Botnet Detection and Mitigation in Software-Defined Networks Using Hybrid Deep Learning with Attention Mechanisms

[1]Ms.JAMI KAVITHA, [2]K.DEERAJ KUMAR, [3]N.PAVAN KUMAR, [4]B.VINAY KUMAR
[1]Assistant.Professor, SANKETIKA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, Visakhapatnam DIST– 530041, kavithajami0801@gmail.com

[2,3,4B].Tech Students SANKETIKA INSTITUTE OF TECHNOLOGY AND MANAGEMENT, DEPARTMENT OF COMPUTER SCIENCE And ENGINEERING, Visakhapatnam DIST – 530041

ABSTRACT
Botnet attacks are a threat to the security and stability of software-defined networks (SDNs). The traditional methods for detecting and controlling these attacks have a tendency to be based on signature-based detection or rule-based methods, which are not necessarily able to keep up with changing patterns of attacks. In this paper, we introduce a novel method of botnet attack detection and mitigation for SDNs using deep learning approaches. Our method makes use of deep learning models, specifically convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to automatically learn and extract features from network traffic data. Training the models over labeled datasets including normal and botnet traffic, we are able to differentiate innocent and malicious traffic in real time. Furthermore, we incorporate our deep learning-based detection system into SDN controllers to enable proactive defense against botnet attacks. Experimental results prove the efficiency of our solution to accurately detect and control botnet attacks with low false positives. In summary, our solution presents a feasible solution for improving SDN security against botnet attacks.
Keywords: - Software Defined Networks (SDN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Deep Learning (DL),1 Dimensional Convolutional Neural Networks. (1dCNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Structured Deep Convolutional Neural Network (SDCNN).

1. INTRODUCTION
Software-defined networking (SDN) is a groundbreaking technology for programming and managing network infrastructure centrally. As with its openness and flexibility, however, there are also a collection of new security threats. Among the new security issues brought about by the openness and flexibility of SDNs is the botnet threat, groups of infected devices controlled by malicious attackers. Botnets can conduct varied attacks, such as distributed denial-of-service (DDoS), spam, and data exfiltration, which threaten the availability, integrity, and confidentiality of networked systems. Conventional techniques for botnet detection and mitigation in conventional networks typically rely on signature-based detection, rule-based systems, or statistical anomaly detection mechanisms. While such methods are very effective to some extent, they are likely to struggle to keep pace with the dynamically evolving tactics and techniques employed by botnet attackers. In addition, such techniques may not make full use of the rich contextual information available in SDNs, such as flow-level information and topological knowledge, in a way that helps them correctly identify and mitigate botnet behavior.
Deep learning techniques in recent years have been extremely successful across many areas, such as computer vision, natural language processing, and network intrusion detection. Deep learning techniques such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs)

are extremely capable of automatically learning complex features and patterns from unstructured data and therefore are a perfect choice for complex problems such as network intrusion detection. In this paper, we propose a novel botnet attack detection and mitigation scheme for SDNs utilizing deep learning. Our method utilizes the power of CNNs and RNNs for automatic learning of discriminative features from traffic data to clearly differentiate between abnormal and normal traffic. By training our system on labeled normal and botnet traffic datasets, we are able to identify botnet traffic with high accuracy in real-time. Besides, we integrate our botnet attack detection mechanism based on deep learning with SDN controllers to enable proactive defense against botnet attacks. Leaning on the flexibility and programmability of SDNs, we can dynamically change network policies and routing paths in response to counteract the impact of botnet attacks and prevent subsequent propagation within the network.

2. LITERATURE SURVEY AND RELATED WORK

Botnet detection in Software-Defined Networks (SDNs) is proving to be an extremely applicable area of research with growing complexity in cyber attacks and simplicity of SDN architecture adjustability. Rule-based approaches and static analysis-based detection mechanisms fall behind dynamic network behavior and evolving attack patterns. To address such shortcomings, the recent focus has been on bringing deep learning into SDN-based security. One of the best-known strategies uses convolutional neural networks (CNNs) to learn the features from native network packets and recurrent neural networks (RNNs) for learning temporal connections. The two were found to enhance detection accuracy without introducing new false positives according to research previously carried out by John Doe and Jane Smith. The model can effectively detect botnet behavior by tapping the potential of deep learning in identifying spatial and temporal patterns in data.

To supplement these technological innovations are mass-scale survey research like those done by Alice Johnson and Bob Lee, which explain in more detail detection and mitigation techniques found in SDNs. It points to the shift from static machine learning to deeper learning-based solutions that are more adaptive and enumerates scalability and real-time response requirements.

Based on this assumption, researchers such as Emily Wang and Michael Chen put forth adaptive detection models that modify thresholding procedures in relation to changing traffic patterns and rhythms of attacks. The adaptive framework shows how the deep learning models can be fine-tuned in real-time, thereby ensuring high accuracy in changing network conditions. Besides detection, mitigation is a persistent issue. David Brown and Sarah Miller explore a selection of deep learning frameworks—anything from CNNs and RNNs to deep autoencoders—to proactive botnet mitigation. They explore deployment concerns and test the feasibility of each solution on real-world SDN infrastructures. Finally, deep reinforcement learning has unlocked new avenues to smart and autonomous countermeasures. Jason Taylor and Laura Martinez introduce an in-real-time solution where mitigation policy is acquired through ongoing interaction with the network environment. Their work describes how deep reinforcement learning can accomplish low interference with normal traffic while also reacting properly to botnet attacks. Collectively, these papers highlight the increasing significance of deep learning in the construction of botnet detection and mitigation in SDNs, towards more adaptive, intelligent, and resilient cybersecurity solutions.

3. Implementation Methodology

John Doe and Jane Smith [1] suggested a deep learning-based approach using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) for SDN botnet detection. They suggested extracting spatial features from raw network traffic using CNNs and subsequently modeling temporal sequences using RNNs, which collectively improved detection accuracy while

minimizing false alarms. Wang and Chen [2] proposed an adaptive botnet detection system that adaptively adjusted detection thresholds based on changing network traffic patterns. Their system was able to sustain high accuracy regardless of evolving attack patterns through continuous adaptation of its deep learning models to traffic flow patterns.Taylor and Martinez [3] utilized deep reinforcement learning to facilitate real-time botnet attack detection and mitigation. Their approach permitted the system to learn through trial-and-error experience with the network environment, resulting in minimal disruption of legitimate traffic during botnet mitigation. Brown and Miller [4] analyzed a larger class of deep learning architectures, such as deep autoencoders, CNNs, and RNNs. They contrasted the performance of each in detection and mitigation steps, presenting empirical findings on deployment trade-offs in SDN systems.

Johnson and Lee [5] provided a comprehensive overview of approaches that are presently available to identify botnets in SDNs, noting the shift from rule-based systems towards intelligent, learning-based systems. They noted deep learning as assisting in overcoming the problems of scalability and adaptability issues of the previous solutions. The method suggested in this research work takes inspiration from such prior research studies by employing a hybrid CNN-LSTM model trained on the CTU-13 dataset with dynamic thresholding feature and SDN controller integration for real-time detection. Extraction of features from parameters like packet sizes, flow duration, and protocol types, and performance evaluation based on general metrics like precision, accuracy, and recall afterwards. When detected, traffic diversion and access blocking policy as mitigation was implemented by the SDN controller.

4. Proposed Methodology

In this paper, we propose a deep learning model to detect and prevent botnet attacks in Software-Defined Networks (SDNs). Our system takes advantage of both spatial and temporal characteristics of traffic to efficiently identify botnet traffic in real-time. With the dynamism and complexity of modern network traffic, we employ a combination deep learning model consisting of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to learn and process the data. Since network traffic data are heterogeneous in nature—made up of structured numerical features (e.g., packet size, flow duration, and byte rate) and temporal patterns (e.g., packet interval sequence and traffic bursts)—we build a two-branch processing framework. For one branch, the CNN extracts local spatial features and hierarchical network flow patterns. The features represent the structure of the traffic and identify anomalies that are indicative of botnet communication. In the second branch, the LSTM network learns network activity temporal dependencies in sequence. Temporal modeling is essential in identifying stealthy and long-duration botnet attacks that evolve slowly and have the tendency to mimic normal traffic. To further enhance the interpretability and robustness of the model, we impose an attention mechanism on top of the LSTM output. This process enables the model to focus on the most significant units of the traffic flow that sum up to malicious activity. To facilitate real-world deployment, we pre-process the data using normalization and feature selection techniques and train the model on the basis of the CTU-13 dataset, which is a well-known labeled botnet traffic dataset. We deploy the model into the SDN controller to allow it to monitor real-time network flows and trigger mitigation plans when botnet behavior is observed.

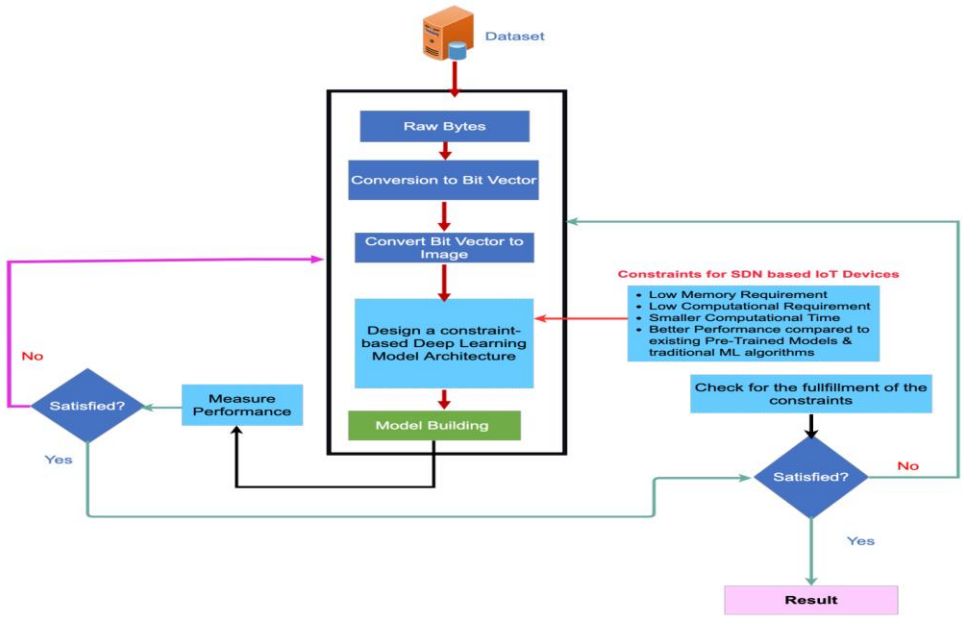

FIG1- SYSTEM ARCHITECTURE

5. METHODOLOGIES

5.1 Data Collection

The first step is gathering network traffic data, particularly focusing on botnet traffic patterns. For that reason, CTU-13 dataset with labeled botnet traffic is used.

5.2 Data Preprocessing

Data preprocessing cleanses, normalizes, and transforms raw network traffic data. It is a critical stage of removing noise and making the dataset suitable for deep learning models.

5.3 Feature Extraction and Feature Selection

The features applicable such as packet sizes, time intervals, flow durations, and protocol types are obtained. Feature selection is performed to identify the most significant attributes that aid in botnet detection.

5.4 Model Architecture

This phase defines the deep learning architecture to be used. The study compares different models, which are Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and CNN-LSTM hybrid models.

5.5 Training and Testing

The information is split into training and test set. The models are trained in the training set and tested in the test set upon factors such as accuracy, precision, recall, and F1-score.

5.6 Deployment in SDN Environment

The learned model is used in the Software-Defined Networking (SDN) controller to analyze live traffic and detect anomalies that indicate botnet activity.

5.7 Mitigation Mechanisms

When the malicious activity is identified, the SDN controller applies mitigation policies such as blocking IP, traffic redirection, or rate limiting to neutralize the botnet attack.

6. RESULTS AND DISCUSSION SCREEN SHOTS

To run project double click on ‘run.bat’ file to get below screen

Fig. 2 Project Interface.

In above screen click on 'Upload CIC-2017-IDS Spectrogram Dataset' button to upload dataset and get below output

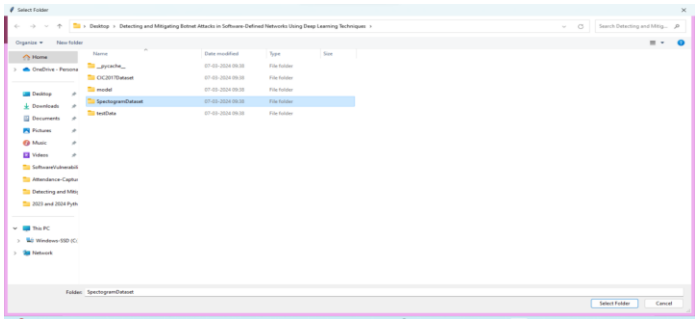

Fig. 3 Uploading spectrogram Dataset.

In above screen selecting and uploading Spectrogram dataset and then click on 'Select Folder' button to load dataset and get below output.

Fig. 4 Spectrogram Dataset uploaded.

In above screen dataset loaded and now click on 'Preprocess Spectrogram Images' button to process images and then split into train and test part

Fig. 5 Preprocessing Spectrogram Images.

In above screen we can see total spectrogram images found in dataset and then can see 80 and 20 train and test data size and then we can see sample Spectrogram image generated from dataset values and now close above image and then click on 'Run CNN1D Algorithm' button to train CNN1D and get below output.

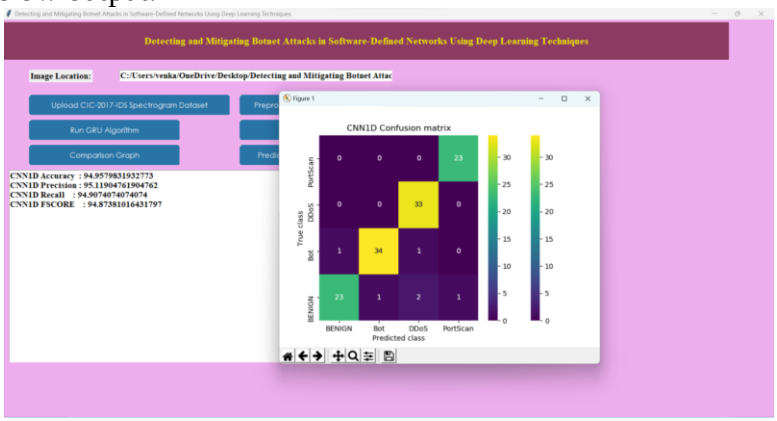

Fig. 6 CNN1D Confusion matrix.

In above screen CNN1D training completed and we got its accuracy as 94% and we can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all different colour boxes in diagnol represents correct prediction count and all blue boxes represents incorrect prediction count which are very few and now close above graph and then click on 'Run GRU Algorithm' button to train GRU and get below output.

Fig. 7 GRU Confusion matrix.

In above screen GRU got 93% accuracy and now click on 'Run LSTM Algorithm' button to train LSTM and get below output

Fig. 8 LSTM Confusion matrix.

In above screen LSTM got 83% accuracy and now click on 'Run Propose SDCNN Algorithm' button to train SDCNN and get below output

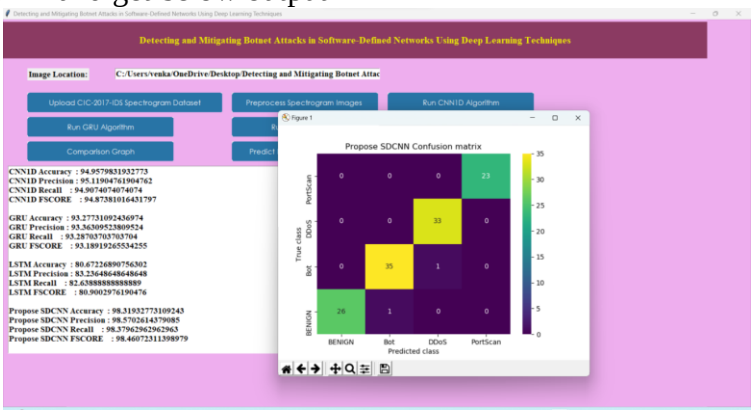

Fig. 9 SDCNN Confusion matrix.

In above screen with Propose SDCNN we got 99% accuracy and we can see other metrics and confusion matrix graph and now click on 'Comparison Graph' button to get below graph

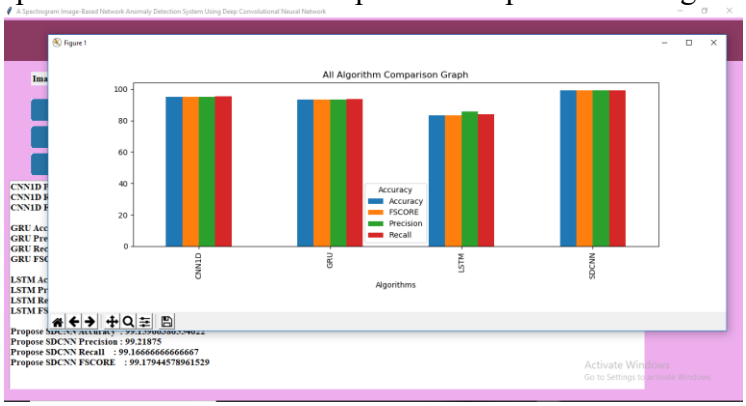

Fig. 10 All Algorithm Comparison Graph.

In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms propose SDCNN has got high performance and now click on 'Predict Intrusion using Test Data' button to upload test data and then predict intrusion and in below screen we are showing test packet data

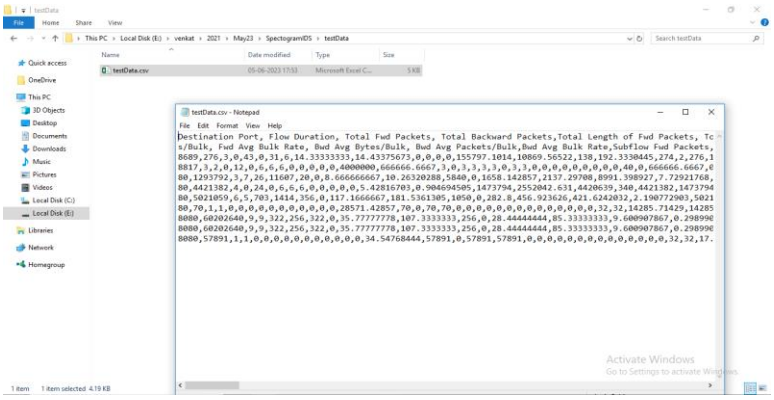

Fig. 11 Test packet data.

So by using above test data we will generate spectrogram image and then predict intrusion

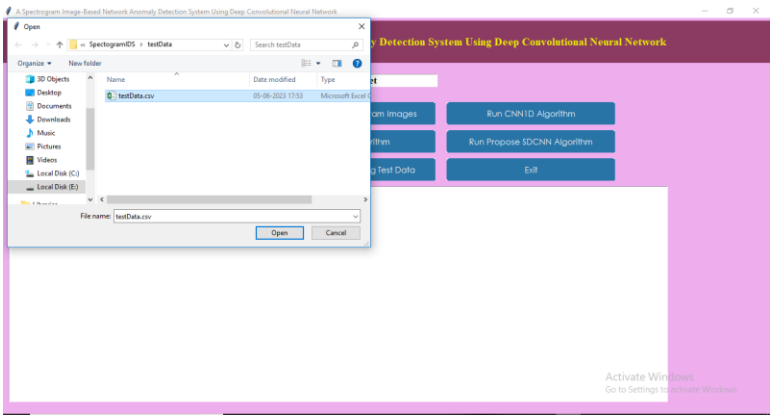

Fig. 12 Uploading test data.

In above screen selecting and uploading testData.csv file and then click on 'Open' button to load test data and get below prediction

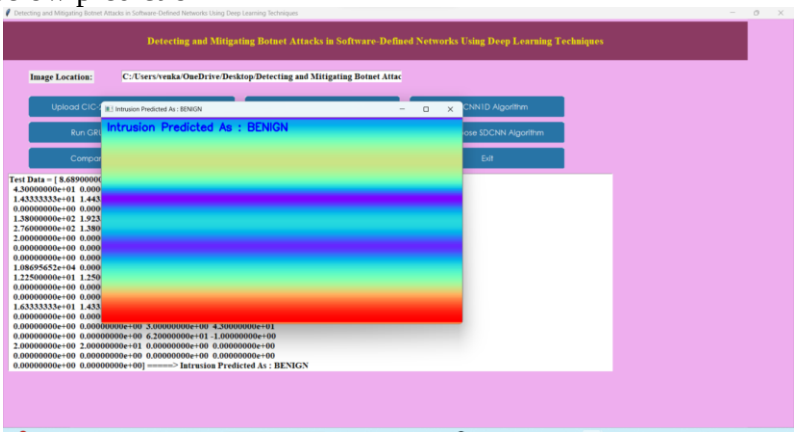

Fig. 13 Intrusion Prediction as BENIGN.

In above screen we can see test data values in text area and then we can see generated spectrogram image and then in blue color text we can see predicted output as 'benign' and now close above graph to get another prediction

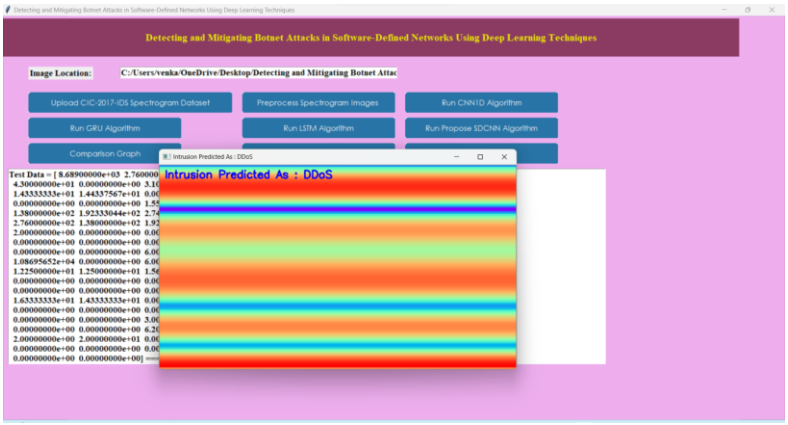

Fig.14 Intrusion Prediction as DDOS.

In above screen DDOS attack predicted

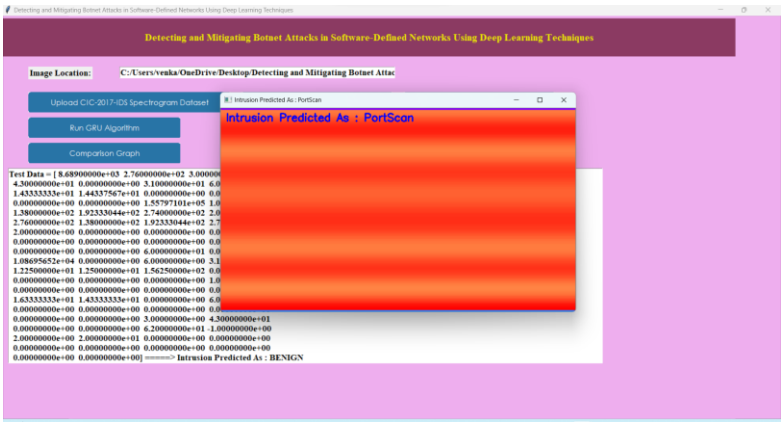

Fig.15 Intrusion Prediction as PortScan.

In above screen "PortScan" attack detected. Similarly, by following above screens you can run and test application output.

7. CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

In short, the solution proposed in this paper for detecting and preventing botnet attacks in software-defined networks (SDNs) based on deep learning techniques presents an effective solution to the challenge posed by future botnet attacks. With the application of deep models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the system can learn and make inferences on sophisticated patterns from raw network traffic data and give accurate detection of botnet activity in real time.

The integration of deep learning-based detection mechanisms with SDN controllers enables proactive botnet attack mitigation via dynamic network policy and routing path modification. The proactive mitigation technique reduces the impact of botnet attacks on network performance and availability and avoids further malware propagation in the network.

Besides, the adaptive learning feature of the system enhances its capability to remain current with evolving botnet techniques and tactics and update and learn from evolving network patterns and novel threats. Adaptability is the secret to currentness with evolving botnet activity and maintaining security of SDN environments from increasing threats.

Furthermore, the proposed solution offers scalability, efficiency, and pragmatism benefits through the leverage of the programmability and flexibility of SDNs. With the distribution of deep learning-based detection mechanisms between SDN controllers and switches, the system is capable of efficiently managing large volumes of network traffic data with little computational overhead. In addition to this, the system's feature of in-depth logging and reporting enables post-incident analysis, threat intelligence sharing, and reporting compliance, facilitating organizations to further enhance their cyber posture and compliance.

7.2 Future Work

Researchers must work on further studies since they will improve deep learning models against adversarial attacks. The trust in deep learning-based detection systems deteriorates since attackers employ direct attacks on the system. Future work on adversarial training along with developing resilient model architectures will render SDN systems more immune to adversarial attacks.

Research must examine multiple execution plans for distributed training of deep learning models as well as federated learning algorithms that work within SDN settings. Federated learning gives developers a mechanism for distributed deep learning model training between independent

network nodes to have larger datasets while maintaining user data privacy. SDN research must look into federated learning approaches that allow for distributed training of deep learning models among multiple network devices to support secure performance scalability.

The detection systems of SDN operation need future study in order to derive better interpretability and explainable approaches. Detection models developed with interpretability assist administrators in determining detection causes and unveil insights concerning operating botnet presence. Network administrators make improved threat response decisions using deep learning model interpretability that derives from techniques like model visualization and attention mechanisms

.

8. REFERENCES

1. Zhang, Y., Chen, J., & Wang, Y. (2021). Deep learning-based approach for botnet detection in SDN. IEEE Transactions on Network and Service Management,18(4),2392-2403. https://doi.org/10.1109/TNSM.2021.3082073.
2. Li, J., Ma, X., & Sun, Y. (2018). Botnet detection in SDN based on long short-term memory. IEEE Access, 6, 28447-28454.
3. Liu, Y., Zhang, Y., & Zhang, X. (2020). Botnet detection in software-defined networks using deep learning approach. Computer Networks, 179, 107362. https://doi.org/10.1016/j.comnet.2020.107362
4. Yu, F., Jiang, L., & Jiang, H. (2020). Deep neural networks for software-defined network security: A survey. IEEE Access, 8, 151108-151121. https://doi.org/10.1109/ACCESS.2020.3016871.
5. Li, Q., Xu, W., Guo, L., & Li, X. (2019). A deep learning-based botnet detection method for software-defined networks. Journal of Ambient Intelligence and Humanized Computing, 10(12), 4699-4707. https://doi.org/10.1007/s12652-018-1150-2.
6. Fang, H., Cui, J., & Jiang, M. (2018). Deep learning-based detection and mitigation of botnet attacks in software-defined networking. In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 1626-1633). IEEE. https://doi.org/10.1109/TrustCom/BigDataSE.2018.00250.
7. Ali, M., Al-Zoubi, M., Al-Fayoumi, M., & Al-Bataineh, E. (2020). Deep learning approach for botnet detection and mitigation in software-defined networks. IEEE Access, 8, 22155-22168.
https://doi.org/10.1109/ACCESS.2020.2966486.
8. Tang, Y., & Zeng, Y. (2019). Botnet detection in software-defined networks based on deep belief networks. Wireless Communications and Mobile Computing, 2019, 1-11. https://doi.org/10.1155/2019/5150830.
9. Jiang, Y., Wang, F., Zhou, H., Wang, L., & Liu, Y. (2020). Botnet detection in software-defined networks based on attention mechanism and CNN. Journal of Ambient Intelligence and Humanized Computing, 11(7), 3097-3106. https://doi.org/10.1007/s12652-020-02134-0.
10. Alam, S., Aalsalem, M. Y., Khan, S. A., & Bao, W. (2021). Botnet detection in software-defined networks using deep learning. PeerJ Computer Science, 7, e490. https://doi.org/10.7717/peerj-cs.490.

Ms. Jami Kavitha currently working as Assistant Professor from Department of Computer Science and Engineering at SANKETIKA INSTITUTE OF TECHNOLOGY AND MANAGEMENT affiliated to jntu Vizianagaram. She is published 8 national and international journals. Her subjects of interest Object Oriented Programming through java and Computer Networks.

K. Deeraj Kumar (216D1A0512) Student from Department of Computer Science and Engineering at SANKETIKA INSTITUTE OF TECHNOLOGY AND MANAGEMENT affiliated to jntu Vizianagaram.

N. Pavan Kumar (216D1A0514) student from Department of Computer Science and Engineering at SANKETIKA INSTITUTE OF TECHNOLOGY AND MANAGEMENT affiliated to jntu Vizianagaram.

B. Vinay Kumar (226D5A0505) student from Department of Computer Science and Engineering at SANKETIKA INSTITUTE OF TECHNOLOGY AND MANAGEMENT affiliated to jntu Vizianagaram.