Deep Ensemble Learning With Model Pruning for Efficient Ddos Attack Detection in IOT Networks

Mr. Shashank Tiwari[*1], Guguloth Pravallika[*2], Thadoori Varintej[*3], K Manikanta[*4]

[*1]Assistant Professor Of Department Of CSE (AI & ML), ACE Engineering College Hyderabad, India.

[*2,3,4]Department CSE (AI & ML) Of ACE Engineering College Hyderabad, India.

ABSTRACT

The Deep Ensemble Learning with Model Pruning for DDoS Detection in IoT Networks is a security framework designed to detect distributed denial-of-service (DDoS) attacks in IoT environments. The system analyzes network traffic features such as packet rate, flow duration, and protocol type to identify malicious behavior. It combines multiple deep learning models, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Deep Neural Networks (DNN), using an ensemble approach to improve detection accuracy. Model pruning is applied to reduce computational complexity and make the system suitable for resource-constrained IoT devices. This approach enables efficient and reliable real-time detection of DDoS attacks in IoT networks.

Keywords: Deep Ensemble Learning, DDoS Attack Detection, Internet of Things (IoT), Model Pruning, Network Security, Deep Learning, Intrusion Detection System.

I. INTRODUCTION

The rapid growth of Internet of Things (IoT) devices has increased the risk of cyber threats, particularly Distributed Denial of Service (DDoS) attacks that disrupt network services. Traditional detection methods often struggle to identify complex attack patterns and may not perform efficiently in IoT environments. This work proposes a **Deep Ensemble Learning approach with Model Pruning** to detect DDoS attacks in IoT networks. The system analyzes network traffic features and combines multiple deep learning models, including **CNN, LSTM, and DNN**, to improve detection accuracy. Model pruning is applied to reduce computational complexity, enabling efficient and real-time deployment in IoT networks..

II. LITERATURE SURVEY

Early Works

1. Deep Learning-Driven IoT Defence: CNN vs LSTM for DDoS Detection:

Patil, V. T., and Deore, S. S. (2021) – This study compared Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) models for detecting DDoS attacks in IoT networks. The research demonstrated that CNN models were more effective at identifying traffic patterns and achieved higher detection accuracy with faster processing. However, the work focused on individual models rather than combining multiple deep learning techniques to improve overall performance.

2. Enhanced LSTM for IoT DDoS Detection Using Honeypot Data:

Researchers in the International Journal of Computational Intelligence Systems proposed a hybrid model combining Conv1D, Bi-LSTM, and GRU architectures for detecting DDoS attacks. The model improved sequential learning of network traffic behavior and achieved high detection accuracy with reduced false alarms. Although the method improved detection performance, the model complexity increased computational requirements, making real-time deployment challenging in resource-limited IoT devices.

DOI: 10.55041/IJSREM57929

3. Ensemble Deep Learning for IoT Intrusion Detection:

A study published in MDPI Applied Sciences introduced an ensemble deep learning framework that combined CNN, GRU, and LSTM models for intrusion detection in IoT networks. The ensemble approach used a voting mechanism to integrate predictions from multiple models, which significantly improved detection accuracy compared to single-model systems. However, the ensemble model required high computational resources and did not address efficiency optimization for IoT environments.

4. LocKedge: Low-Complexity Cyberattack Detection in IoT Edge Computing:

Truong Thu Huong et al. proposed a lightweight deep learning-based intrusion detection system designed for edge computing environments. The framework focused on reducing computational overhead while maintaining accurate detection of cyberattacks. Although the model improved efficiency, it did not utilize ensemble learning or advanced optimization techniques such as model pruning.

OBJECTIVES:

- **Accurate Detection of DDoS Attacks**
  The primary objective of this project is to detect Distributed Denial of Service (DDoS) attacks in IoT networks by analyzing network traffic patterns and identifying abnormal behavior that disrupts normal network services.
- **Use of Deep Ensemble Learning Models**
  The project aims to improve detection performance by combining multiple deep learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Deep Neural Networks (DNN). The ensemble approach allows the system to capture different characteristics of network traffic and make more reliable predictions.
- **Use of Deep Ensemble Learning Models**
  The project aims to improve detection performance by combining multiple deep learning models such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Deep Neural Networks (DNN). The ensemble approach allows the system to capture different characteristics of network traffic and make more reliable predictions.
- **Real-Time Deployment in IoT Environments**
  The system is designed to operate efficiently in real-time IoT networks by analyzing traffic features such as packet rate, protocol type, and flow duration, enabling faster detection and mitigation of DDoS attacks in resource-constrained devices.

III. METHODOLOGY

The **Deep Ensemble Learning with Model Pruning for DDoS Detection in IoT Networks** framework integrates network traffic monitoring, deep learning–based classification, ensemble decision making, and model optimization to accurately detect DDoS attacks in IoT environments while maintaining computational efficiency.

1. Data Collection & Preprocessing

User deploys the detection system at an **IoT gateway or monitoring node** → Network traffic is captured from connected IoT devices → Raw packet data is collected from public IoT intrusion detection datasets or live traffic → Data preprocessing is performed to remove noise, handle missing values, and normalize features → Important network traffic features such as packet rate, flow duration, protocol type (TCP, UDP, ICMP), packet size, and source–destination behavior are extracted for model training.

2. Deep Learning-Based Traffic Analysis:

• **Feature Extraction:** Preprocessed network traffic features are prepared as input for deep learning models.
• **CNN Model:** Convolutional Neural Networks analyze spatial patterns in network traffic and identify abnormal packet behavior.
• **LSTM Model:** Long Short-Term Memory networks capture temporal patterns and sequential dependencies in traffic

flows.
- **DNN Model:** A Deep Neural Network performs feature-based classification to distinguish between normal and attack traffic.
- **Prediction Generation:** Each model independently predicts whether the network traffic is normal or malicious.

3. Ensemble Decision Mechanism

- **Model Integration:** Outputs from CNN, LSTM, and DNN models are combined using ensemble techniques such as voting or averaging.
- **Final Classification:** The ensemble system produces a final prediction with improved accuracy and reduced false positives.
- **Attack Identification:** The system classifies traffic as either normal network activity or a DDoS attack.

4. Model Optimization using Pruning

- **Parameter Reduction:** Model pruning removes unnecessary neurons and weights from trained models.
- **Efficiency Improvement:** This reduces memory consumption and computational overhead.
- **Lightweight Deployment:** The optimized model becomes suitable for real-time execution on resource-constrained IoT devices and gateways.

5. Real-Time Detection & Monitoring

- **Traffic Monitoring:** Incoming network traffic is continuously analyzed in real time.
- **Attack Detection:** If abnormal traffic patterns exceed predefined thresholds, the system identifies them as potential DDoS attacks.
- **Alert & Logging:** Detected attacks are logged and alerts can be generated for system administrators.

Key Components:

- **User Interface / Dashboard:** Python-based interface (Streamlit or Flask) for traffic input, monitoring, and result visualization.
- **Deep Learning Framework:** TensorFlow/Keras used to implement CNN, LSTM, and DNN models for DDoS detection.
- **Data Processing:** NumPy, Pandas, and Scikit-learn used for traffic preprocessing, feature extraction, and evaluation.
- **Network Monitoring:** Packet analysis tools used to capture and analyze TCP, UDP, and ICMP traffic patterns.
- **Development Environment:** Python, Jupyter Notebook or PyCharm with Git/GitHub for development and deployment.

IV. PROPOSED SYSTEM

The proposed system uses **Deep Ensemble Learning with Model Pruning** to detect DDoS attacks in IoT networks. It analyzes network traffic features such as packet rate, flow duration, and protocol type to identify abnormal behavior. Multiple deep learning models including **CNN, LSTM, and DNN** are combined using an ensemble approach to improve detection accuracy. Model pruning is applied to remove unnecessary parameters, reducing computational cost and enabling efficient **real-time deployment in IoT environments**..

System Overview

The proposed system includes:

- **Traffic Monitoring** – Captures network traffic from IoT devices to analyze communication patterns.
- **Feature Extraction** – Extracts important traffic features such as packet rate, flow duration, and protocol type.
- **Deep Learning Models** – Uses CNN, LSTM, and DNN models to learn traffic behavior and detect attack patterns.
- **Ensemble Decision** – Combines predictions from multiple models to improve detection accuracy.

- **Model Pruning** – Removes unnecessary parameters to reduce model size and improve execution efficiency.
- **Attack Detection** – Identifies abnormal traffic and classifies it as a DDoS attack for real-time network protection.

System Operation

1. Data Collection & Preprocessing Phase

Network traffic is captured from IoT devices or public intrusion detection datasets → Data is cleaned and normalized → Important traffic features such as packet rate, flow duration, packet size, and protocol type (TCP/UDP/ICMP) are extracted for analysis.

2. Model Training & Ensemble Phase

Extracted features are used to train multiple deep learning models including **CNN, LSTM, and DNN** → Each model learns different traffic patterns → Predictions from these models are combined using an **ensemble approach** to improve detection accuracy.

3. Optimization & Detection Phase

Model pruning removes unnecessary parameters to reduce model size and computational cost → The optimized model monitors incoming traffic in real time → Abnormal patterns are detected and classified as **DDoS attacks**, enabling faster network protection.

Hardware & Software Components

- **Frontend:** Python-based interface (Streamlit/Flask) for monitoring traffic and displaying results
- **Backend:** Python with TensorFlow/Keras for deep learning model implementation
- **Data Processing:** NumPy, Pandas, Scikit-learn for preprocessing and evaluation
- **Tools & Environment:** PyCharm, Jupyter Notebook, Git/GitHub, and local/cloud deployment
- **Hardware:** IoT devices, network gateway, and central server for model training and monitoring

V. APPLICATIONS

The proposed **Deep Ensemble Learning with Model Pruning framework** is designed to detect and prevent Distributed Denial of Service (DDoS) attacks in IoT networks. The system analyzes network traffic patterns and applies optimized deep learning models to identify abnormal behavior in real time.

Traffic Feature Extraction Algorithm

Purpose: Extracts important network traffic features used for attack detection.

Algorithm Steps:

1. Capture network packets from IoT devices or network monitoring systems.
2. Extract key traffic features such as packet rate, flow duration, protocol type, and packet size.
3. Normalize and preprocess the extracted data.
4. Store the processed features for model training and prediction.

Deep Learning Detection Algorithm

Purpose: Classifies network traffic as normal or malicious using deep learning models.

Algorithm Steps:

1. Input the extracted traffic features into the trained models.
2. CNN analyzes spatial patterns in network traffic data.
3. LSTM learns sequential behavior and time-based traffic patterns.

4. DNN performs feature-based classification.
5. Each model produces a prediction for the traffic sample.

Ensemble Decision Algorithm

Purpose: Combines predictions from multiple models to improve detection accuracy.

Algorithm Steps:

1. Collect predictions from CNN, LSTM, and DNN models.
2. Apply ensemble techniques such as voting or averaging.
3. Generate a final prediction for the traffic sample.
4. Classify the network activity as normal or DDoS attack.

Model Pruning Optimization Algorithm

Purpose: Reduces model complexity to improve execution efficiency.

Algorithm Steps:

1. Identify redundant neurons and parameters in trained models.
2. Remove unnecessary weights without affecting model performance.
3. Retrain the pruned model to maintain accuracy.
4. Deploy the optimized lightweight model for real-time detection.

Real-Time Monitoring Algorithm

Purpose: Continuously monitor network traffic and detect attacks.

Algorithm Steps:

1. Monitor incoming network traffic in real time.
2. Extract features from new traffic data.
3. Run the pruned ensemble model for prediction.
4. If abnormal traffic is detected, classify it as a potential DDoS attack.
5. Log the detection results for network monitoring and analysis.

VI. RESULT

System Performance Evaluation

The Deep Ensemble Learning with Model Pruning system was evaluated using network traffic datasets to measure its ability to accurately detect DDoS attacks in IoT environments.

Testing confirmed:

• **Accurate Detection**: The ensemble model (CNN, LSTM, DNN) successfully identified malicious traffic patterns with high detection accuracy.

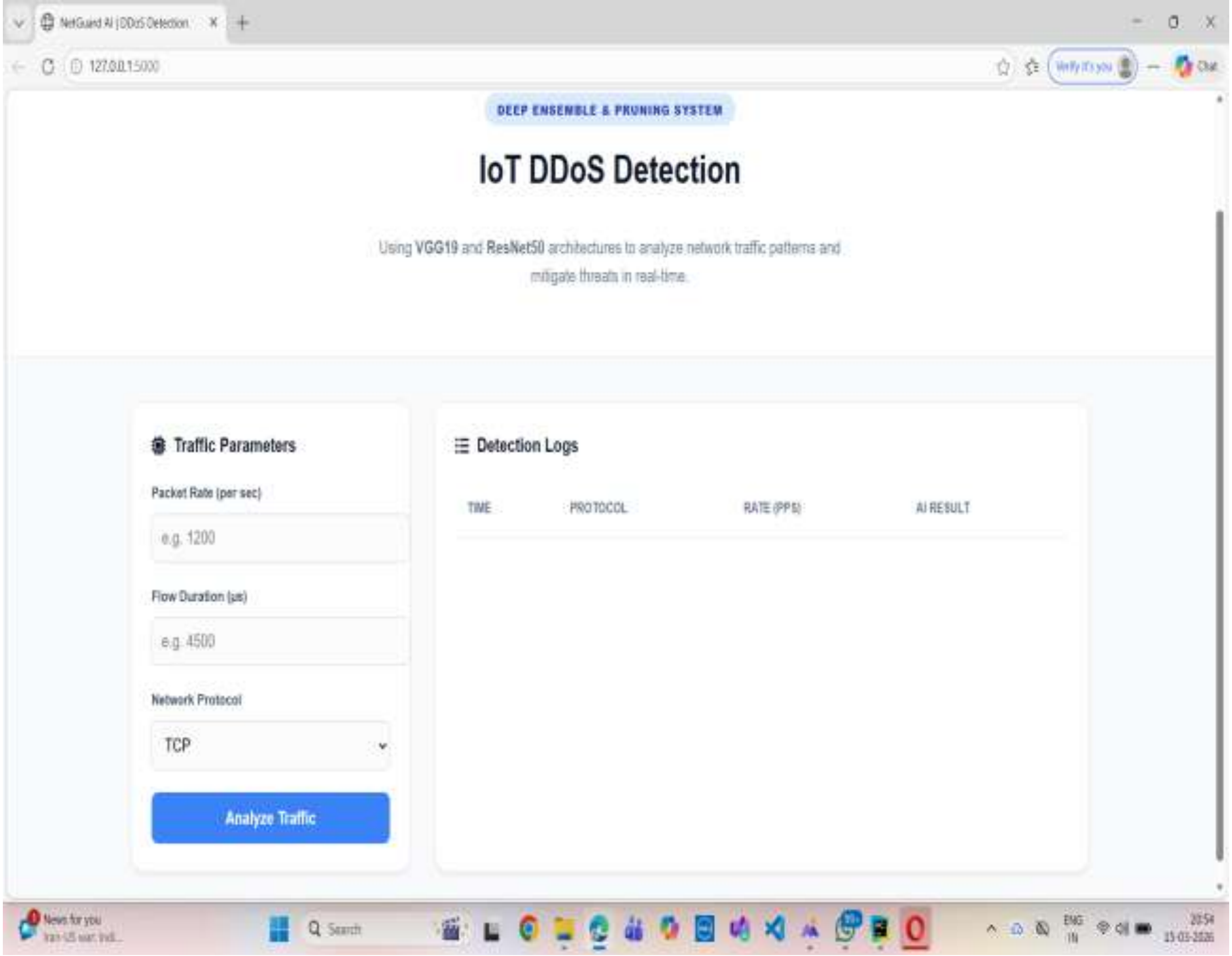
• **Efficient Feature Analysis**: The system effectively analyzed network traffic features such as packet rate, flow duration, and protocol type.

• **Reduced False Positives**: The ensemble learning approach improved prediction reliability compared to individual models.

• **Optimized Performance**: Model pruning reduced model complexity and computational overhead without significantly affecting accuracy.

• **Real-Time Capability**: The optimized model was able to process network traffic efficiently, enabling faster detection suitable for IoT environments.

The system demonstrated reliable performance and improved efficiency, showing that deep ensemble learning with model pruning is an effective approach for real-time DDoS detection in IoT networks.



DEEP ENSEMBLE & PRUNING SYSTEM

IoT DDoS Detection

Using VGG19 and ResNet50 architectures to analyze network traffic patterns and mitigate threats in real-time.

Traffic Parameters

Packet Rate (per sec): e.g. 1200

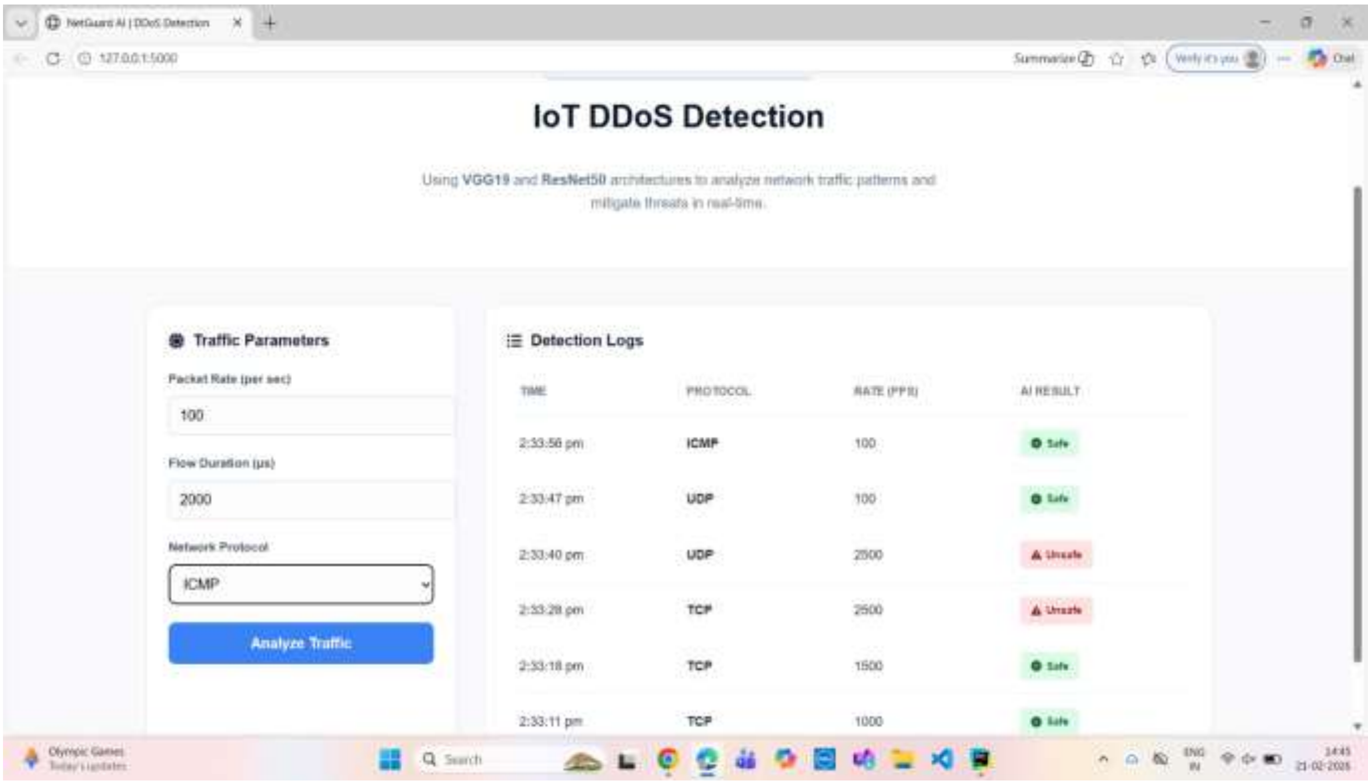
Flow Duration (µs): e.g. 4500

Network Protocol: TCP

Analyze Traffic

Detection Logs

TIME	PROTOCOL	RATE (PPS)	AI RESULT



IoT DDoS Detection

Using VGG19 and ResNet50 architectures to analyze network traffic patterns and mitigate threats in real-time.

Traffic Parameters

Packet Rate (per sec): 100

Flow Duration (µs): 2000

Network Protocol: ICMP

Analyze Traffic

Detection Logs

TIME	PROTOCOL	RATE (PPS)	AI RESULT
2:33:56 pm	ICMP	100	● Safe
2:33:47 pm	UDP	100	● Safe
2:33:40 pm	UDP	2500	⚠ Unsafe
2:33:28 pm	TCP	2500	⚠ Unsafe
2:33:18 pm	TCP	1500	● Safe
2:33:11 pm	TCP	1000	● Safe

VII. CONCLUSION

The proposed **Deep Ensemble Learning with Model Pruning framework** provides an effective solution for detecting DDoS attacks in IoT networks. By combining multiple deep learning models such as **CNN, LSTM, and DNN**, the system improves detection accuracy and reduces false positives compared to single-model approaches. The integration of **model pruning** optimizes the model by reducing unnecessary parameters, making it suitable for resource-constrained IoT environments. The results demonstrate that the proposed approach can efficiently analyze network traffic patterns and identify malicious activities in real time, providing a reliable and lightweight solution for enhancing IoT network security.

VIII. FUTURE ENHANCEMENT

The proposed **Deep Ensemble Learning with Model Pruning framework** can be further improved to enhance DDoS detection capabilities in IoT networks. Future developments will focus on:

1. **Advanced Attack Detection** – Extending the system to detect additional cyber threats such as intrusion attempts, malware attacks, and network anomalies.
2. **Adaptive Model Optimization** – Implementing dynamic pruning and model compression techniques to further improve efficiency for resource-constrained IoT devices.
3. **Edge-Based Deployment** – Integrating the detection system directly into IoT gateways or edge devices for faster real-time monitoring and response.
4. **Larger Dataset Integration** – Training the model with more diverse and large-scale network datasets to improve robustness and detection accuracy.
5. **Automated Response Mechanisms** – Developing automated mitigation strategies such as traffic filtering, IP blocking, and alert systems to respond to detected attacks instantly.

IX. REFERENCES

[1] V. T. Patil and S. S. Deore, “Deep learning-driven IoT defence: CNN vs LSTM for DDoS detection,” *International Journal of Network Security*, 2021.

[2] T. T. Huong, P. H. Nam, and D. N. Nguyen, “LocKedge: Low-complexity cyberattack detection in IoT edge computing,” *IEEE Internet of Things Journal*, 2020.

[3] A. O. and A. A. T., “Ensemble deep learning for intrusion detection in IoT networks,” *Applied Sciences*, MDPI, 2021.

[4] H. Li, K. Ota, and M. Dong, “Learning IoT in edge: Deep learning for the Internet of Things with edge computing,” *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[5] M. Chiang and T. Zhang, “Fog and IoT: An overview of research opportunities,” *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[7] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, 2015.