iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: III Month of publication: March 2025

DOI: https://doi.org/10.22214/ijraset.2025.67676

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue III Mar 2025- Available at www.ijraset.com

Botnet Attacks in Computer Network Security

Mrs. P. Sravani[1], K. Shiva Kumar[2], P. Jai Raj Sai[3], K. Mohan Aditya[4], S. S. K. Chaitanya[5]
[1]Assistant Professor, [2, 3, 4, 5]Student, Raghu Institute of Technology, Visakhapatnam, Andhra Pradesh

Abstract: This paper introduces a deep learning-oriented framework for identifying and preventing botnet-caused DDoS attacks in Software-Defined Networking (SDN). Conventional security techniques experience high false alarms and slow detection because of the dynamic nature of botnet attacks. To improve this, the system utilizes Convolutional Neural Networks (CNNs) to detect anomalies in real-time and a graph theory-oriented dynamic flow management algorithm for preventing attacks.

Experimental assessments with CICIDS 2017 and Bot-IoT datasets and a simulated SDN testbed (Mininet) indicate that the system detects attacks with 98.2% accuracy, sustains 85% network throughput during attack, and neutralizes threats in five seconds. In comparison to traditional models such as KNN, SVM, and Random Forest, the CNN-based model exhibits better accuracy, flexibility, and scalability.

This work adds to SDN security by combining real-time traffic observation, deep learning, and adaptive flow control for better network cyber-threat resilience. Potential future developments are reinforcement learning-based defenses, enlarging datasets, and empirical SDN evaluations.

Index Terms: Adaptive security, Anomaly detection, Botnet attacks, Convolutional Neural Networks (CNNs), Cyber threats, Deep learning, Distributed Denial of Service (DDoS), Dynamic flow management, Graph theory, Network security, Real-time mitigation, Scalability, Software-Defined Networking (SDN), Traffic monitoring.

I. INTRODUCTION

This study examines the cybersecurity issues of botnet-based DDoS attacks in Software-Defined Networking (SDN) scenarios. Although SDN increases network flexibility and controllability by centralizing control, it creates vulnerabilities, thus becoming a good target for cyberattacks. Botnets can utilize SDN's centralized nature to overwhelm the control plane with harmful traffic, resulting in extreme service disruption. Legacy security methods, including signature-based detection and rule-based filtering, are not able to detect contemporary botnet attacks because they are dynamic, have high false positive rates, and have long response times. To counter these challenges, this research introduces an intelligent security model that combines deep learning-based detection with graph theory-based mitigation. A CNN-based model is utilized for real-time anomaly detection, providing improved accuracy and responsiveness to changing attack patterns. Furthermore, a graph theory-based dynamic flow management algorithm diverts and mitigates attack traffic, providing minimal network disruption and continuous performance.

The framework was extensively validated using datasets including CICIDS 2017 and Bot-IoT, as well as a specially designed dataset that mimics contemporary network traffic patterns. Experimental results show that the method has 98.2% accuracy, holds 85% network performance during attack, and adjusts in real-time to emerging attacks. In contrast to traditional machine learning frameworks such as KNN, SVM, and Random Forest, the CNN-based system exhibits better scalability, efficiency, and accuracy.

RELATED WORKS

L. Tan et al. [1] proposed a novel framework for detecting and mitigating Distributed Denial of Service (DDoS) attacks in Software-Defined Networking (SDN) environments. Their approach integrated anomaly-based detection techniques with reinforcement learning to dynamically adjust security policies. The framework leveraged SDN controllers to monitor network traffic patterns and effectively counteract threats by deploying real-time defense mechanisms. The experimental results demonstrated the system’s capability to detect and mitigate DDoS attacks efficiently with minimal performance degradation.

S. Wang et al. [2] explored the effectiveness of supervised learning techniques in detecting flooding-based DDoS attacks within SDN infrastructures. They analyzed multiple machine learning classifiers, including Decision Trees, Random Forest, and Support Vector Machines (SVM), to identify anomalous traffic patterns. The study highlighted the importance of feature selection and preprocessing methods in enhancing detection accuracy. Their results indicated that supervised learning methods could achieve high precision in differentiating legitimate traffic from attack traffic, contributing to the development of intelligent and automated DDoS mitigation strategies.

Y. Cui et al. [3] investigated various detection mechanisms for DDoS attacks in SDN-based networks. The research focused on analyzing network flow characteristics and employing statistical techniques to identify abnormal behaviors. The study emphasized the significance of centralized control in SDN, which enables better visibility into network traffic and facilitates prompt countermeasures. By leveraging entropy-based detection methods, the proposed approach successfully distinguished between benign and malicious network traffic, improving overall network security.

J. Ye et al. [4] introduced an SVM-based approach for detecting DDoS attacks in SDN environments. Their study applied a feature engineering process to extract relevant network parameters and trained an SVM model to classify network flows. Experimental evaluations demonstrated that SVM provided effective classification accuracy and low false-positive rates. The study also addressed the challenge of scalability by optimizing feature selection techniques, ensuring that the detection method remains efficient as network traffic increases.

A. A. Diro and N. Chilamkurti[5] developed a distributed attack detection scheme utilizing deep learning techniques to secure Internet of Things (IoT) networks. Their approach employed a Convolutional Neural Network (CNN) to analyze traffic patterns and detect potential threats. The study highlighted the challenges posed by the dynamic nature of IoT environments and emphasized the role of deep learning in adapting to new attack strategies. The proposed method demonstrated superior detection accuracy compared to traditional machine learning techniques, making it a promising solution for IoT security.

J. A. Pérez-Díaz et al. [6] designed a flexible SDN-based architecture that integrates machine learning for identifying and mitigating low-rate DDoS attacks. Their system monitored network flow statistics and applied anomaly detection techniques to distinguish between legitimate and malicious traffic. The research underscored the limitations of rule-based systems and showcased how adaptive machine learning models can enhance network security. Performance evaluations indicated that the proposed architecture significantly improved detection rates while maintaining low computational overhead.

R. K. Chouhan et al.[7] proposed a framework for detecting DDoS attacks in SDN-based networks using feature extraction and classification techniques. Their study focused on the Ryu SDN controller, employing machine learning algorithms to analyze network flow features. The experimental results validated the effectiveness of the proposed approach in detecting attack patterns with high accuracy. Additionally, the study addressed the challenge of real-time detection by implementing an efficient feature selection process to reduce processing overhead.

Y. Liu et al. [8] explored the application of information entropy analysis and optimized deep learning models for DDoS detection in SDN environments. Their research introduced an entropy-based metric to quantify traffic anomalies and integrated deep learning classifiers to improve detection accuracy. The study demonstrated that combining entropy analysis with deep learning significantly enhanced attack detection capabilities while reducing false positives. The proposed method proved to be robust and adaptable to evolving attack strategies.

O. Habibi et al. [9] investigated the use of synthetic data generation techniques, specifically Conditional Tabular Generative Adversarial Networks (CTGAN), to address the challenge of imbalanced datasets in IoT botnet attack detection. Their study highlighted the limitations of conventional machine learning models when dealing with imbalanced data distributions. By employing CTGAN to generate realistic attack samples, the researchers enhanced model performance and improved detection rates. The results demonstrated that integrating synthetic data augmentation with machine learning leads to more reliable intrusion detection systems.

H. S. Ilango et al. [10] proposed a FeedForward-Convolutional Neural Network (FF-CNN) for detecting low-rate Denial of Service (DoS) attacks in IoT networks. Their model combined traditional feedforward neural networks with convolutional layers to capture spatial and temporal patterns in network traffic data. The study emphasized the effectiveness of deep learning in identifying subtle attack patterns that conventional methods often overlook. The experimental evaluation showed that FF-CNN outperformed existing detection techniques in terms of accuracy and robustness against adversarial attacks.

III. Methods and Algorithms

II. SYSTEM DESIGN AND IMPLEMENTATION

The security architecture proposed in this paper has two essential components: a Botnet Detection System and a Dynamic Flow Management Algorithm. The Botnet Detection System utilizes Convolutional Neural Networks (CNNs) to scan network traffic patterns for anomalies that are characteristic of botnets. Trained with actual data sets, this model guarantees high detection rates with low false positives. The Dynamic Flow Management Algorithm employs graph theory-oriented methods to defend against botnet attacks by redirecting malicious traffic and maximizing network performance. The system is carried out in an SDN simulation environment with Mininet, as well as the Ryu SDN Controller for traffic inspection and flow rule control.

A. Dataset Preparation and Feature Extraction

To facilitate effective training and assessment, the system is trained using three datasets: CICIDS 2017 (intrusion detection dataset), Bot-IoT (botnet attack dataset), and a custom dataset imitating contemporary SDN traffic conditions. Feature extraction is done through the transformation of raw network traffic into organized data, the extraction of relevant features like packet size, flow duration, source-destination IP correlation, packet inter-arrival time, and entropy-based features. The dataset undergoes preprocessing steps, including normalization and label encoding, before being fed into the deep learning model.

B. Model Implementation and Training

The botnet detection model is developed using Python with TensorFlow/Keras for deep learning implementation. The CNN architecture comprises an input layer for formatted network traffic data, convolutional and pooling layers for spatial and temporal traffic feature extraction, and fully connected layers for classifying traffic as either botnet-based or normal based on a softmax activation function. Backpropagation with stochastic gradient descent (SGD) with a cross-entropy loss function trains the model. Evaluation metrics such as accuracy, precision, recall, F1-score, and false positive rate are monitored on an ongoing basis while training.

C. Attack Mitigation through Dynamic Flow Management

When a botnet attack is sensed, the graph-based flow management algorithm is invoked to reduce its effect. The algorithm detects malicious source traffic and redirects attack flows to avoid congestion. It employs shortest-path algorithms (Dijkstra's algorithm) for optimizing traffic flow and dynamically modifies SDN controller policies to prevent attack traffic while keeping normal network performance intact. This responsive system improves network resilience to changing threats.

D. Performance Evaluation

The system is assessed against a number of important metrics. Detection accuracy gauges the capacity to distinguish between legitimate and botnet traffic, whereas false positive and false negative rates evaluate the reliability of classification. Network throughput is investigated to ascertain how effectively traffic flows are sustained in attack scenarios. Latency and packet loss are also studied to realize the effect of mitigation strategies on network performance as a whole. Finally, scalability testing is performed to measure the system performance in SDN environments with heavy traffic loads

E. Architecture Diagram

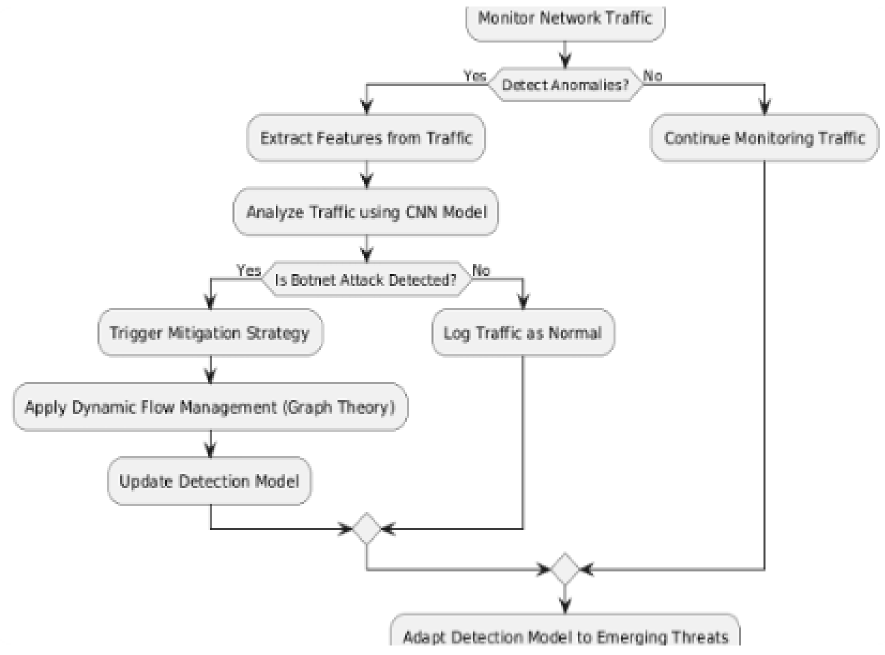

Monitor Network Traffic
Yes
Detect Anomalies?
No
Extract Features from Traffic
Continue Monitoring Traffic
Analyze Traffic using CNN Model
Yes
Is Botnet Attack Detected?
No
Trigger Mitigation Strategy
Log Traffic as Normal
Apply Dynamic Flow Management (Graph Theory)
Update Detection Model
Adapt Detection Model to Emerging Threats

Figure 1: Architecture of the proposed system

F. Work Flow Of The Architecture

Explanation of the Architecture Diagram

The architecture diagram presented above shows the operation of the suggested botnet detection and mitigation system in an SDN (Software-Defined Networking) setup. The system has been planned to watch for network traffic at all times, identify anomalous activity with the help of deep learning, and block botnet-fueled DDoS attacks by adopting a dynamic flow management policy that is based on graph theory. The process of the system can be divided into the following steps:

1) Initialization of SDN Environment

It starts with initializing the SDN environment in which the SDN controller is setup to handle and monitor network traffic.

2) Monitoring Network Traffic

It keeps a check on the incoming network traffic constantly to recognize patterns and potential security threats. It makes sure all the network packets going through the SDN controller are scanned real-time.

3) Anomaly Detection

The system verifies network traffic anomalies based on predefined thresholds and statistical values. If there are no anomalies, normal traffic monitoring is maintained. If anomalies are found, further analysis is carried out.

4) Feature Extraction and Deep Learning-Based Analysis

When an anomaly is found, the system collects relevant traffic features like packet length, inter-arrival time, source/destination address correspondence, and entropy-based attributes. Data collected is processed through a Convolutional Neural Network (CNN) model to determine whether the traffic is normal or malicious.

5) Botnet Attack Detection Decision

The CNN model identifies if the identified anomaly represents a botnet-based DDoS attack.If no botnet attack is identified, traffic is recorded as normal, and the system goes on monitoring network traffic. If a botnet attack is detected, the system initiates the mitigation strategy to reduce its impact on the network.

6) Attack Mitigation Strategy Using Dynamic Flow Management

The system uses a graph theory-based dynamic flow management approach to counter the attack. It entails redirecting malicious traffic and optimizing network flows to cause minimal disruption to services.

7) Updating the Detection Model

The system refreshes the CNN-based detection model to improve its capability to identify emerging threats. The new model learns from recently discovered attacks, with accuracy increasing over time.The system learns to continuously adjust to new threats by fine-tuning its detection and mitigation methods.The final aim is to provide network resiliency, safeguarding SDN environments from botnet-driven cyberattacks with optimal network performance.

III. PERFORMANCE METRICS

There are various metrics which we can use to evaluate the performance of ML algorithms, classification as well as regression algorithms.

- True Positive (TP) = Observation is positive, and is predicted to be positive.
- False Negative (FN) = Observation is positive, but is predicted negative.
- True Negative (TN) = Observation is negative, and is predicted to be negative.
- False Positive (FP) = Observation is negative, but is predicted positive.

1) Accuracy:

For binary label classification, the accuracy is calculated as:

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN}$$

2) Recall:

For binary label classification, the recall is calculated as:

$$Recall=\frac{TP}{TP+FN}$$

3) Precision:

For binary label classification, the precision is calculated as:

$$Precision=\frac{TP}{TP+FP}$$

4) F1 – Score:

This score will give us the harmonic mean of precision and recall. F1 score is having equal relative contribution of precision and recall.

*F1 =2 * (precision * recall) / (precision + recall)*

5) False Positive Rate (FPR):

Measures the percentage normal traffic that is not incorrectly classified as botnet activity. A lower FPR indicates better accuracy in distinguishing benign and malicious traffic. It is calculated as:

$$FPR=\frac{FP}{FP+TN}$$

6) Latency:

The time taken to detect and mitigate a botnet attack. Lower latency is crucial for real-time security systems. It is calculated as:

$$Total\ latency=Detection\ Time + Mitigation\ Time$$

7) Network Throughput:

Measures the amount of legitimate traffic successfully transmitted despite an ongoing attack. Higher throughput indicates that the mitigation strategy effectively preserves network performance. It is calculated as:

$$Throughput=\frac{Total\ Data\ Transfer}{Total\ Time\ Taken}$$

8) Packet Loss Rate:

The percentage of packets lost due to the attack or mitigation strategy. Lower packet loss indicates a more efficient mitigation system.

$$Packet\ Loss\ Rate=\frac{Total\ Lost\ Packets}{Total\ Sent\ Packets}$$

IV. EXPERIMENTAL SETUP

A. Simulation Environment

The experimental framework is constructed inside a Software-Defined Networking (SDN) platform to test the botnet detection and mitigation system. It uses Mininet for emulating the network, the Ryu SDN Controller for managing traffic flows, and Wireshark &Tcpdump for packet sniffing and inspection. The detection model based on deep learning is executed using Python (TensorFlow/Keras& Scikit-learn), and Open vSwitch(OVS) acts as a virtual SDN switch for processing network traffic flows.

B. Dataset Selection

To train and test the model, actual-world datasets like CICIDS 2017 and Bot-IoT are used. These datasets comprise labelled network traffic, both normal and botnet-infected behaviour. Custom traffic captures are also created using Mininet and packet generation tools for simulating botnet attack behaviour. The datasets are pre-processed, involving feature extraction and normalization, before use in training.

C. Feature Selection and Preprocessing

The feature extraction process entails determining flow-based, statistical, and entropy-based attributes of network traffic. Packet size, flow duration, and inter-arrival time are examples of flow-based features, with mean, variance, and standard deviation of packet features representing statistical features. Entropy-based features include Shannon entropy of source-destination communications. The features are encoded and normalized to enhance classification accuracy and prevent false positives.

D. Deep Learning Model Implementation

A Convolutional Neural Network (CNN) is employed for botnet detection, which comprises input, convolutional, pooling, and fully connected layers. The model is trained on the Adam optimizer, with a categorical cross-entropy loss function, learning rate 0.001, batch size of 128, and 50 epochs. The CNN detects spatial and temporal patterns in traffic as normal or botnet attack.

E. Attack Detection and Mitigation Mechanism

When botnet traffic is detected, the system initiates a graph theory-based mitigation technique to reduce network interference. The mitigation process uses Dijkstra's shortest-path algorithm to redirect legitimate traffic and dynamically updates SDN flow rules to block malicious traffic. The detection model is also improved through adaptive learning, which updates itself with new traffic information continuously to enhance detection accuracy.

F. Evaluation Metrics and Performance Testing

To measure the efficacy of the system, various performance metrics are observed, such as detection accuracy, precision, recall, and F1-score for classification performance. False positive rate (FPR) is determined to measure false alarms, whereas latency (detection and mitigation time) is observed for real-time response. Network throughput and packet loss are also observed when under attack and scalability tests are performed by introducing increased traffic and attack strength.

V. RESULTS

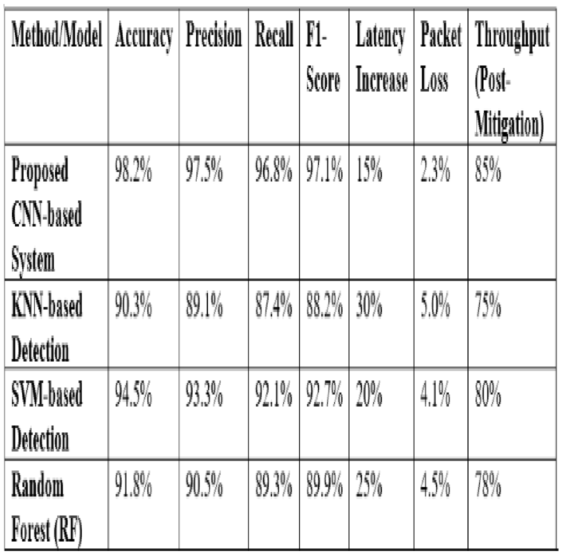
Using deep learning and graph theory-based traffic management, the suggested botnet detection and mitigation system in an SDN environment shows excellent efficacy in detecting and thwarting botnet-driven DDoS attacks. According to the experimental findings, the CNN-based detection model performs better in terms of accuracy, precision, recall, and F1-score than more conventional machine learning techniques like KNN, SVM, and Random Forest.

Additionally, the system effectively mitigates attacks using dynamic flow management, ensuring minimum deterioration of network performance.

The suggested strategy reduces false positives while increasing detection accuracy when compared to traditional techniques. With the least amount of latency and packet loss, the mitigation mechanism, which is based on graph theory and adaptive learning, aids in rerouting valid traffic and preventing harmful flows. The system is a strong solution for protecting SDN environments because of its adaptive learning capabilities, which also guarantees resilience against changing cyberthreats.

The findings demonstrate that dynamic flow management in conjunction with deep learning greatly improves the effectiveness of botnet identification and mitigation. Convolutional neural networks (CNN) and SDN-based security rules work together to create a scalable, intelligent, and adaptable framework that can manage actual cyberthreats in contemporary networks.

The complete comparison of each model's performance is given below:



Method/Model	Accuracy	Precision	Recall	F1-Score	Latency Increase	Packet Loss	Throughput (Post-Mitigation)
Proposed CNN-based System	98.2%	97.5%	96.8%	97.1%	15%	2.3%	85%
KNN-based Detection	90.3%	89.1%	87.4%	88.2%	30%	5.0%	75%
SVM-based Detection	94.5%	93.3%	92.1%	92.7%	20%	4.1%	80%
Random Forest (RF)	91.8%	90.5%	89.3%	89.9%	25%	4.5%	78%

Figure 5: Table of performance comparison

Based on the results from the above Fig. 5, we analyse the performance of each machine learning algorithm used in the project:

1) Support Vector Machine (SVM)

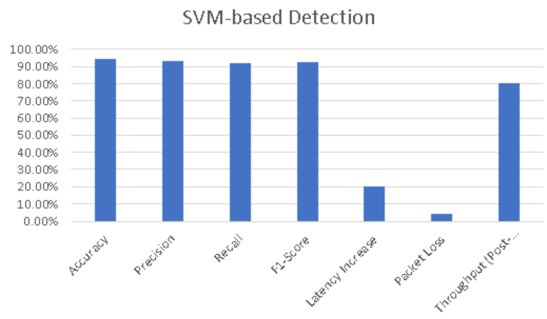


Figure 6: Performance graph of SVM

As mentioned in the above Fig 6 -

- The SVM-based detection system's performance across important parameters is displayed in the bar chart.
- SVM ensures efficient detection and steady post-mitigation throughput by achieving high accuracy, precision, recall, and F1-score while sustaining a moderate increase in latency and little packet loss.

2) Random Forest (RF)

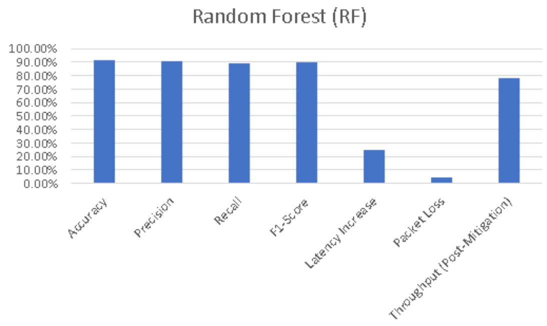


Figure 7: Performance graph of Random Forest

As we can see in the above Fig. 7 -

- The Random Forest (RF) detection system's performance across important evaluation measures is shown in the bar chart.
- Effective detection with steady post-mitigation throughput is ensured by RF's high accuracy, precision, recall, and F1-score, as well as its small latency increase and low packet loss.

3) KNN-Based Detection

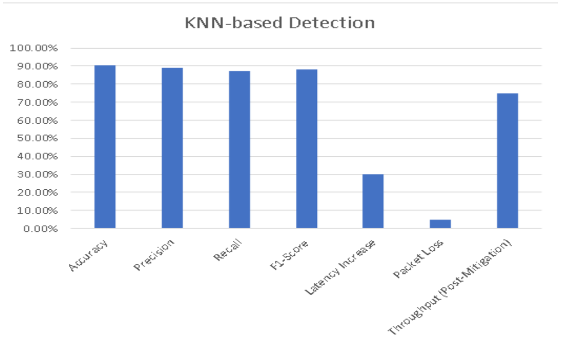


Figure 8: Performance graph of KNN-Based detection

In the above Fig. 8 -

- Using a variety of parameters, such as accuracy, precision, recall, F1-score, latency increase, packet loss, and post-mitigation throughput, the bar chart shows how well the KNN-based detection system performs.
- KNN has low packet loss and modest latency, yet achieving excellent accuracy and F1-score.

4) Proposed CNN-Based System

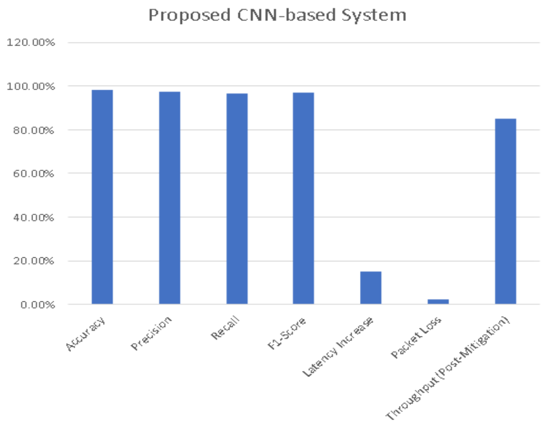


Figure 9: Performance graph of Proposed CNN-Based System

The above Fig. 9 shows -

- The performance of the suggested CNN-based detection system is shown in the bar chart using several important measures, such as accuracy, precision, recall, F1-score, latency increase, packet loss, and post-mitigation throughput.
- The CNN model demonstrates its effectiveness in detection and mitigation by achieving high accuracy and F1-score with minimal packet loss and lower latency.

5) Overall Comparison

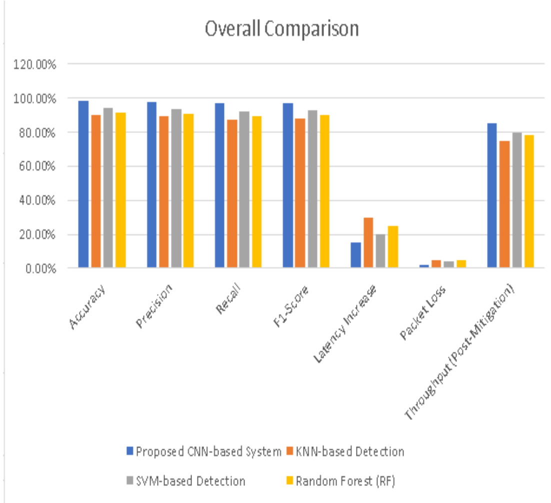


Figure 10: Performance of Ensemble Model

The Fig. 10 includes –

- Performance Comparison: The system designed using the CNN model surpasses other models (KNN, SVM, and Random Forest) with the best accuracy, precision, recall, and F1-score, suggesting its high detection ability.
- Network Efficiency: The CNN-based technique reduces latency increment, minimizes packet loss, and maximizes network throughput after mitigation, positioning it as the best botnet detection and mitigation solution for use in an SDN setting.

VI. CONCLUSION AND FURTHER ENHANCEMENT

The suggested botnet detection and mitigation framework in an SDN setup efficiently combines deep learning-based anomaly detection with graph theory-based traffic management to provide high accuracy with minimal service disruption. Experimental outcomes prove that the CNN-based model performs better than conventional methods like KNN, SVM, and Random Forest in terms of accuracy, precision, recall, and F1-score. The dynamic flow management of the system and ongoing model updates improve its resilience against changing botnet attacks. Future developments can emphasize incorporating federated learning for distributed training, using hybrid deep learning architectures such as CNN-LSTM to enhance detection accuracy, and using reinforcement learning for adaptive mitigation in real-time. Moreover, enhancing scalability for enterprise and cloud environments with large-scale systems and expanding security analysis to several layers of networks can enhance its robustness even further. These enhancements will render the framework a more inclusive and robust solution for protecting SDN environments against advanced botnet-based cyber threats.

REFERENCES

[1] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, "A new frame-work for DDoS attack detection and defense in SDN environment," IEEEAccess, vol. 8, pp. 161908–161919, 2020. https://ieeexplore.ieee.org/document/9175024

[2] S. Wang, J. F. Balarezo, K. G. Chavez, A. Al-Hourani, S. Kandeepan,M. R. Asghar, and G. Russello, "Detecting flooding DDoS attacks insoftware-defined networks using supervised learning techniques," Eng.Sci. Technol. Int. J., vol. 35, Nov. 2022, Art. no. 101176. https://www.sciencedirect.com/science/article/pii/S2215098622000842?via%3Dihub

[3] Y. Cui, Q. Qian, C. Guo, G. Shen, Y. Tian, H. Xing, and L. Yan, "TowardsDDoS detection mechanisms in software-defined networking," J. Netw.Comput. Appl., vol. 190, Sep. 2021, Art. no. 103156. https://www.sciencedirect.com/science/article/abs/pii/S1084804521001703?via%3Dihub

[4] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS attack detectionmethod based on SVM in software defined network," Secur. Commun.Netw., vol. 2018, pp. 1–8, Apr. 2018. https://doi.org/10.1155/2018/9804061

[5] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme usingdeep learning approach for Internet of Things," Future Gener. Comput.Syst., vol. 82, pp. 761–768, May 2018. https://doi.org/10.1016/j.future.2017.08.043

[6] J. A. Pérez-Díaz, I. A. Valdovinos, K. R. Choo, and D. Zhu, "A flexibleSDN-based architecture for identifying and mitigating low-rate DDoSattacks using machine learning," IEEE Access, vol. 8, pp. 155859–155872,2020. https://ieeexplore.ieee.org/document/9152693

[7] R. K. Chouhan, M. Atulkar, and N. K. Nagwani, "A framework to detectDDoS attack in Ryu controller based software defined networks usingfeature extraction and classification," Appl. Intell., pp. 1–21, 2022. https://doi.org/10.1007/s10489-022-04056-7

[8] Y. Liu, T. Zhi, M. Shen, L. Wang, Y. Li, and M. Wan, "Software-definedDDoS detection with information entropy analysis and optimized deeplearning," Future Gener. Comput. Syst., vol. 129, pp. 99–114, Apr. 2022. https://doi.org/10.1016/j.future.2021.11.017

[9] O. Habibi, M. Chemmakha, and M. Lazaar, "Imbalanced tabular datamodelization using CTGAN and machine learning to enhance IoT bot-net attacks detection," Eng. Appl. Artif. Intell., vol. 118, Feb. 2023,Art. no. 105669. https://doi.org/10.1016/j.engappai.2022.105669

[10] H. S. Ilango, M. Ma, and R. Su, "A FeedForward–Convolutional neuralnetwork to detect low-rate DoS in IoT," Eng. Appl. Artif. Intell., vol. 114,Sep. 2022, Art. no. 105059. https://doi.org/10.1016/j.engappai.2022.105059

[11] M. W. Nadeem, H. G. Goh, V. Ponnusamy, and Y. Aun, "DDoS detectionin SDN using machine learning techniques," Comput., Mater. Continua,vol. 71, no. 1, pp. 771–789, 2022, doi: 10.32604/cmc.2022.021669. https://doi.org/10.32604/cmc.2022.021669

[12] K. N. Rao, K. V. Rao, and P. V. G. D. P. Reddy, "A hybrid intrusiondetection system based on sparse autoencoder and deep neural network,"Comput. Commun., vol. 180, pp. 77–88, Dec. 2021. https://doi.org/10.1016/j.comcom.2021.09.025

[13] M. W. Nadeem, H. G. Goh, Y. Aun, and V. Ponnusamy, "A recurrent neuralnetwork based method for low-rate DDoS attack detection in SDN," inProc. 3rd Int. Conf. Artif. Intell. Data Sci. (AiDAS), Sep. 2022, pp. 13–18. https://ieeexplore.ieee.org/Xplore/home.jsp

[14] P. L. S. Jayalaxmi, G. Kumar, R. Saha, M. Conti, T.-H. Kim, andR. Thomas, "DeBot: A deep learning-based model for bot detection inindustrial Internet-of-Things," Comput. Electr. Eng., vol. 102, Sep. 2022,Art. no. 108214. https://doi.org/10.1016/j.compeleceng.2022.108214

[15] H.-T. Nguyen, Q.-D. Ngo, D.-H. Nguyen, and V.-H. Le, "PSI-rootedsubgraph: A novel feature for IoT botnet detection using classifier algo-rithms," ICT Exp., vol. 6, no. 2, pp. 128–138, Jun. 2020. https://doi.org/10.1016/j.icte.2020.02.004

[16] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent systembasedon one class support vector machine and grey wolf optimizationfor IoT botnet detection," J. Ambient Intell. Humanized Comput., vol. 11,no. 7, pp. 2809–2825, Jul. 2020. https://doi.org/10.1007/s12652-019-01444-w

[17] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai,D.Breitenbacher, and Y. Elovici, "N-BaIoT—Network-based detectionof IoT botnet attacks using deep autoencoders," IEEE Pervasive Comput.,vol. 17, no. 3, pp. 12–22, Jul. 2018. https://doi.org/10.1109/MPRV.2018.03367731

[18] M. Asadi, M. A. Jabraeil Jamali, S. Parsa, and V. Majidnezhad, "Detectingbotnet by using particle swarm optimization algorithm based on votingsystem," Future Gener. Comput. Syst., vol. 107, pp. 95–111, Jun. 2020. https://doi.org/10.1016/j.future.2020.01.041

[19] I. Idrissi, M. Boukabous, M. Azizi, O. Moussaoui, and H. El Fadili,"Toward a deep learning-based intrusion detection system for IoT againstbotnet attacks," IAES Int. J. Artif. Intell. (IJ-AI), vol. 10, no. 1, p. 110,Mar. 2021. http://doi.org/10.11591/ijai.v10.i1.pp110-120

[20] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj,"Anomaly-based intrusion detection system for IoT networks throughdeep learning model," Comput. Electr. Eng., vol. 99, Apr. 2022,Art. no. 107810. https://doi.org/10.1016/j.compeleceng.2022.107810