CHAPTER 6
RESULTS AND DISCUSSIONS

This chapter evaluates the AI-Driven Dynamic Fuzz Testing approach using Graph Neural Networks (GNNs) for detecting and mitigating DDoS attacks in IoT networks. It discusses the training and validation accuracy of the GNN model, as well as the effectiveness of the mitigation strategy in improving network performance during attacks. Visualizations of the model's behavior and network traffic are also presented, demonstrating the system's capability to enhance IoT security.

6.1 Introduction

In modern IoT networks, the growing number of connected devices has increased risks in the DDoS attack concerning security. Most of the devices are characterized as heterogeneous in terms of the mesh of networks through which they are connected; it alone is a rich playground for malicious entities to launch attacks using compromised devices as vectors for widespread damage. This problem is beyond the capabilities of traditional security mechanisms because most IoT devices are resource-constrained, and the attack patterns change dynamically, evolve over time, and make use of space and signature-based techniques.

To address this problem, we created an AI-Driven Dynamic Fuzz Testing approach using GNNs that is tested on the NS3 network simulation to identify and counter DDoS attacks.

6.2 Results

As shown clearly in the results below, the approach identifies malicious traffic patterns while using an active mitigation strategy to safeguard IoT systems.

6.2.1 Training and Accuracy of the GNN Model

We used the synthetic dataset produced with large NS3 simulations for our training. We included both benign and DDoS traffic to make the model learn from both types of patterns. The class balance of the dataset was ensured before training, making sure that the two types were equal. We show two plots below showing training loss and accuracy:

The training loss, as depicted in the first graph, displays a very sharp dip during the preliminary stages of training. The learning by the model is represented here. At the initial stages, the loss was high because the data was unknown to the GNN. However, training loss declined very sharply down to 0 as epochs proceeded, reflecting that the model correctly identified the patterns from the data. At around 40 epochs, the loss curve stabilizes and becomes around 0, indicating convergence of the model.

The second plot is the training accuracy: it oscillates around 50%, which might imply that the model was simply wandering through its features and updating its weights with back-propagation. Accuracy increases steadily with epochs, peaking at a high of 100%, implying that GNN learned to classify the training data with full accuracy. But training with 100% accuracy may overfit; sometimes the model might go very well on the training data but may not generalize well using new unseen data.

6.2.2 Results of Validation

While impressive accuracy on the training set, the real strength of the model is in performance over unseen data. We judged the GNN on a validation set extracted from an independent set of simulations using NS3. This validation dataset also contained an equal proportion of benign and DDoS traffic compared to the training data.

6.2.2.1 Validation Accuracy:

The average validation accuracy achieved by the GNN model is 74%. While this is quite lower than the corresponding training accuracy, it still indicates that the model learns well from the data. Lower validation accuracy generally means that, in most cases, the model would correctly detect DDoS traffic while normally and effectively responding; however, there are probably some edge cases or variations in the attack patterns that it has not been able to classify correctly. This promises a good level of performance since it will be deployed in real-world scenarios of new evolving types of DDoS attacks.

6.2.2.2 Confusion Matrix:

The confusion matrix for the validation results shows even further the model's ability to distinguish between benign and malicious traffic. It presents a relatively healthy scale of true positives, which are the correct identification of DDoS attacks, and true negatives, referring to the correct identification of benign traffic. Moreover, the matrix also indicates false positives and false negatives. In practice, a false positive (major classification of benign traffic as malicious) might prompt unwarranted mitigating steps, whereas a false negative DDoS traffic will not be detected, permitting attacks to run amok. Minimizing these types of errors will, therefore, be critical in increasing the reliability of the system.

6.2.3 Mitigation Strategy

Detection is only half the equation; mitigation is equally important so that the IoT network stays healthy. The GNN identifies malicious traffic, and then the strategy for packet filtering engages to block packet traffic coming from those malicious IP addresses.

6.2.3.1 Packet Dropping:

By using such a model in the NS3 simulation, the malicious IP addresses were easily traced, and their packets dropped. Packet filtering is the prime mitigation mechanism for DDoS traffic because it guards against resource exhaustion in the network by preventing flooding. The probability for packet dropping reduces congestion across the network. Therefore, legitimate traffic flows freely, thus enhancing system performance during an attack.

6.2.3.2 Performance on Network:

The state of the network before, during, and after deploying the mitigation strategy was evaluated by tracking several performance metrics. These include:

Throughput: Before the attack, throughput was reliable in the network. The legitimate devices communicated properly. During the attack, the throughput drastically dropped due to network traffic congestion from DDoS traffic. However, once the packet filtering mechanism was activated, the throughput returned to normal levels, implying reduced malicious traffic.

Latency: Like throughput, latency surged dramatically during the DDoS attack because much traffic was introduced to the network to process. Once mitigation occurred, latency returned to normal since the filtering technique forced traffic back into the acceptable network congestion zone.

6.3 Visualization of Results

To explain in greater detail the GNN model's behaviour as well as the effectiveness of the mitigation strategy, we have used several visualization methods throughout the course of this project.

6.3.1 Loss and Accuracy Graphs:

The training procedure is followed by metrics tracking in terms of the loss and accuracy at each epoch. The graphs shown in the following images represent the learning process of the model.

It is proved that the loss curve decreases because the model learns every detail of the data; this is reflected by the graph on accuracy that improves rapidly for the model in terms of its ability to classify objects.

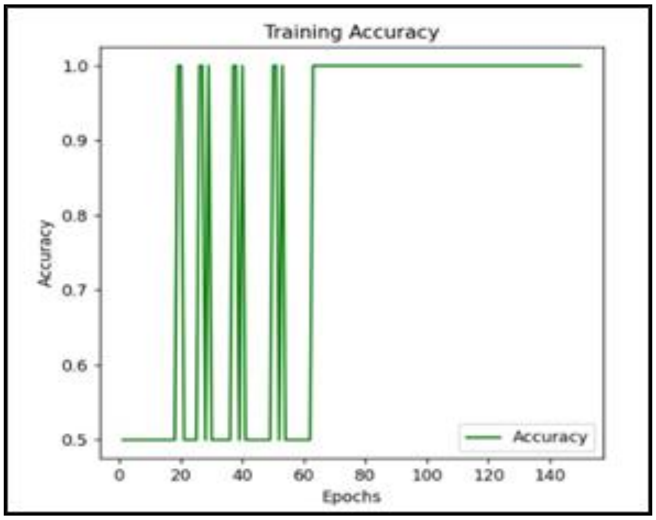

Fig. 6.1 GNN Model Training Accuracy Graph

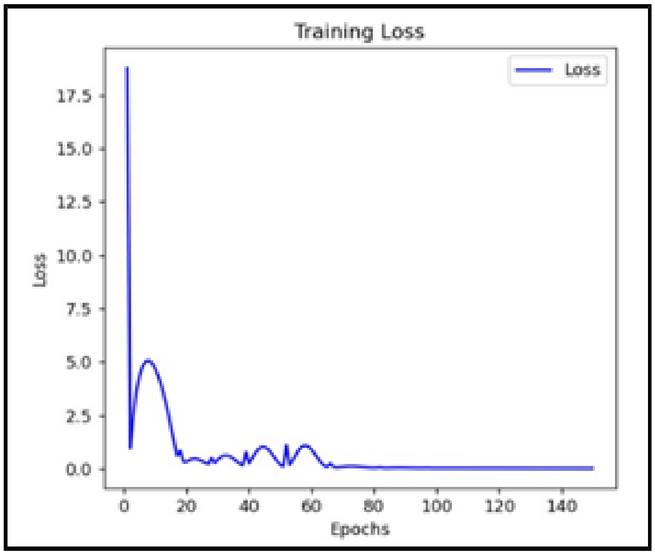

Fig. 6.2 GNN Model Training Loss Graph

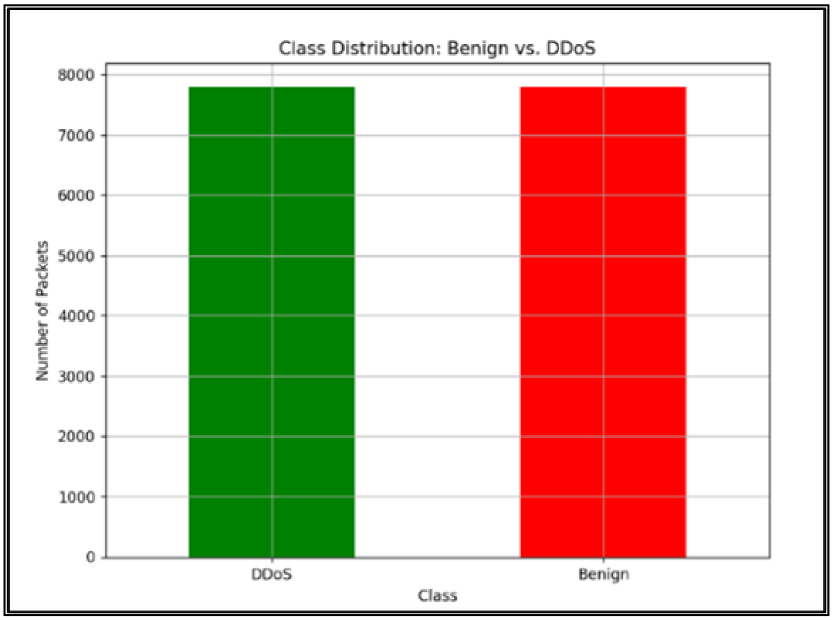

Fig. 6.3 Class Distribution Graph

6.3.2 Confusion Matrix:

The confusion matrix will be represented to analyse how well the model distinguishes between benign and DDoS traffic, and the visualization may reveal patterns in false positives and negatives, helping identify regions where the model may need additional tuning.

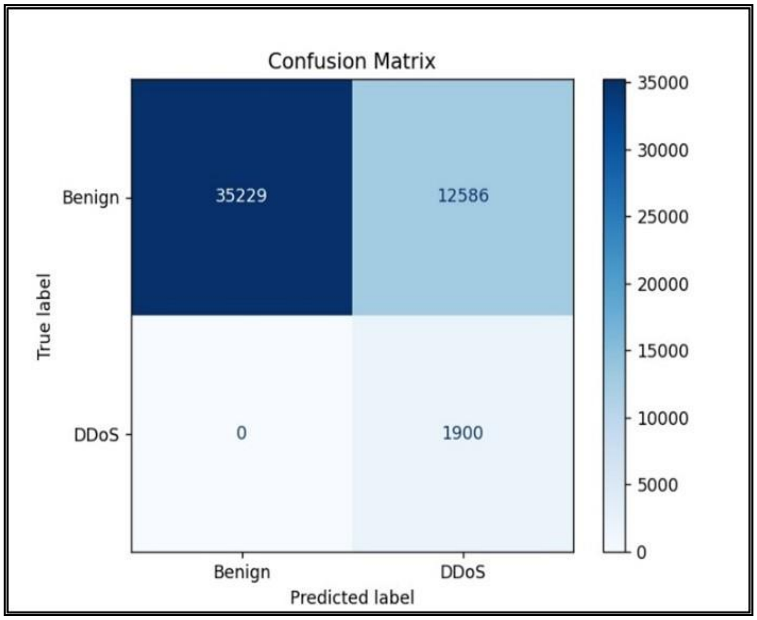

Fig. 6.4 Confusion Matrix

6.3.3 Network Animation:

The NS3 simulation results came alive using the NetAnim tool, visualizing the network topology, the flow of traffic, and the impact of the applied mitigation strategy. Such animation enables us to visualize the whole interaction between nodes during the attack as well as how the patterns of traffic changed once packet filtering was applied. It also provides a perspective into how the nodes were distributed in space and the nature of the traffic movement between them, which is vital to understand the true effects of the mitigation strategy in real time.

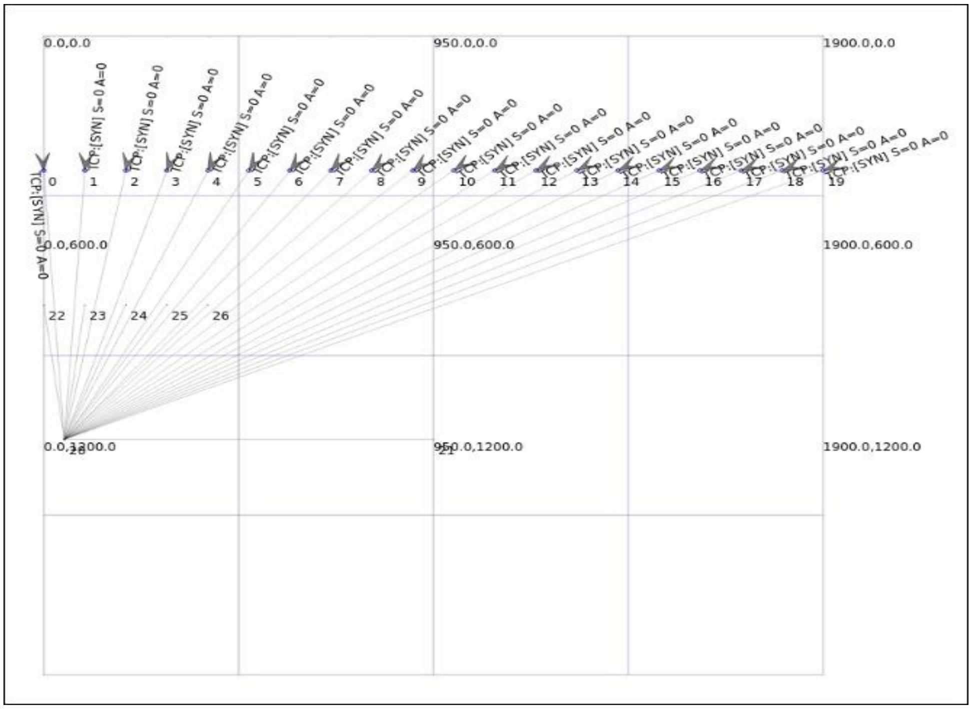

Fig. 6.5 Network Animation Simulation