National College of Ireland

Configuration Manual

MSc Research Project
MSc in Cybersecurity

Srinivas Dammu
Student ID: X23228733

School of Computing
National College of Ireland

Supervisor: Raza UI Mustafa

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Srinivas Dammu

Student ID: X23228733

Programme: MSc in Cybersecurity **Year:** 2024-25

Module: MSc research project

Lecturer: Raza UI Mustafa

Submission Due Date: 12/12/2024

Project Title: Enhancing Cloud Security in the Financial Sector: An AI-Driven Threat Detection and Response Framework

Word Count: 972 **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: .....

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):	

Configuration Manual

Srinivas Dammu
Student ID: X23228733

System Overview

Objective: Propose the use of two-tier Intrusion Detection System (IDS), based on AI models, for protection of financial cloud infrastructures.

Models Used:

- K-Nearest Neighbours (KNN) technique for classification of the dataset as benign traffic or the malicious traffic.
- Proposed Multiclass Deep Neural Network (DNN) for seven specific types of attack classification.

Dataset: CICIDS2018 is chosen because it includes many aspects of scenarios, which can well represent the network traffic distribution.

Data Configuration

Data Preprocessing:
we select features using correlation matrices
For normalization we choose Min – Max scaling
Class balancing ensures that both benign and malicious traffic are represented evenly.

Model Training

- **KNN Model:** It is used for binary classification and achieved accuracy of 92.79% for training configurations selected 11 neighbours
- **DNN Model:** It is used for multiclass classification, and it has an accuracy of 94.19%. It has a multilayer hidden layer structure and features ReLU activation functions – Adam optimizer, categorical cross-entropy loss to address the classifier challenge of handling the extensive range of attack types present in the dataset.

System Design

The System architecture is intended to provide robust intrusion detection and response methods that interface smoothly with financial transaction procedures. The layered detection technique serves as security barrier between users and payment gateway, monitoring for threats in real time. When possible, risks are recognized, and automatic reaction procedures are activated to prevent malicious transactions and its immediate warn users and administrators.
Front-End Features

The front-end, or GUI presented to the user, seems simple and is intended to display the current and instant threat level. It consists of one more secure way of login to the user account, the great visualization of transactions and the threats found. The interface enables users to be informed when undergoing transactions by providing them with probabilities of threats and types of attacks if any making users develops trust and engage more on the application**.**

Backend Implementation

The backend is coded in Python with Flask with stable features for model integration and API creation. Jupyter Notebook is employed for experiment, this is due to the flexibility of data analysis and continuous enhancement of the created models. The inference system includes binary (KNN) and multiclass (DNN) classification models to distinguish traffic as normal and malign and determine the attacks. The findings of all threats detected, and the classification results are documented with extreme detail for reference in the improvement of the current system and future threats analysis.

Installation and Connection Steps

Hardware:

- Processor: AMD Ryzen 7 7735U with Radeon Graphics 2.70 GHz
- RAM: 16 GB
- Storage: 1TB

Software:

- Operating **System**: Windows 11
- Python **Version**: Python 3.13.0
- Libraries: TensorFlow, Scikit-learn, Pandas, Matplotlib, Flask

1 Step 1: Install Python

1. Download Python 3.13.0 from the official Python website.
2. During installation:
   - Ensure that you check the option to add Python to your system PATH.
   - Verify the installation by running python --version in the terminal.
3. Download the latest version of Anaconda from the official website.

4. Activate the Environment

```
(base) C:\Users\srini>conda activate myproject

(myproject) C:\Users\srini>cd C:\Users\srini\OneDrive\Desktop\FrontEnd\Enhancing cloud security using AI driven threat detection and response\client
```

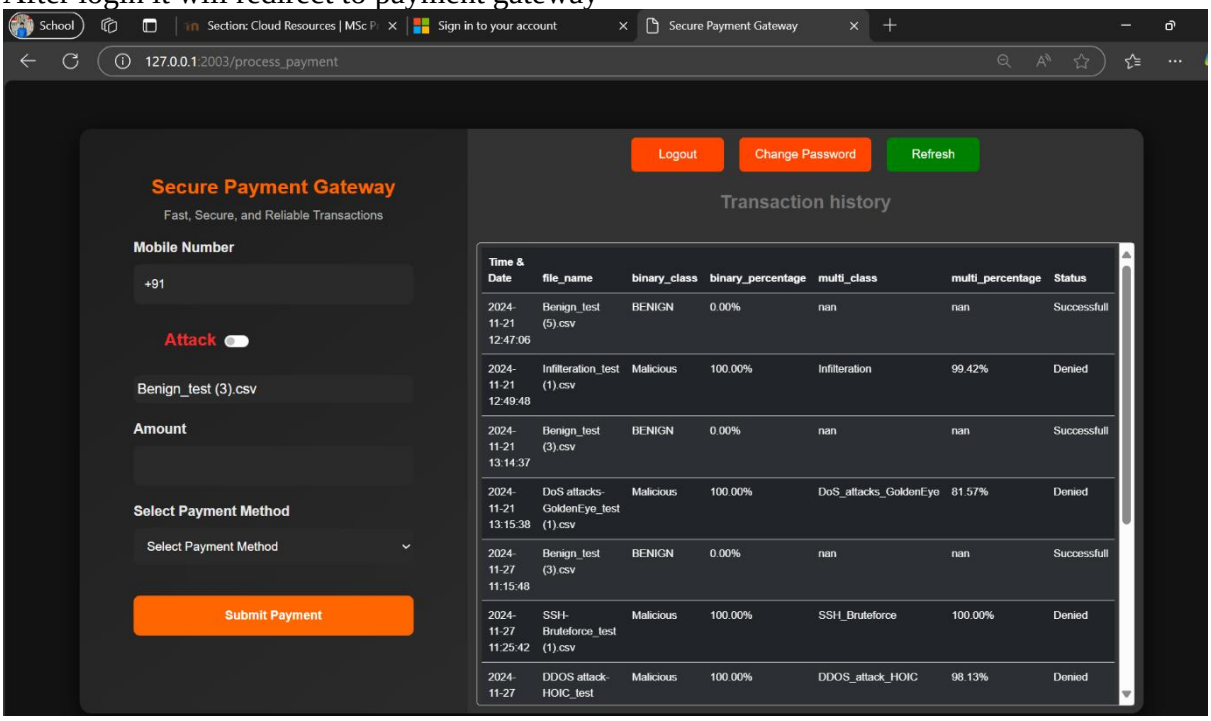
5. Install the required Python libraries for the project:
   Cmd: pip install tensorflow scikit-learn pandas matplotlib flask
6. Download CICIDS2018 dataset from URl: https://www.unb.ca/cic/datasets/ids-2018.html
7. Run preprocessing scripts to prepare dataset
   cmd : python preprocess_data .py

8. Train the KNN model & DNN model for binary classification and multiclass classification
   Cmd : python train_binary_model.py
   Cmd : train_multiclass_model.py

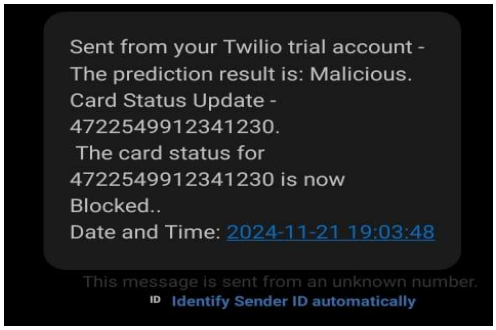
9. The ytrained models (binary_model.pk1 and multiclass_model.pk1) are saved in models directory
10. Setup the backend file and run
    Cmd : python app.py

```
(myproject) C:\Users\srini\OneDrive\Desktop\FrontEnd\Enhancing cloud security using AI driven threat detection and response\client>p
thon app.py
2024-12-11 13:41:07.675616: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64
110.dll'; dlerror: cudart64_110.dll not found
2024-12-11 13:41:07.675992: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GP
 set up on your machine.
2024-12-11 13:41:14.075373: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.d
l'; dlerror: nvcuda.dll not found
2024-12-11 13:41:14.075480: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2024-12-11 13:41:14.081208: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for ho
t: SrinivasDammu
2024-12-11 13:41:14.081444: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: SrinivasDammu
2024-12-11 13:41:14.084128: I tensorflow/core/platform/cpu_feature_guard.cc:151] This TensorFlow binary is optimized with oneAPI Dee
 Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations:  AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:2003
Press CTRL+C to quit
 * Restarting with stat
2024-12-11 13:41:15.324297: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64
110.dll'; dlerror: cudart64_110.dll not found
2024-12-11 13:41:15.324507: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GP
 set up on your machine.
2024-12-11 13:41:18.961169: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'nvcuda.d
l'; dlerror: nvcuda.dll not found
2024-12-11 13:41:18.961336: W tensorflow/stream_executor/cuda/cuda_driver.cc:269] failed call to cuInit: UNKNOWN ERROR (303)
2024-12-11 13:41:18.965682: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for ho
t: SrinivasDammu
```

11. Open a web browser and paste : httpp://127.0.0.1:2003
12. It will display a admin credentials login page
13. After login it will redirect to payment gateway



14. On the attack and it will denied the transaction
15. It view live updates on detected threats, transaction history and logs.
16. Cloud Storage Service: AWS S3.
17. API Key: Enter your cloud service API key.
18. Access Credentials: Add necessary credentials to enable seamless data integration
19. I use twillio for SMS alerts . If a transaction is denied by a attack it will directly send an alert to the user



Performance Metrics

- The KNN model demonstrates strong precision and recall for both benign and malicious traffic classification.
- While highly accurate, the model has highlighted areas requiring improvement, particularly in reducing false negatives, which are critical for ensuring no threats are missed.
- Analyzing the results of the DNN model assessment of the attacks, it can be noted that the DNN model has successfully identified all types of attacks even the most complex ones including DoS, DDoS and SSH Brute Force attacks.
- There are some discrepancies that are infrequent and these result from boundary conditions where certain class imbalances or less training samples for certain attack types could have caused the problem. Such problems are solved with the help of methods, such as oversampling and adjusting architecture of the model.

References

Chang, V., Golightly, L., Modesti, P., Xu, Q.A., Doan, L.M.T., Hall, K., Boddu, S. and Kobusińska, A., 2022. *A survey on intrusion detection systems for fog and cloud computing*. *Future Internet*, 14(3), p.89.

Dietz, K., Mühlhauser, M., Kögel, J., Schwinger, S., Sichermann, M., Seufert, M., Herrmann, D. and Hoßfeld, T., 2024. *The Missing Link in Network Intrusion Detection: Taking AI/ML Research Efforts to Users*. *IEEE Access*.

Ghanem, W.A.H., Jantan, A., Ghaleb, S.A.A. and Nasser, A.B., 2020. *An efficient intrusion detection model based on hybridization of artificial bee colony and dragonfly algorithms for training multilayer perceptrons*. *IEEE Access*, 8, pp.130452-130475.

Islam, U., Muhammad, A., Mansoor, R., Hossain, M.S., Ahmad, I., Eldin, E.T., Khan, J.A., Rehman, A.U. and Shafiq, M., 2022. *Detection of distributed denial of service (DDoS) attacks in IoT-based monitoring system of banking sector using machine learning models*. *Sustainability*, 14(14), p.8374.

Milosevic, M.S. and Ciric, V.M., 2022. *Extreme minority class detection in imbalanced data for network intrusion*. *Computers & Security*, 123, p.102940.