National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Abhiram Adluru
Student ID: x22187898

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Abhiram Adluru
Student ID:	x22187898
Programme:	Cloud Computing
Year:	2023
Module:	Msc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	12/12/2024
Project Title:	Configuration Manual
Word Count:	703
Page Count:	4

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:	Abhiram Adluru
Date:	11th December 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

Abhiram Adluru
x2218798

1 Introduction

The Configuration Manual provides all the details for integrating APT Application,the system is deployed using python for model training and evaluation and deployed on a AWS cloud environment for real-time network Analysis

2 System Specifications

2.1 HardWare Requirements

2.1.1 HardWare Configuration for the web application

- Processor: Intel 11th Gen Core i3 @ 2.4 GHz
- 16 GB DDR4 RAM 3200MHz
- Storage: 512 GB SSD
- Operating System: Windows 10, 64-bit

2.1.2 Cloud Based Machine learning Model Development

- processor : Hosted Multi-Core CPU on Google Cloud
- RAM : 12 GB
- Environment : Pre-Configured Jupyter Notebook Interface with Python Libraries pre-installed.

2.2 Software Requirements

The Whole Project was developed and implemented using python for model development and back-end processing,the apt detection system has conv-LSTM model along with random forest and svm to classofy various stages of network activities,the web application is develpoed using flask and is integrated with machine learning Back end witj interface usinh HTML,CSS.

2.2.1 Frame Work and Tools

- Flask : Back End Frame work for serving with the model predictions.
- Jupyter Notebook : for develpoing and testing the machine learning model VS Code IDE : for Writing and managing and running the code locally

2.2.2 Python Dependencies

The Following Python Libraries were used for model development and to ensure efficient handling of data and visual exploration.

- Pandas : for handling and preprocessing datasets
- matplotlib : for static visualization of the data.
- ploty.express : for interactive visualizations such as bar and pie charts.
- sklearn.preprocessing: for scaling nunerical data to specified range

3 Dataset Description

The dataset used is the DAPT 2020 Dataset, taken from Kaggle, which includes network traffic logs designed for detecting APTs. The features describe packet lengths, flow attributes, and flag counts among others, while categorical labels include Stage, such as Benign and Reconnaissance, and Activity to identify concrete behaviors. The dataset was preprocessed by handling missing values, encoding categorical columns, and normalizing numerical features to prepare it for machine learning. It was split into training and testing sets, with a split of 70-30, respectively. The training data was used in developing models such as Conv-LSTM, Random Forest, and SVM, while the test data were used for evaluation. This dataset allows the system to imitate real-life APT scenarios, supporting model training, testing, and real-time applicationsMyneni (2020).

- `https://www.kaggle.com/datasets/sowmyamyneni/dapt2020?resource=download`.

4 Environmental setup

4.1 Google Colab Setup

for training and testing the APT detection model upload the APT-attacks-final.ipynb code file and other datasets required to run the code

1. run the following code to connect to google drive
   - from google.colab import drive
   - drive.mount('/content/drive')
2. upload all the required datasets and .pynb code file provided in the artifacts to a folder in google drive and give necessary permissions to access and change the location of the path in your code using below format

- `file_path = '/content/drive/MyDrive/apt_project/your_file.csv'`

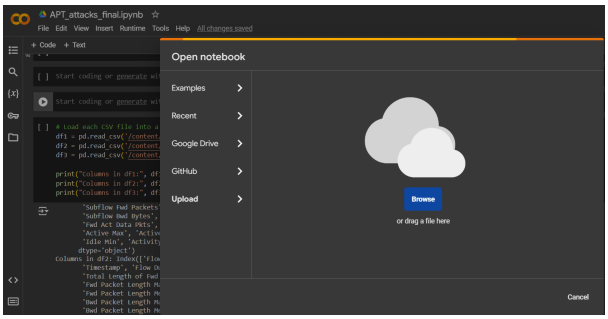

Figure 1: Google Colab Setup

3. Execute the code in google colab to load the data and succesfully running the code

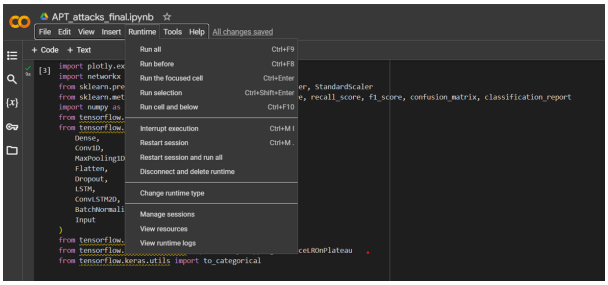

Figure 2: Execute the Code

4.2 Application Setup

1. ensure python 3.8 or higher version is installed on your machine and create a virtual environment for running the flask application.

```
python –m venv venv
.\venv\Scripts\activate
```

2. once the virtual environment is activated install the required flask version and dependencies from requirements.txt for running the application

```
pip install Flask==3.1.0
```

```
pip install –r requirements.txt
```

3. once the flask is installed navigate to project directory where app.py is located and run the following command

```
python app.py
```

4. open another terminal and navigate to directory where client.py is located to initiate the communication between client and server

```
python client.py
```

Figure 3: Interface of the application

5 Application deployment on AWS elastic beanstalk

1. sign into AWS management Console navigate to Elastic Beanstalk and create a new environment
2. upload and deploy provided .zip file which contains all the deployment files required for the deployment of the application.

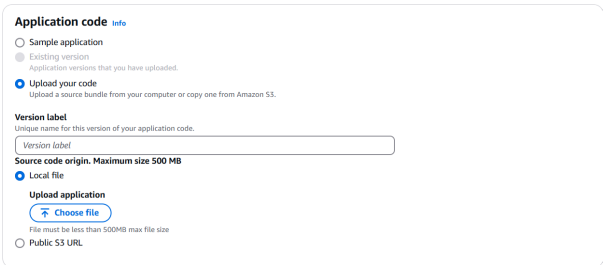

Figure 4: Upload the zip code

3. after deployment,test the application by accessing the provided url in the elastic beanstalk

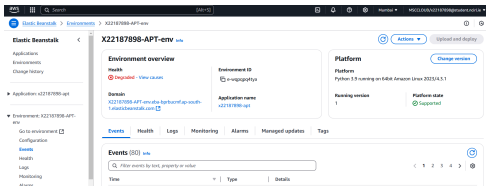

Figure 5: successful deployment of the application on elasticbeanstalk

References

Myneni, S. (2020). Kaggle dataset: Dapt 2020. Accessed: 2024-12-07.
URL: *https://www.kaggle.com/datasets/sowmyamyneni/dapt2020/code*