International Journal of Multidiscipline

TLEP – International Journal of Multidiscipline
(Technology, Language, Education, and Psychology)
ISSN: 2488-9342 (Print) | 2488-9334 (Online)

Open Access | Peer-Reviewed | Monthly Publication | Impact factor: 8.497 / 2025

A Review of Machine Learning Approaches for DDoS Attack Detection and Mitigation in Software-Defined Networking

Azizjon Meliboev
Digital technology and Mathematics, Kokand University

Abstract
Software-defined networking (SDN) has changed the way networks are designed and controlled by separating the control plane from the data plane. This separation gives administrators a more programmable, centralized, and flexible way to manage traffic. At the same time, the SDN controller becomes a critical point of exposure. Distributed denial-of-service (DDoS) attacks can overload the controller, switches, links, or servers and can prevent legitimate users from accessing network services. The reviewed paper examines DDoS threats in SDN by discussing attack categories, public datasets, detection methods, and mitigation frameworks. It emphasizes the importance of open datasets for reproducible experiments and highlights that many available datasets are either outdated, not SDN-specific, or limited in attack diversity. The study also shows that machine learning and deep learning models have become central tools for detection, although their real-world usefulness depends on dataset quality, controller overhead, scalability, and false-positive control.
Keywords: software-defined networking; distributed denial-of-service; DDoS detection; SDN security; machine learning; open datasets; cyberattack mitigation.

1. Introduction

In conventional distributed networks, devices such as routers and switches normally combine forwarding responsibilities with control decisions. Network policies, routing behavior, quality of service settings, and security rules are commonly configured device by device. This approach has supported many forms of networked computing, including client-server systems, peer-to-peer communication, and distributed storage. However, it can become rigid and difficult to manage when networks grow, change frequently, or require fast policy updates.

SDN addresses many of these limitations by separating the control plane from the data plane. The control plane is moved to a logically centralized controller, while switches primarily perform packet forwarding according to rules issued by that controller. This design improves network visibility, simplifies policy management, enables vendor-independent operation, and makes it easier to automate network behavior. As a result, SDN is attractive for cloud computing, 5G systems, Internet of Things (IoT) networks, data centers, and large enterprise infrastructures.

The same architectural feature that makes SDN powerful also introduces a security risk. Because the controller has a global view of the network and manages forwarding rules, it becomes a high-value target. A successful DDoS attack can flood the controller with excessive requests, overload switch flow tables, consume bandwidth, or interrupt communication between the controller and network devices. Once the controller is slowed down or disabled, legitimate flows may be delayed, dropped, or incorrectly handled. Therefore, protecting SDN from DDoS attacks is a central problem in network security research.

DDoS attacks are normally launched through many compromised machines, often called bots or zombies. The direct victim is the service or network being attacked, while the compromised machines act as secondary victims because they are abused to generate malicious traffic. This

distributed nature makes attribution difficult and enables attackers to produce very high traffic volumes. DDoS attacks may be motivated by financial gain, political conflict, extortion, sabotage, or attempts to disrupt public services and business operations.

1.1. DDoS Attack Categories

The reviewed paper organizes DDoS attacks into two broad groups: bandwidth-depletion attacks and resource-depletion attacks. Bandwidth-depletion attacks attempt to saturate the network link so that legitimate traffic cannot reach the service. Resource-depletion attacks try to exhaust computing, memory, protocol, or flow-table resources so that a server, controller, or switch cannot process normal requests.

Bandwidth attacks include flood and amplification attacks. Flood attacks send large volumes of traffic directly toward the target. UDP floods and ICMP floods are common examples. In a UDP flood, attackers send many datagrams to random or selected ports, forcing the victim to process them and sometimes respond with unreachable messages. In an ICMP flood, large numbers of ping-related packets consume the victim network bandwidth and processing capacity. Amplification attacks exploit broadcast or reflection behavior so that a small attacker request causes a much larger response to be sent toward the victim. Smurf and Fraggle attacks are examples of such amplified traffic patterns.

Resource-depletion attacks include protocol-exploitation and malformed-packet attacks. TCP SYN floods misuse the connection setup process by sending many incomplete connection requests, tying up server or controller resources. PUSH+ACK misuse forces unnecessary buffer handling. Malformed-packet attacks send invalid or unusual packet structures, such as packets with confusing source and destination addresses or manipulated IP options, which can cause additional processing effort or even system crashes.

2. Literature Review

The reviewed article surveys a wide range of studies that use SDN, open datasets, machine learning, deep learning, and security frameworks to detect or mitigate DoS and DDoS attacks. The works differ in their testbeds, controllers, traffic-generation tools, datasets, features, algorithms, and evaluation metrics. Some studies focus on SDN-enabled IoT environments; others focus on controller protection, flow-table overload, traceback, vehicular networks, or general intrusion detection.

2.1. SDN-Managed IoT Networks

IoT environments are especially vulnerable because many devices have limited memory, processing power, and security capabilities. Several studies reviewed in the source paper combine SDN with automated intrusion detection to improve IoT network protection. One approach uses flow-feature extraction and a Random Forest classifier at the SDN application layer to identify malicious traffic and install high-priority rules in the data plane. This design shows that SDN can respond actively to detected attacks, not merely observe them. However, some experiments were limited to a small number of attack types or relied on one malicious host, which does not fully represent real-world botnet behavior.

Other researchers generated SDN-based IoT datasets containing millions of records from static and dynamic networks. These datasets included benign traffic and attacks such as DoS, DDoS, port scanning, operating-system fingerprinting, and fuzzing. Such datasets are valuable because they reflect IoT-specific traffic, but the lack of fully open traffic-generation tools can make reproduction difficult.

The Bot-IoT dataset is another important resource. It contains normal IoT traffic and multiple attack categories associated with botnet behavior. Researchers evaluated models such as Support Vector Machines, Recurrent Neural Networks, and Long

Short-Term Memory networks on this dataset. The dataset helped demonstrate how statistical and machine learning methods can detect IoT attacks, although its very large size made full-scale testing difficult and required researchers to use smaller samples.

Frameworks such as Counter-based DDoS Attack Detection (C-DAD) and SD-IoT detection algorithms also show the value of SDN in IoT networks. These approaches observe counters, packet-in rates, or similarity patterns to detect attack behavior. Their main benefit is fast reaction, but some are tested only in limited simulated environments, and their performance across multiple heterogeneous IoT networks remains uncertain.

2.2. Detection and Mitigation of DoS and DDoS Using SDN

A central group of studies focuses on detecting and mitigating attacks through SDN controllers and programmable switches. Stateful SDN approaches use extensions to OpenFlow switches so that traffic states can be monitored and suspicious behavior can be acted upon quickly. Experiments using Ryu, Mininet, and datasets such as Bot-IoT show that entropy and bandwidth-related measurements can identify attack periods. However, using only one attack protocol, such as UDP, limits the generality of findings.

Other researchers propose two-phase detection systems in which packet-header information is collected first and then used to compute trust scores for users or flows. A dynamic trust value can help distinguish suspicious users from legitimate users more flexibly than a single static threshold. Nevertheless, threshold-based systems always face the challenge of balancing false positives and false negatives. If the threshold is too strict, normal users may be blocked; if it is too lenient, attacks may pass unnoticed.

Machine learning has also been integrated into SDN defense systems. For example, convolutional neural networks have been tested for detecting DDoS traffic by inspecting network activity in short time intervals. These systems may be combined with mitigation modules that adjust packet discard policies or install defensive forwarding rules. The advantage is accurate detection, while the limitation is that testbeds often contain fewer hosts and simpler traffic patterns than production networks.

Traceback mechanisms attempt to identify the source path of malicious traffic. Hybrid deep learning models such as CNN-ELM have been evaluated with datasets including CICIDS2017 and InSDN. The reported results show high accuracy, but supervised models require labeled training data. In real operational environments, obtaining accurate labels for new and changing attacks remains difficult.

Frameworks such as ProDefence and OpCloudSec represent broader architectural responses. ProDefence uses components such as flow collectors, policy engines, detectors, and mitigation engines to support smart-city-scale SDN environments. OpCloudSec uses anomaly detection and deep belief networks to improve attack response in cloud settings. These frameworks are promising, but their generalizability depends on controller consistency, programming environments, and whether the datasets used truly represent SDN behavior.

2.3. Deep Learning and Machine Learning Techniques

The reviewed literature shows a strong movement toward machine learning and deep learning for DDoS detection. LSTM-based frameworks are used because they can learn temporal patterns in traffic. This is useful when attacks develop over time or when normal traffic has seasonal behavior. Source-side DDoS detection using LSTM

can reduce downstream congestion, but it requires careful selection of collaborating detection sites and adaptive treatment of changing traffic patterns.

Semi-supervised learning is another direction because attack labels are often incomplete or expensive to obtain. The LEDEM approach, based on semi-supervised extreme learning, demonstrates that partially labeled data can still support effective DDoS detection. However, semi-supervised methods can also suffer from false positives when normal traffic patterns are incorrectly interpreted as suspicious.

Traditional machine learning algorithms remain important because they can be faster and easier to deploy than complex neural networks. Decision Trees, Random Forests, Support Vector Machines, Multi-Layer Perceptrons, K-Nearest Neighbors, Naive Bayes, and Logistic Regression have all been evaluated in SDN attack-detection settings. Random Forest often produces high accuracy, while Decision Trees can offer shorter processing time. In SDN, detection speed matters because an accurate model that reacts too slowly may still fail to protect the controller.

Feature selection is a recurring theme. Studies comparing Information Gain, correlation coefficients, Chi-square, forward selection, backward elimination, recursive feature elimination, and embedded methods such as Lasso demonstrate that feature quality strongly affects classification performance and controller load. Reducing irrelevant features can increase accuracy and decrease computational burden. Yet oversimplification may remove features that are useful for identifying complex attacks.

Entropy-based detection is also widely used because attack traffic often changes the distribution of source IP addresses, destination ports, packet counts, or flow characteristics. Static entropy thresholds are easy to implement but may not work across different datasets. Dynamic thresholds can adapt to changing traffic, but they may increase false positives unless combined with classification models.

2.4. XGBoost-Based Intrusion Detection

The source paper also reviews studies that use XGBoost for intrusion detection, including settings related to vehicular ad hoc networks and IoT systems. XGBoost is attractive because it combines multiple decision trees through gradient boosting and often performs well on tabular security datasets. In one reviewed study, Chi-square feature selection reduced a large feature set to a smaller group, while SMOTE addressed class imbalance. This improved model efficiency and helped reduce bias toward the majority class.

Other systems use artificial neural networks or XGBoost on datasets such as UNSW-NB15 and IoT-23. These studies show that modern classifiers can achieve good detection rates, but fair comparison requires consistent datasets, comparable preprocessing, and equivalent evaluation protocols. When different studies use different datasets, sampling rates, or attack categories, their accuracy values cannot be compared directly without caution.

2.5. Artificial Intelligence and Machine Learning in SDN Security

AI and machine learning are increasingly viewed as essential tools for SDN and network function virtualization. Their long-term promise is to support networks that can self-configure, self-adapt, and self-protect. However, this promise is limited by scarce public datasets, constrained network resources, and the need to make decisions quickly without overloading the controller.

Some research uses moving target defense and cyber deception. These methods aim to mislead attackers by changing the apparent attack surface, making reconnaissance and exploitation more difficult. SDN is suitable for such techniques because it can modify forwarding rules and network views dynamically. Nevertheless, deception

frameworks may still be vulnerable to resource-exhaustion attacks and may struggle to detect insider threats.

Other approaches combine algorithms, such as Support Vector Machines with Kernel Principal Component Analysis and Genetic Algorithms. KPCA can extract more useful nonlinear features, while genetic algorithms can optimize model parameters. Combined algorithms may improve accuracy, but they can increase complexity and may not scale well when multiple controllers, switches, and large traffic volumes are involved.

The InSDN dataset is an important contribution because it was designed specifically for SDN intrusion research. It contains normal traffic and multiple attack categories, including DoS, DDoS, web attacks, botnet traffic, password guessing, probing, and exploitation. Evaluation with classifiers such as Decision Tree, Random Forest, AdaBoost, KNN, Naive Bayes, SVM, and MLP shows the dataset's usefulness. At the same time, class imbalance, controller diversity, and limited hardware scale remain weaknesses.

3. Open Datasets for DDoS and SDN Security Research

Open datasets are one of the most important requirements for progress in SDN security. They allow researchers to train models, compare methods, repeat experiments, and evaluate detection accuracy under similar conditions. Without shared datasets, each study may use a different environment, making results difficult to compare. The reviewed article therefore places strong emphasis on datasets containing DDoS attacks and, where possible, SDN-specific traffic.

Older datasets such as DARPA 1998, DARPA 1999, and DARPA 2000 contributed to early intrusion detection research by providing labeled attack and normal traffic. They remain historically important, but they do not fully represent modern SDN, cloud, IoT, or encrypted traffic environments. The CAIDA DDoS 2007 dataset is useful for studying real DDoS traffic traces, although it lacks the full range of SDN-specific controller and flow-table behavior.

Datasets such as KDD99 and NSL-KDD are widely used in machine learning experiments, but they are frequently criticized for age, redundancy, unrealistic traffic distribution, and limited reflection of current attack techniques. Despite this, many studies still use them because they are accessible and familiar. Researchers must therefore interpret high accuracy on these datasets carefully.

More recent datasets include UNB ISCX, CICIDS2017, CSE-CIC-IDS2018, and CICDDoS2019. These datasets provide richer traffic and broader attack categories, and CICDDoS2019 contains many DDoS variants. They are useful for training and testing machine learning models, but not all were generated in SDN testbeds. Therefore, results from these datasets may not fully predict performance in SDN networks.

IoT-focused datasets such as Bot-IoT, ToN-IoT, and IoT-23 help researchers study attacks against IoT devices and botnets. Their advantage is that they include IoT-specific traffic and attack behavior. Their weakness is that they can require heavy preprocessing, may have strong class imbalance, and may not include enough SDN-controller behavior.

InSDN and other SDN-oriented datasets are especially valuable because they include traffic generated in SDN environments. They help researchers test how attacks interact with controllers, switches, flow tables, and SDN APIs. However, even SDN datasets are often limited by small testbeds, one controller type, restricted hardware resources, and a narrower range of topologies than real enterprise or carrier-grade networks.

Table 1. Condensed overview of dataset groups discussed in the reviewed paper

Dataset group	Typical examples	Main value	Main limitation
Classical IDS datasets	DARPA, KDD99, NSL-KDD	Useful for historical comparison and basic ML testing	Often outdated and not SDN-specific
Modern IDS/DDoS datasets	UNB ISCX, CICIDS2017, CSE-CIC-IDS2018, CICDDoS2019	Broader attack coverage and richer traffic	May not capture SDN controller behavior
IoT datasets	Bot-IoT, ToN-IoT, IoT-23	Important for IoT botnet and device-level attacks	Large size, imbalance, preprocessing complexity
SDN-specific datasets	InSDN and SDN testbed datasets	Closer to controller, switch, and flow-table behavior	Limited topology scale and controller diversity

4. Detection and Mitigation Strategies

The reviewed paper groups defensive work into detection and mitigation. Detection identifies whether a flow, host, or traffic pattern is suspicious. Mitigation then attempts to reduce damage by filtering packets, installing new rules, rate-limiting traffic, isolating sources, balancing loads, or changing routing behavior. In SDN, these tasks can be coordinated centrally through the controller, which is both an advantage and a risk.

Flow-based monitoring is common because SDN switches expose useful information about flows, ports, packet counts, byte counts, and flow-table entries. By observing unusual increases in packet-in messages, flow creation requests, entropy changes, or traffic concentration, a controller can detect signs of a DDoS attack. The challenge is to collect enough information for accurate detection without overwhelming the controller with monitoring tasks.

Rule-based and threshold-based systems are easy to deploy and can respond quickly. For example, a controller can block sources that exceed a packet-in threshold or ports that show abnormal traffic levels. However, attackers can evade static thresholds by using low-rate or distributed patterns. Thresholds also need to adapt to normal traffic variation, such as daily peaks or sudden legitimate demand.

Machine learning systems improve adaptability by learning normal and malicious traffic patterns from data. Random Forest, Decision Tree, SVM, KNN, XGBoost, CNN, LSTM, and hybrid models all appear in the reviewed studies. Their performance is usually measured with accuracy, precision, recall, F1-score, false positive rate, and sometimes detection time. High accuracy is useful, but SDN systems also require low latency and low computational overhead.

Mitigation may involve installing high-priority drop rules, redirecting suspicious traffic, limiting packet rates, updating forwarding paths, balancing traffic across controllers, or using traceback methods to locate attack sources. Some frameworks also use game theory, deception, or moving-target defense to make attacks less effective. The best mitigation strategy depends on the attack type. A UDP flood, a TCP SYN flood, a low-rate DDoS attack, and a flow-table exhaustion attack each require different indicators and responses.

Table 2. Paraphrased summary of defense strategies

Strategy	How it works	Key concern
Entropy/statistical detection	Tracks changes in traffic distribution, packet rates, or flow behavior	Static thresholds may fail across different networks
Machine learning classification	Learns attack and benign patterns from labeled or partially labeled datasets	Needs high-quality datasets and low controller overhead
Deep learning	Uses temporal or spatial traffic patterns through CNN, LSTM, DBN, or hybrid models	May require large data and careful deployment
Traceback and source identification	Attempts to locate the origin or path of malicious traffic	Hard when botnets use spoofing and distributed sources
Rule installation and filtering	Blocks, redirects, or limits suspicious traffic through SDN rules	Incorrect rules may block legitimate users
Distributed controller defense	Spreads control workload across multiple controllers	Needs consistency and secure controller coordination

5. Critical Discussion

The main strength of the reviewed paper is that it brings together three connected topics: DDoS attack vectors, public datasets, and SDN detection and mitigation frameworks. This combination is useful because a defense system cannot be evaluated properly without suitable data, and datasets are meaningful only when they support realistic attack and mitigation research.

A major finding is that dataset availability remains a serious challenge. Many studies report very high accuracy, sometimes above 99%, but those results are often obtained through offline experiments. In real SDN networks, traffic changes quickly, attack patterns evolve, and controller resources are limited. A classifier that performs well on an offline dataset may produce different results when deployed inside a controller that must make decisions in real time.

Another issue is reproducibility. Research papers often use different traffic generators, simulation settings, controller platforms, feature sets, sampling methods, preprocessing steps, training-test splits, and performance metrics. Because of these differences, comparing one accuracy value with another can be misleading. Standardized benchmark datasets and consistent evaluation procedures would make the field more reliable.

Scalability is also central. Many experiments are conducted with Mininet and a small number of switches, hosts, and controllers. Such testbeds are useful for controlled research, but they cannot fully represent large service-provider networks, smart cities, 5G systems, or industrial IoT infrastructures. Future studies need larger, more diverse, and more realistic SDN environments.

The reviewed work also shows that attack diversity matters. DDoS is not a single technique. Bandwidth floods, amplification attacks, protocol misuse, malformed packets, low-rate attacks, and flow-table exhaustion attacks behave differently. A

defense model trained on one attack type may not detect another. Modern benchmark datasets should therefore include multiple DDoS families, benign traffic variations, encrypted traffic, and mixed attack scenarios.

Finally, SDN security must balance accuracy with operational cost. Monitoring every packet or running heavy deep learning models in the controller may reduce performance. Lightweight feature extraction, edge-assisted detection, distributed controllers, and efficient mitigation rules are therefore important directions for practical deployment.

6. Conclusion and Future Work

The reviewed paper concludes that DDoS defense in SDN depends on three foundations: understanding attack vectors, using appropriate open datasets, and designing detection and mitigation frameworks that can operate efficiently in SDN environments. SDN improves flexibility and centralized control, but it also increases the importance of protecting the controller, switches, APIs, and flow tables.

Open datasets allow researchers to test models and compare results, but current datasets have limitations. Some are outdated, some are not SDN-specific, some are too small or imbalanced, and some do not include enough attack diversity. SDN-specific datasets such as InSDN are valuable, but more comprehensive benchmark datasets are needed.

Future work should focus on building realistic SDN testbeds, generating datasets with multiple topologies and controllers, including broad DDoS scenarios, and defining standard evaluation metrics. Collaboration between universities, industry, and cybersecurity organizations can help create datasets that are both realistic and shareable. In addition, future models should be evaluated not only by accuracy but also by detection delay, controller overhead, scalability, false-positive cost, and mitigation effectiveness.

Overall, machine learning and deep learning will continue to play an important role in SDN security, but they must be supported by reliable data, reproducible experiments, and practical deployment strategies. A strong DDoS defense for SDN should combine fast traffic monitoring, adaptive detection, accurate classification, controller-aware mitigation, and continuous evaluation against new attack patterns.

Reference

Hill, W., Acquaah, Y. T., Mason, J., Limbrick, D., Teixeira-Poit, S., Coates, C., & Roy, K. (2024). DDoS in SDN: a review of open datasets, attack vectors and mitigation strategies. Discover Applied Sciences, 6, Article 472. https://doi.org/10.1007/s42452-024-06172-x

S. Sezer et al., “Are we ready for SDN? Implementation challenges for software-defined networks,” IEEE Commun. Mag., vol. 51, no. 7, pp. 36–43, Jul. 2013.

I.A. Valdovinos et al., “Emerging DDoS attack detection and mitigation strategies in software-defined networks: Taxonomy, challenges and future directions,” J. Netw. Comput. Appl., vol. 187, 2021, Art. no. 103093.

Open Networking Foundation, Jun. 2014. [Online]. Available: https://www.opennetworking.org/

C. Hu, L. Han, and S.M. Yiu, “Efficient and secure multi-functional searchable symmetric encryption schemes,” Secur. Commun. Netw., vol. 9, pp. 34–42, 2016.

A.Praseed and P.S. Thilagam, “DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web

applications," IEEE Commun. Surv. Tutor., vol. 21, pp. 661–685, 2019.

T.Mahjabin, Y.Xiao, G.Sun, and W.Jiang, "A survey of distributed denial of-service attack, prevention, and mitigation techniques," Int. J. Distrib. Sens. Netw., vol. 13, 2017.

M.A.M. Yusof, F.H.M. Ali, and M.Y. Darus, "Detection and Defense Algorithms of Different Types of DDoS Attacks," Int. J. Eng. Technol., vol. 9, pp. 410–444, 2018.

N. Ahuja, G. Singal, and D. Mukhopadhyay, "DDOS attack SDN Dataset," Mendeley Data, V1, 2020, doi: 10.17632/jxpfjc64kr.1. Available: https://www.kaggle.com/datasets/aikenkazin/ddos-sdn-dataset

I.Sharafaldin, A.H. Lashkari, S. Hakak, and A.A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Tax onomy," in Proc. 2019 Int. Carnahan Conf. Security Technol. (ICCST), Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419.

Ö. Tonkal, H. Polat, E. Başaran, Z. Cömert, and R. Kocaoğlu, "Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking,"Electronics, vol. 10, no. 1227, 2021.

J.A. Perez-Diaz, I.A. Valdovinos, K.K.R. Choo, and D. Zhu, "A Flex ible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning," IEEE Access, vol. 8, pp. 155859–155872, 2020.

N. Ravi and S.M. Shalinie, "Learning-Driven Detection and Mitigation of DDoS Attack in IoT via SDN-Cloud Architecture," IEEE Internet Things J., vol. 7, pp. 3559–3570, 2020.

H. Polat, O. Polat, and A. Cetin, "Detecting DDoS Attacks in Software Defined Networks Through Feature Selection Methods and Machine Learning Models," Sustainability, vol. 12, no. 1035, 2020.

M. AbdulRaheem, I.D. Oladipo, and A.L. Imoize, "Machine learning assisted snort and zeek in detecting DdoS attacks in software-defined networking," Int. J. Inf. Technol., 2023. Available: https://doi.org/10.1007/s41870-023-01469-3

A.Makuvaza, D.S. Jat, and A.M. Gamundani, "Deep Neural Network (DNN) Solution for Real-time Detection of Distributed Denial of Service (DDoS)Attacks in Software Defined Networks (SDNs)," SN Comput. Sci., vol. 2, 2021.

M. Mittal, K. Kumar, and S. Behal, "DL-2P-DDoSADF: Deep learning based two-phase DDoS attack detection framework," J. Inf. Secur. Appl., vol. 78, 2023, Art. no. 103609.

A.Makuvaza, D.S. Jat, and A.M. Gamundani, "Hybrid deep learning model for real-time detection of distributed denial of service attacks in soft ware defined networks," in Emerging Trends in Data Driven Computing and Communications: Proc. DDCIoT 2021, Springer Singapore, 2021, pp. 1-13.

S. Haider, A. Akhunzada, G. Ahmed, and M. Raza, "Deep Learning based Ensemble Convolutional Neural Network Solution for Distributed Denial of Service Detection in SDNs," in Proc. 2019 UK/China Emerging Technologies (UCET), 2019, pp. 1-4, doi: 10.1109/UCET.2019.8881856.

C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of DDoS attack–based on

deep learning in OpenFlow-based SDN," Int. J. Commun. Syst., vol. 31, no. 5, p. e3497, 2018.

T. E. Ali, Y.-W. Chong, and S. Manickam, "Machine learning techniques to detect a DDoS attack in SDN: A systematic review," Applied Sciences, vol. 13, no. 5, p. 3183, 2023.

A.Hamarshe, H. I. Ashqar, and M. Hamarsheh, "Detection of DDoS At tacks in Software Defined Networking Using Machine Learning Models," in *International Conference on Advances in Computing Research*, 2023, pp. 640–651.

A. Alashhab, M. S. Zahid, B. Isyaku, A. A. Elnour, W. Nagmeldin, A. Abdelmaboud, T. A. A. Abdullah, and U. Maiwada, "Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model," IEEE Access, 2024.