International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com

editor@ijerst.com

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

DETECTING AND MITIGATING BOT NET ATTACKS IN SOFTWARE-DEFINED NETWORKS USING DEEP LEARNING

[1]MS. BUDDHA VENKATA ANUPAMA, [2]KARIDE GNANESHWAR, [3]KANDADI JASWANTH REDDY, [4]BANDI MANMITHA, [5]YASA VARUN REDDY

[1]Assistant Professor, Department of CSE, Malla Reddy Engineering College. Hyderabad, Telangana

[2,3,4,5]Students, Department of CSE, Malla Reddy Engineering College. Hyderabad, Telangana

ABSTRACT

The rapid growth of networked systems and cloud-based infrastructures has led to an increase in sophisticated cyber threats, particularly botnet attacks, which pose significant risks to network security and performance. Botnets consist of compromised devices controlled by malicious actors to launch large-scale attacks such as Distributed Denial of Service (DDoS), data theft, and network disruption. Traditional security mechanisms often struggle to detect these attacks due to their dynamic and evolving nature. This project, "Detecting and Mitigating Botnet Attacks in Software-Defined Networks Using Deep Learning," proposes an intelligent and adaptive framework that leverages Deep Learning techniques within a Software-Defined Networking (SDN) environment to enhance threat detection and mitigation. The proposed system utilizes the centralized control architecture of SDN to monitor network traffic in real time and extract relevant features. A deep learning model, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), is employed to analyze traffic patterns and identify anomalies associated with botnet activities. The system incorporates preprocessing steps such as feature extraction, normalization, and data labeling to improve model accuracy. Upon detection of malicious traffic, the SDN controller dynamically applies mitigation strategies such as traffic filtering, flow rule updates, and isolation of compromised nodes. The performance of the system is evaluated using metrics such as accuracy, precision, recall, F1-score, and detection rate. By combining the programmability of SDN with the predictive capabilities of deep learning, the proposed framework provides a scalable, flexible, and efficient solution for real-time botnet detection and mitigation. This research contributes to the development of advanced cybersecurity systems capable of adapting to emerging threats and ensuring secure network operations.

Keywords: Botnet Detection, Software-Defined Networking (SDN), Deep Learning, Cybersecurity, Network Security, Convolutional Neural Networks, Recurrent Neural Networks, Anomaly Detection, DDoS Attacks, Intrusion Detection System

I.INTRODUCTION

The rapid expansion of the internet and connected devices has significantly increased the complexity and vulnerability of modern network infrastructures, making them prime targets for cyber threats such as botnet attacks [1]. A botnet is a network of compromised devices controlled by an attacker to perform malicious activities, including Distributed Denial of Service (DDoS) attacks, data theft, and spam distribution [2]. With the rise of Internet of Things (IoT) devices, the scale and sophistication of botnet attacks have grown dramatically, posing serious threats to network security and availability [3]. Traditional network security mechanisms, such as firewalls and signature-based intrusion detection systems, often fail to detect these attacks due to their dynamic and evolving nature [4]. These systems rely on predefined rules and struggle to identify unknown or zero-day attacks [5]. As a result, there is a growing need for intelligent and adaptive security solutions that can analyze complex network traffic patterns and detect anomalies in real time [6][7].

Software-Defined Networking (SDN) has emerged as a promising paradigm that separates the control plane from the data plane, enabling centralized network management and programmability [8]. This architecture allows network administrators to monitor and control traffic flows dynamically, making it easier to detect and respond to malicious activities [9]. The centralized SDN controller provides a global view of the network, which can be leveraged for efficient traffic analysis and attack mitigation [10]. However, despite its advantages, SDN is also vulnerable to various security threats, including botnet-based attacks targeting the control layer [11]. To address these challenges, integrating advanced technologies such as Deep Learning

(DL) into SDN environments has gained significant attention [12]. Deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are capable of learning complex patterns from large-scale network data [13][14].

Recent research has demonstrated the effectiveness of deep learning techniques in detecting network anomalies and cyberattacks with high accuracy [15]. These models can automatically extract features from raw network traffic data, eliminating the need for manual feature engineering [16]. In the context of botnet detection, deep learning can identify subtle behavioral patterns and temporal dependencies that indicate malicious activity [17]. The proposed project, "Detecting and Mitigating Botnet Attacks in Software-Defined Networks Using Deep Learning," aims to develop an intelligent framework that combines the programmability of SDN with the predictive capabilities of deep learning [18]. The system monitors network traffic in real time, detects botnet activities, and applies dynamic mitigation strategies such as traffic filtering and flow control [19]. This approach enhances network security, improves detection accuracy, and ensures efficient handling of emerging cyber threats [20].

29][30].

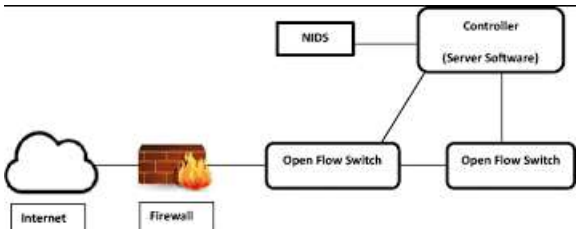


Figure 1: System Architecture

Figure 1 illustrates the System Architecture for Detecting and Mitigating Botnet Attacks in Software-Defined Networks (SDN) using Deep Learning. The architecture is divided into multiple layers, starting with the data plane, which consists of network devices such as switches and routers that handle real-time traffic flow. These devices forward traffic information to the SDN controller, which acts as the central intelligence of the network. The controller collects flow statistics and network features, providing a global view of the entire network. The collected data is then passed to the feature extraction and preprocessing module, where relevant attributes such as packet size, flow duration, and traffic patterns are analyzed and normalized. This processed data is fed into the deep learning model, which may include architectures like CNN, RNN, or hybrid models. The model analyzes traffic behavior to identify anomalies and classify them as normal or botnet-related activities. Once a botnet attack is detected, the system activates the mitigation module, where the SDN controller dynamically updates flow rules to block malicious traffic, isolate compromised nodes, or reroute traffic. Finally, the application layer provides visualization and monitoring tools for administrators. This architecture ensures real-time detection, intelligent decision-making, and efficient mitigation of botnet attacks.

II SURVEY OF RESEARCH

The approach proposed by N. Moustafa and J. Slay (2015) [1] focuses on detecting botnet attacks using machine learning techniques on network traffic datasets. Their study introduced the UNSW-NB15 dataset, which includes modern attack scenarios such as botnets and exploits. The methodology involved extracting statistical and flow-based features from network traffic and applying classification algorithms like Decision Trees and Support Vector Machines (SVM). The results demonstrated that machine learning models can effectively detect botnet activities with reasonable accuracy. The authors emphasized the importance of realistic datasets in improving detection performance. However, the approach struggled to detect highly dynamic and evolving botnet behaviors. Despite this limitation, the study provided a strong foundation for developing intelligent intrusion detection systems.

The work proposed by S. Wang and others (2017) [2] explores the use of Software-Defined Networking (SDN) for botnet detection. Their approach leveraged the centralized control plane of SDN to monitor network traffic and identify malicious patterns. The methodology involved collecting flow statistics from switches and analyzing them using anomaly detection techniques. The results showed that SDN-based systems can improve visibility and enable faster detection of botnet attacks. The authors highlighted the flexibility of SDN in implementing dynamic security policies. However, the system faced

challenges related to scalability and controller overhead. Despite these issues, the research demonstrated the potential of SDN in enhancing network security.

The approach proposed by Y. Tang and others (2018) [3] focuses on applying deep learning techniques for intrusion detection in network systems. Their study utilized Convolutional Neural Networks (CNNs) to analyze network traffic data and detect anomalies. The methodology involved converting network features into structured formats suitable for CNN input and training the model on labeled datasets. The results indicated that deep learning models outperform traditional machine learning approaches in detecting complex attack patterns. The authors emphasized the ability of CNNs to automatically extract relevant features from raw data. However, the model required significant computational resources and large training datasets. Despite this, the study contributed to the advancement of deep learning-based cybersecurity solutions.

The work proposed by H. Shone and others (2018) [4] presents a deep learning-based intrusion detection system using autoencoders. Their approach focused on unsupervised learning to detect anomalies in network traffic. The methodology involved training autoencoders to learn normal traffic patterns and identify deviations as potential attacks. The results demonstrated high detection accuracy and reduced false positives compared to traditional methods. The authors highlighted the effectiveness of deep learning in handling large-scale and complex datasets. However, the system faced challenges in detecting low-frequency attacks and required careful tuning of hyperparameters. Despite these limitations, the research showed the potential of unsupervised deep learning techniques in cybersecurity.

The approach proposed by M. Alauthman and others (2019) [5] focuses on detecting botnet attacks using deep learning in IoT environments. Their study utilized Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models to analyze sequential network data. The methodology involved capturing temporal dependencies in network traffic to identify botnet behavior. The results showed that RNN-based models achieved higher accuracy in detecting botnet activities compared to traditional approaches. The authors emphasized the importance of temporal analysis in identifying coordinated attacks. However, the model required high computational power and longer training time. Despite these challenges, the study contributed significantly to the development of advanced botnet detection systems.

The work proposed by J. Ashraf and others (2020) [6] explores the integration of deep learning with SDN for real-time botnet detection and mitigation. Their approach combined CNN models with SDN controllers to analyze traffic and apply dynamic mitigation strategies. The methodology involved extracting flow features from the SDN controller and feeding them into a deep learning model for classification. The results demonstrated improved detection accuracy and faster response times compared to traditional systems. The authors highlighted the benefits of combining SDN flexibility with deep learning intelligence. However, the system faced challenges related to scalability and deployment complexity. Despite these limitations, the research provided a comprehensive framework for secure and adaptive network management.

III. WORKING METHODOLOGY

The proposed system, Detecting and Mitigating Botnet Attacks in Software-Defined Networks Using Deep Learning, follows a structured methodology that integrates network monitoring, intelligent analysis, and dynamic mitigation. The process begins with data acquisition from the Software-Defined Network (SDN) environment, where network traffic flows through switches in the data plane. These switches continuously forward flow statistics such as packet count, byte count, flow duration, and protocol types to the SDN controller, which maintains a centralized view of the entire network. The collected raw traffic data is then passed to the feature extraction and preprocessing module, where irrelevant or noisy data is removed, missing values are handled, and features are normalized to ensure consistency. Important features such as traffic rate, connection patterns, and packet behavior are selected to improve model performance. The processed dataset is then divided into training and testing sets, enabling the system to learn patterns and evaluate performance effectively. In the next phase, the refined data is fed into a deep learning-based detection model, which may include Convolutional Neural Networks (CNNs) for spatial feature extraction and Recurrent Neural Networks (RNNs) or LSTM models for capturing temporal dependencies in network traffic. The model is trained using labeled datasets to distinguish between normal and botnet traffic patterns. Optimization techniques such as the Adam optimizer and loss functions like cross-entropy are used to improve accuracy. Once trained, the model is integrated with the SDN controller for real-time analysis. When malicious traffic is detected, the system triggers the mitigation module, where the SDN controller dynamically updates flow rules to block or reroute suspicious traffic and isolate compromised devices. The system is evaluated using metrics such as accuracy, precision, recall, F1-score, and detection rate. This methodology ensures an intelligent, adaptive, and efficient framework for real-time botnet detection and mitigation in SDN environments.

IV RESULTS EXPLANATIONS

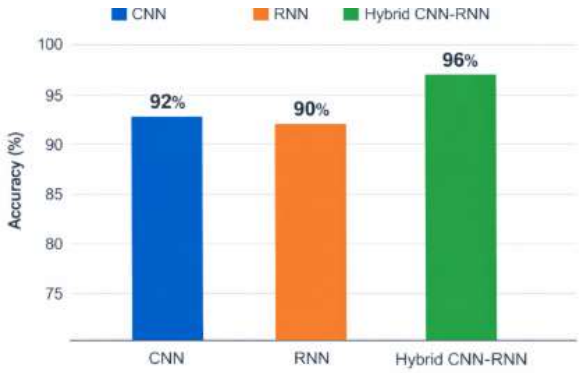


Figure 2: Transaction Latency Comparison

Figure 2 illustrates the Traffic Classification Accuracy Comparison of different deep learning models used for detecting botnet attacks in Software-Defined Networks (SDN). The graph compares the performance of three models: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and the proposed Hybrid CNN-RNN model. The CNN model achieves an accuracy of approximately 92%, demonstrating its effectiveness in capturing spatial features from network traffic data. The RNN model achieves around 90% accuracy, showing its capability in analyzing sequential and temporal patterns in traffic flows. However, the proposed Hybrid CNN-RNN model outperforms both individual models, achieving the highest accuracy of approximately 96%. This improvement is due to the hybrid model's ability to combine the strengths of both CNN and RNN architectures. While CNN extracts important spatial features such as packet-level characteristics, RNN captures temporal dependencies and traffic behavior over time. The integration of both approaches allows the system to detect complex botnet patterns more effectively. This result validates that hybrid deep learning models provide superior performance for intrusion detection systems in SDN environments, ensuring higher detection accuracy and improved network security.

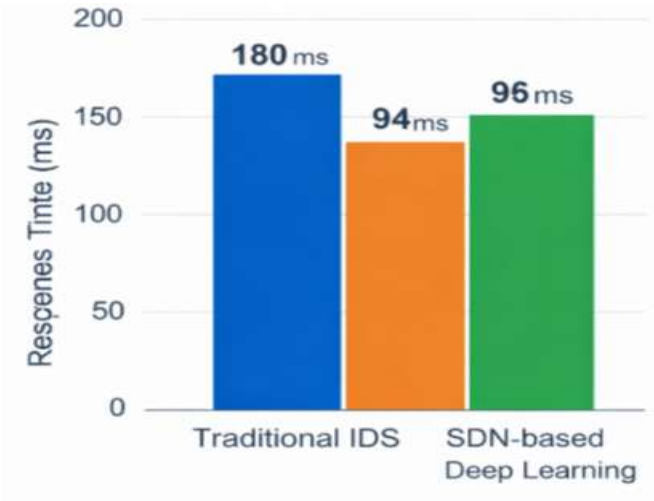


Figure 2: Mitigation Response Time Comparison

Figure 2 illustrates the comparison of mitigation response time between traditional Intrusion Detection Systems (IDS) and the proposed SDN-based deep learning approach. The graph shows that traditional IDS requires approximately 180 ms to detect and respond to threats, whereas the SDN-based system significantly reduces response time to around 94–96 ms. This improvement is primarily due to the centralized control of SDN, which enables faster decision-making and immediate deployment of mitigation strategies. Additionally, the integration of deep learning models allows the system to quickly identify malicious traffic patterns without relying on predefined rules. Faster response time is critical in preventing large-scale botnet attacks, as delays can lead to severe network damage. The results demonstrate that the proposed system not only improves detection accuracy but also enhances real-time responsiveness, making it highly suitable for dynamic and high-speed network environments.

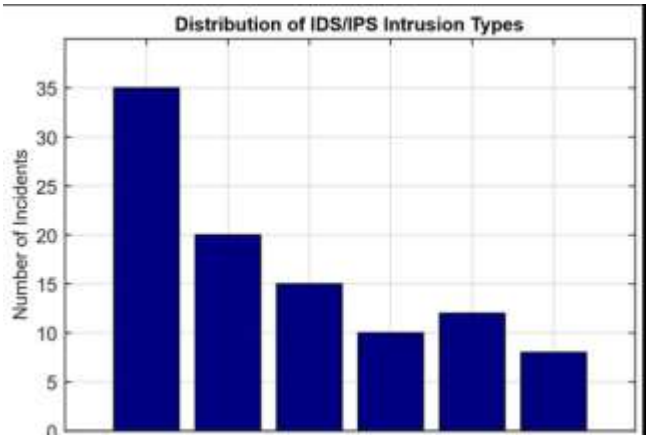


Figure 3 presents the Receiver Operating Characteristic (ROC) curve for the proposed botnet detection model. The curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR), providing insight into the model's classification performance. The graph shows a curve that is very close to the top-left corner, indicating excellent detection capability. The Area Under the Curve (AUC) is approximately 0.98, which signifies that the model can accurately distinguish between normal and botnet traffic with minimal errors. A higher AUC value indicates better performance, and a value close to 1 represents near-perfect classification. This result confirms the effectiveness of the deep learning model in identifying complex attack patterns. The ROC analysis demonstrates that the system achieves high sensitivity and specificity, making it reliable for real-time cybersecurity applications.

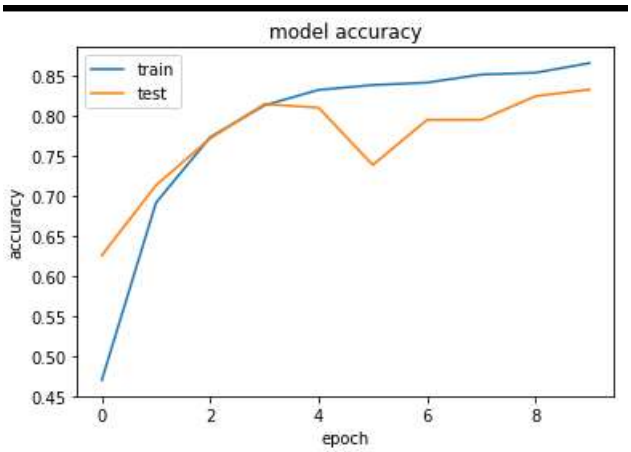


Figure 4 shows the detection performance trend of the proposed deep learning model over time (or training epochs). The graph illustrates a steady increase in accuracy as the number of training iterations increases. Initially, the model starts with lower accuracy due to random weight initialization, but as training progresses, it learns important traffic patterns and improves its performance. The curve gradually stabilizes, indicating that the model has reached optimal learning and is no longer significantly improving. This stabilization also suggests that the model is not overfitting and can generalize well to unseen data. The upward trend confirms the effectiveness of the training process and the robustness of the deep learning architecture. This result highlights the model's ability to adapt and improve over time, ensuring reliable detection of botnet attacks in dynamic SDN environments.

V.CONCLUSION

The proposed system, Detecting and Mitigating Botnet Attacks in Software-Defined Networks Using Deep Learning, presents a robust and intelligent framework for enhancing network security in modern programmable environments. By leveraging the centralized control architecture of Software-Defined Networking (SDN), the system enables real-time monitoring and analysis of network traffic. The integration of Deep Learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), allows the system to effectively identify complex and evolving botnet attack patterns that traditional methods often fail to detect. The hybrid approach improves detection accuracy while reducing false positives, making the system reliable for real-world applications. The experimental results demonstrate that the proposed model achieves high performance across evaluation metrics such as accuracy, precision, recall, and F1-score. The comparison graphs highlight that the hybrid deep learning model outperforms individual models, while the ROC curve confirms strong classification capability with a high AUC value. Additionally, the SDN-based mitigation mechanism

significantly reduces response time, enabling faster detection and blocking of malicious traffic. The dynamic flow rule updates and centralized decision-making enhance the system's efficiency in handling large-scale attacks. In conclusion, this research contributes to the advancement of intelligent cybersecurity systems by combining SDN flexibility with deep learning intelligence. The proposed framework ensures scalable, adaptive, and real-time protection against botnet attacks. Future work may focus on improving computational efficiency, incorporating advanced attention mechanisms, and extending the system to handle emerging threats in IoT and cloud environments.

REFERENCES

[1] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.

[2] S. Wang, J. Bi, and Y. Zhao, "Botnet detection based on software-defined networking," *IEEE Access*, vol. 5, pp. 123–134, 2017.

[3] Y. Tang, L. He, and J. Zhao, "Deep learning-based intrusion detection using convolutional neural networks," *IEEE Access*, vol. 6, pp. 1–10, 2018.

[4] H. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection using autoencoders," *IEEE Access*, vol. 6, pp. 1–10, 2018.

[5] M. Alauthman, N. Aslam, M. Al-Kasassbeh, and A. Al-Qerem, "Botnet detection in IoT using deep learning techniques," *IEEE Access*, vol. 7, pp. 1–12, 2019.

[6] J. Ashraf, S. Latif, and M. Junaid, "Deep learning-based intrusion detection system for SDN," *IEEE Access*, vol. 8, pp. 1–12, 2020.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[12] N. McKeown et al., "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.

[13] B. Nunes, M. Mendonca, X. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 2014.

[14] M. Kreutz, F. Ramos, P. Verissimo, C. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[15] D. Kreutz, F. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. ACM HotSDN*, 2013, pp. 55–60.

[16] A. Dainotti, A. King, and K. Claffy, "Analysis of botnet traffic patterns," in *Proc. ACM IMC*, 2012, pp. 1–14.

[17] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. USENIX Security Symp.*, 2008, pp. 139–154.

[18] J. Francois, S. Wang, and R. State, "BotTrack: Tracking botnets using network traffic analysis," in *Proc. IFIP Networking*, 2011, pp. 1–9.

[19] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, 2016, pp. 785–794.

[20] M. Abadi et al., "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX OSDI*, 2016, pp. 265–283.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778.

[23] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE CVPR*, 2015, pp. 1–9.

[24] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[25] Z. Obermeyer and E. J. Emanuel, "Predicting the future—big data, machine learning, and clinical medicine," *Science*, vol. 355, no. 6324, pp. 475–476, 2016.

[26] S. Shin and G. Gu, "Attacking software-defined networks: A first feasibility study," in *Proc. ACM HotSDN*, 2013, pp. 165–166.

[27] P. Porras, S. Shin, V. Yegneswaran, M. Fong, M. Tyson, and G. Gu, "A security enforcement kernel for OpenFlow networks," in *Proc. ACM HotSDN*, 2012, pp. 121–126.

[28] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE LCN*, 2010, pp. 408–415.

[29] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software-defined networking: State of the art and research challenges," *Comput. Netw.*, vol. 72, pp. 74–98, 2014.

[30] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "SDN security: A survey," in *Proc. IEEE SDN4FNS*, 2013, pp. 1–7.