.figure(figsize=(12, 6))
        sns.countplot(x=df1["Response Action"], palette="coolwarm")
        plt.title("Count of Different Response Actions")
        plt.xlabel("Response Action")
        plt.ylabel("Frequency")
        plt.show()
```

Fig 7.2: Response Action Count

8. Model Development & Training

8.1 Preprocessing for Model Input

```
In [22]: y = df['label']
         X = df.drop(columns=['label'])
         scaler = StandardScaler()
         X_scaled = scaler.fit_transform(X)
         X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)
```

Fig 7.1: Preprocessing

8.2 Supervised Learning – Decision Tree

The Decision Tree Classifier was implemented in Scikit-learn. It is used to classify known cyber attack types. Due to its interpretability and low training time, it served as a baseline model.

```
In [23]: clf = DecisionTreeClassifier()
         clf.fit(X_train, y_train)
         y_pred = clf.predict(X_test)
         print("Decision Tree Classification Report:")
         print(classification_report(y_test, y_pred))
```

Fig 8.1: Decision Tree Classifier

8.3 Unsupervised Learning – K-Means Clustering

K-Means was used for anomaly detection and unsupervised clustering of similar flows. Whenever clusters are formed, the cluster quality is measured according to the Silhouette Score.

```
In [24]: kmeans = KMeans(n_clusters=3, random_state=42)
         kmeans.fit(X_scaled)
         kmeans_labels = kmeans.labels_
         print("Silhouette Score:", silhouette_score(X_scaled, kmeans_labels))
```

Fig 8.1: K-Means Clustering

8.4 Deep Learning – Neural Network

With the TensorFlow/Keras framework, we constructed a feedforward neural network to classify cyber threats exhibiting complex, nonlinear patterns. Dropout layers were included in the construction to combat overfitting (TensorFlow Developers, 2024; Keras, 2024).

```
model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dropout(0.3),
    Dense(len(y.unique()), activation='softmax')
])

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test, y_test))
```

Fig 8.3: Neural Network Model Implementation (Keras)

Result: Achieved validation accuracy of 74.4%, outperforming other models in generalization.

References

Miro, 2024. *Miro Online Collaborative Whiteboard Platform*. [online] Available at: https://miro.com [Accessed 7 Apr. 2025].

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, pp.2825–2830. Available at: https://scikitlearn.org [Accessed 24 Apr. 2025].

TensorFlow Developers, 2024. *TensorFlow*. [online] Available at: https://www.tensorflow.org [Accessed 24 Apr. 2025].

Keras, 2024. *Keras: Deep Learning for Humans*. [online] Available at: https://keras.io [Accessed 7 Apr. 2025].

Hunter, J.D., 2007. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), pp.90–95. Available at: https://matplotlib.org [Accessed 8 Apr. 2025].

Waskom, M., 2023. *Seaborn: Statistical Data Visualization*. [online] Available at: https://seaborn.pydata.org [Accessed 8 Apr. 2025].

Lemaître, G., Nogueira, F. and Aridas, C.K., 2017. *Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning*. Journal of Machine Learning Research, 18(1), pp.559–563. Available at: https://imbalanced-learn.org [Accessed 8 Apr. 2025].

Kaggle, 2024. *Network Traffic Dataset for Cybersecurity Research*. [online] Available at: https://www.kaggle.com [Accessed 6 Apr. 2025].