AI-Driven Dynamic Fuzz Testing for IoT Security

Panel No. 06
Supervisor Name
Dr. Balaji Srikaanth P, AP/NWC
Dr. S. Nagendra Prabhu, AP/CINTEL

Batch No. NW000156
Shaurya Singh Srinet – RA2111032010006
Shounak Chandra – RA2111032010026
Charvi Jain – RA2111047010113

Functional Document for User Story 3: Train GNN Model for DDoS Detection

1. Introduction

This document outlines the functionality required for training a Graph Neural Network (GNN) model for detecting Distributed Denial-of-Service (DDoS) attacks in IoT network traffic. This is part of a larger AI-driven IoT security project focusing on dynamic fuzz testing and mitigation of cyberattacks.

2. Product Goal

The primary goal is to accurately detect and classify DDoS traffic using a GNN model trained on pre-processed datasets generated from IoT network simulations. The model will differentiate between benign and malicious traffic, aiding cybersecurity efforts in real-time threat detection.

3. Demography (Users and Locations)

- Target Users Data scientists, cybersecurity analysts, and AI researchers.
- User Characteristics Proficient in network security, AI modelling, and working knowledge of GNNs.
- Location Intended for global use by professionals and researchers involved in cybersecurity.

4. Business Processes

- Model Architecture Design
  - Define the architecture for the GNN model, considering input features like network traffic flow and topology.
  - Implement layers tailored for anomaly detection.

- Model Training
  - Use the pre-processed dataset to train the GNN model to recognize DDoS traffic.
  - Split the dataset into training, validation, and test sets.

- Model Testing and Saving
    - Evaluate the model on a separate validation dataset to assess its detection accuracy.
    - Store the trained model for deployment and further testing.

- Features
    - Model Training and Evaluation
    - Training process using supervised learning on labelled IoT network traffic data.
    - Validation to ensure the model achieves at least 75% accuracy in detecting DDoS attacks.
    - Utilize cross-validation to ensure robustness.

- Hyperparameter Optimization
    - Various settings tested for optimal model performance such as:
        - Learning Rate
        - Epochs
        - Hidden Layers
        - Batch Size
        - Optimizer
        - Weight Initialization
        - Regularization
        - Activation Functions
        - Loss Function
        - Early Stopping
        - Number of Layers

- Model Saving
    - Save the final model for use in deployment environments, enabling real-time detection.

5. Authorization Matrix

Role	Access Level
Data Scientist	Full access to model training and tuning processes
Analyst	Access to trained model and its outputs for threat analysis
Admin	Full access to system resources and document

6. Assumptions

- The dataset is pre-processed and contains relevant traffic patterns for benign and DDoS scenarios.
- Adequate computational resources are available for training the GNN.
- Model evaluation metrics (accuracy, precision, recall) are pre-defined for validation.

7. Target Audience

Audience Data Scientists, AI Researchers, Cybersecurity Analysts.

8. Effort Estimation

- Model Architecture Design: 3 days
- Model Training: 5 days
- Hyperparameter Tuning: 7 days
- Documentation: 1 day
- Total: 16 days

9. Acceptance Criteria

- The GNN model achieves at least 75% accuracy in detecting DDoS attacks.
- Hyperparameters are tuned to optimize performance.
- The model differentiates between DDoS and benign traffic.
- Training procedures are well documented, and the trained model is saved for deployment.

10. Checklist

- Model architecture designed and implemented.
- Dataset pre-processed and ready for training.
- GNN model trained and validated.
- Hyperparameters tuned to optimize detection performance.
- Model saved for deployment.
- Documentation completed.