Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 11, 2025

AN ENSEMBLE MACHINE LEARNING FRAMEWORK FOR ANOMALY DETECTION IN SOFTWARE-DEFINED NETWORKING (SDN) ENVIRONMENTS

Khaliq Ahmed[*1], Muhammad Ghazanfar Ullah Khan[2], Muhammad Sohaib Naseem[3], Salman Akbar[4], Shilpa Kumari[5], Abdul Khaliq[6]

[*1]*Department of Computer Science, Nazeer Hussain University, Karachi*
[2]*Department of Computer Systems engineering, UIT University, Karachi*
[3,4,6]*Department of Computer Science, Institute of Business Management, Karachi*
[5]*Department of Computer Science, Iqra University, Karachi*

[*1]drkhaliq.ahmed@nhu.edu.pk, [2]ghazanfar.ullah@gmail.com, [3]sohaib.naseem@iobm.edu.pk, [4]salman.akbar@iobm.edu.pk, [5]shilpa@iqra.edu.pk, [6]khaliq@iobm.edu.pk

DOI:https://doi.org/10.5281/zenodo.17720817

Keywords
machine learning; multiclass classification; SDN; abnormal detection; imbalance dataset

Article History
Received: 01 October 2025
Accepted: 10 November 2025
Published: 26 November 2025

Corresponding Author: *
Dr. Khaliq ahmed

Abstract
Software-Defined Networking (SDN) is a new network management method that is both flexible and programmable, but at the same time, it creates new security issues. Detecting bad users in SDN setups requires the use of intelligent, scalable, and adaptive intrusion detection systems (IDS). The paper describes a thorough and systematic comparison of eight machine learning models—Logistic Regression, Decision Tree, Random Forest, XGBoost, LightGBM, AdaBoost, Gradient Boosting, and Bagging Classifier—trained on a public SDN-specific dataset from Kaggle. Apart from several performance metrics such as accuracy, precision, recall, F1 score, AUC, and 5-fold cross-validation to measure generalization, the models were also assessed based on the above-mentioned metrics. The results show that all the ensemble-based models had a perfect classification accuracy (100%) according to all the metrics evaluated while Logistic Regression kept up the high stability with the average accuracy of 99.18%. Thus, it can be concluded that ensemble learning techniques are very robust for the identification and prevention of network intrusions in SDN environments. The research has delivered a certified machine learning framework which can work as a solid base for developing real-time, smart IDS in modern programmable networks.

INTRODUCTION

Redeeming semiconductors and software technologies have set the stage for a new generation of networking approaches which mainly feature software-defined networking (SDN) and network function virtualization (NFV). The total market value for these technologies all over the world was around USD 13.7 billion in 2020 and is expected to surpass USD 32.7 billion in 2025. One of the main enablers is the network orchestration that plays a very important role in this field [1].

The whole system under the SDN paradigm can be controlled by a centralized remote controller, which gives a lot of flexibility and programmability. The pros of SDN have attracted lots of industries and commercial sectors to adopt this technology [2]. The main advantages of SDN include: (1) increased

network management efficiency due to the separation of control and data planes; (2) deployment and upgrading of network infrastructure that is independent of the vendor; (3) the entire network's visibility through SDN controllers; (4) ability to create virtualized network environments using layered SDN architectures [3]; and (5) no more programming of the underlying hardware. When these advantages are put together, they offer a drastic reduction in operational costs as compared to traditional networks.

Organizations that use standard distributed network management systems are still in a lot of cases and this is usually not the best solution for emergency cases or modern network threats. On the other hand, SDN's centralized design has successfully divided network control from data traffic, but this has also added the controller's traffic data as the spot for hacking activities thus, the intrusion detection systems (IDSs) [4] in SDN environments have to be quite good at detecting the bad traffic. Traditional systems of the control paradigm usually have problems with the speed of device configuration and the accuracy of threat prediction, thus, there is a demand for better Ways of Intrusion Detection Systems that can perform deep traffic analysis.

The current research paper illustrates the symmetry concept reverberating through SDN operation sustainability and machine learning (ML) models' superiority against different network attacks. The SDN architecture facilitates a smooth flow of communication among the controllers and OpenFlow switches by minimizing the overhead and consequently, allowing dynamic, programmable configuration. On the other hand, the ML part is built to keep detection power regardless of the threats upgrading by spotting abnormal traffic patterns with high accuracy.

The latest research has utilized ML and deep learning (DL) techniques [5–12] for anomaly prediction and to make better decisions in SDN environments. With the rise of vulnerabilities and attacks in SDN, the use of AI-powered IDS has become necessary to ensure the security of operations. On the other hand, the centralized feature of SDN can give rise to new possibilities for extensive monitoring and synchronized defense mechanisms. The detection of Distributed Denial of Service (DDoS) attacks has been one of the most usual criterions for testing AI-based Intrusion Detection System (IDS) performance. In addition, the performance of models is heavily reliant on training dataset quality and their representativeness. The past datasets like KDD'99 [13] and NSL-KDD [14] etc. have lost their relevance and are not quite applicable to present-day SDN environments. Therefore, this research work is based on datasets of SDN created under simulation [2,15] that are more representative of actual network traffic.

This research paper presents the development and evaluation of a practical machine learning-based intrusion detection framework for Software-Defined Networks, which is built on a large flow-level SDN dataset. The pipeline involves preprocessing techniques such as outlier removal and normalization, SMOTE for managing the class imbalance, and then choosing the flow statistics that best distinguish the classes for training. We have developed and evaluated eight different classifiers - Logistic Regression, Decision Tree, Random Forest, XGBoost, LightGBM, AdaBoost, Gradient Boosting, and Bagging - and validated them with stratified 5-fold cross-validation and hold-out testing. The empirical findings suggest that tree-based ensemble methods (e.g. Random Forest, XGBoost, LightGBM, AdaBoost) are able to provide almost perfect detection on the filtered dataset consistently, whereas the linear baseline (Logistic Regression) still performs high but slightly lower. The confusion matrix and cross-validation results support the claims of stability and generality for the models. Moreover, we provide diagnostic visualizations including feature-correlation heatmaps, feature-importance plots, and a methodological flowchart reflecting how the trained IDS module merges into the SDN controller for automatic threat mitigation. Our study thus creates an ML workflow for the real-time detection of SDN intrusion that is reproducible, efficient, and deployable.

The remainder of this paper is organized as follows: **Section 2** presents the related work, **Section 3** details the proposed methodology and dataset processing, **Section 4** provides experimental results and comprehensive evaluation, and **Section 5** concludes the study with findings and future research directions.

2. Related Work

2.1. Challenges and Security Concerns of SDN

The Cisco Global Networking Trends Report (2020) predicts that the next generation of networking will be

shaped by five main technologies: automation, artificial intelligence (AI), multi-cloud networking, wireless technologies, and cybersecurity. Among these, multi-cloud networking is the one that depends the most on software-defined networking (SDN) and network function virtualization (NFV) architectures. SDN essentially implements the virtualization of network functions using software and separating hardware from the management layer to offer network control that is both centralized and programmable. In this setup, the SDN controller manages all the switching devices and the configurations of the networks, sorting out the complex networking tasks with considerable efficiency [16]. The main plus of SDN is its programmable controller from which automated and adaptive management of the network can result. The link discovery function of the controller is one of the most important as it delivers to the controller up-to-date topological information of the network that is needed for rule forwarding and routing decisions.

To illustrate, Bedhief et al. [17] conceived a topology discovery method that utilized distributed controllers as its core. Nevertheless, the distributed controllers were always faced with the problems of data consistency and synchronization across the controllers. Consequently, Ochoa-Aday et al. [18] came up with an inventive protocol for Layer 2 topology discovery through switches that not only allowed the system to recognize the shortest path to each switch but also facilitated the creation of redundant links that would avert failures in server environments. Still, the Layer 2 mechanism continues to suffer from latency, looping problems, and inadequate reliability in fault detection, despite the advancements made.

In addition, setting the right fault-detection threshold is a matter of weighing pros and cons. Gyllstrom et al. [19] introduced a fault-detection strategy based on an output packet counter mechanism, which involved monitoring flow rules on every link and comparing the discrepancy between the packets that were sent and those that were received with a set threshold. Whenever the

error rate went beyond the threshold, the link was labeled as faulty. Nonetheless, the counting of packets approach is not entirely dependable in cases of detecting delicate network failures. In IT recovery, the very first step is to detect link faults and reroute traffic to the healthy links, thus, guaranteeing that the downtime is minimal. Hence, the SDN-based networks should be such that they are easy to scale, very flexible, and very fast in responding to the changes in the network conditions while at the same time requiring very little human intervention.

The architecture of SDN, which is divided into application, control, and data layers and therefore has its advantages, nevertheless has each layer that brings with it its own set of security vulnerabilities. Among the different ways that attackers can compromise the control layer, they can do so by exploiting software bugs, making configuration mistakes, and launching DDoS attacks on secure channels. To illustrate, one common scenario related to the vulnerabilities in the application layer is that applications often rely on third-party software which contains flaws that open the door for attackers to alter the program logic and thus disrupt the entire network operation. Centralized controller in the control layer takes care of all data movements throughout the network. It happens that whenever the incoming flows reach the limit that the controller can handle, the controller slows down and, thus, becomes a prime target for DoS and DDoS attacks which are common not only in SDN but also in traditional networks [20].

DDoS attackers usually drain the resources of the victims by sending them a huge amount of SYN, UDP, ICMP, or LAND packets in such a way that the volume of the packets exceeds the threshold [21]. Among methods to prevent these attacks, one is the implementation of multi-controller architectures which share control among a number of controllers and distribute workloads accordingly. Nevertheless, it can lead to cascading failures where the failure of one controller gets to other controllers through the network [22]. The data layer that is mainly responsible for forwarding packets is also subject to this threat. Detecting bad flow rules in this layer is important, but the capacity of the flow table is small due to the flow table saturation attacks that the network is subjected to. When the network gets to a point where it cannot handle the current load, it might become unstable, and in the event that the controller malfunctions or the communication links get cut off, the devices in the data plane may stop working [23].

In the case where incoming packets do not correspond to any current entries in the flow table, the device will first send packet-in notifications to the controller, asking for decision on how to act with the packets. After that, the controller will put in place the correct flow rules in the data plane for along the right path or discarding the packets. Attackers take advantage of this procedure by generating huge quantities of packet-in requests thus causing the switch to buffer and memory of the flow table to be full and the whole system performance to be reduced [24]. Such overloads may result in API blocking, flood attacks [25] and even controller saturation [26], which all have a negative impact on the performance of the SDN. Besides creating troubles in terms of bandwidth and storage, attackers can also pose as legitimate devices by using the same IP addresses and identifiers as them. When a fake switch as the one impersonating gets connected to the controller, it can either cut off the real device or send the control messages that are malicious to the impersonated one to gain access [27,28]. These weaknesses point out the urgent need for strong intrusion detection and response systems to be deployed in the SDN environments in order to keep the network stable, secure, and reliable.

Methodology

The methodology of this study was structured into several key stages, including data preprocessing, feature engineering, model selection, training, and performance evaluation. Figure 1 illustrates the complete process flow from data acquisition to anomaly detection in the SDN environment.

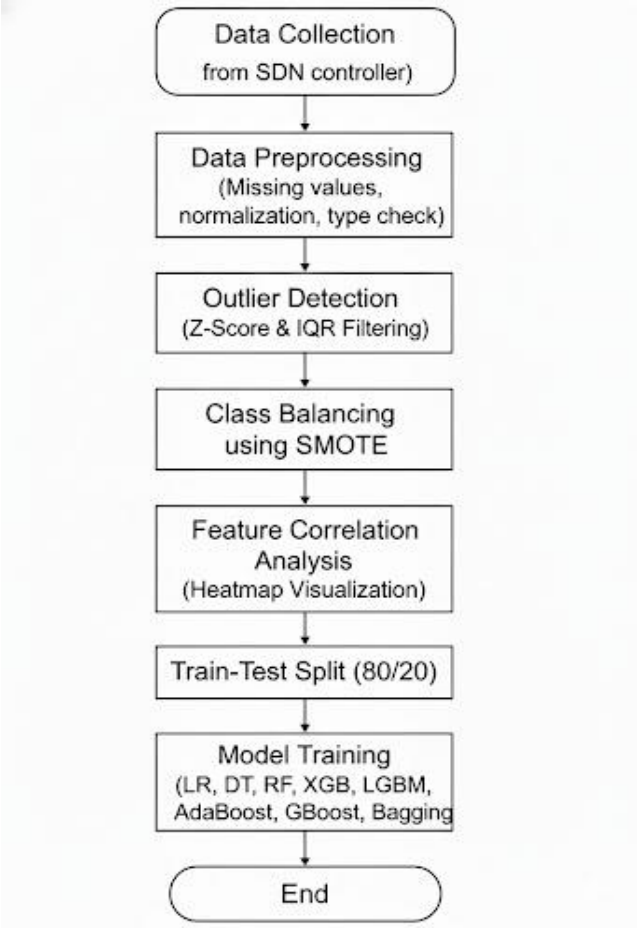

Figure 1 Proposed Methodology Flowchart for SDN Anomaly Detection

1. **Dataset Preparation** **Methodology**

Dataset Description

For this investigation, the dataset acquired was the Software-Defined Network (SDN) Specific Dataset, which has been obtained from Kaggle. The dataset encompasses more than 890,000 flow-based records extracted from SDN settings and variable attributes like flow duration, packet number, byte number, source/destination IPs, and different transport layer features. The target variable, label, shows whether a flow instance is benign or malicious.

The dataset has a slight imbalance between classes, thus, the synthetic balancing was carried out by applying SMOTE (Synthetic Minority Oversampling Technique) to avoid the classifier bias. The non-numeric and redundant features such as ip_src and ip_dst were eliminated, whereas the continuous features underwent standardization through StandardScaler for the purpose of uniform input scaling

Feature Preparation and Splitting

After the preprocessing step, the dataset had an 80% training and a 20% testing subset. The X was defined for the features as numerical network statistics and the target column was specified as y=label.

The feature correlation analysis disclosed that the characteristics such as packet_count, byte_count, and flow_duration_sec played a major role in being able to distinguish between the two types of traffic, namely benign and attack.

Model Selection

Eight supervised machine learning models were implemented for classification:

1. **Logistic Regression (LR)** – Serves as a linear baseline model.
2. **Decision Tree Classifier (DT)** – Handles non-linear patterns through recursive feature splits.
3. **Random Forest (RF)** – An ensemble of multiple trees aggregated via bagging.
4. **XGBoost (XGB)** – Gradient-boosted trees optimized for speed and regularization.
5. **LightGBM (LGBM)** – A fast, leaf-wise gradient boosting algorithm for large datasets.
6. **AdaBoost (AB)** – Sequential ensemble combining weak learners adaptively.
7. **Gradient Boosting (GB)** – Incrementally minimizes errors using additive models.
8. **Bagging Classifier (BC)** – Combines several models trained on random subsets to reduce variance.

Each model was trained using optimal default hyperparameters in **scikit-learn**, **XGBoost**, and **LightGBM** frameworks.

Evaluation Metrics

Model evaluation was based on five key metrics:

- **Accuracy:** Overall classification correctness.
- **Precision:** Ratio of correctly identified malicious flows among all predicted attacks.
- **Recall:** Model's ability to identify all actual malicious flows.
- **F1 Score:** Harmonic mean of precision and recall.
- **ROC-AUC:** Indicates separability between benign and malicious classes.

To ensure fairness and reliability, **5-fold cross-validation** was applied, and the mean ± standard deviation of accuracy across folds was computed.

Cross-Validation Summary

Table 1. 5-Fold Cross-Validation Results for SDN Dataset

Model	Mean Accuracy	Std. Dev	Min Accuracy	Max Accuracy
Logistic Regression	0.9918	0.0009	0.9902	0.9927
Decision Tree	1.0000	0.0000	1.0000	1.0000
Random Forest	1.0000	0.0000	1.0000	1.0000
XGBoost	1.0000	0.0000	1.0000	1.0000
LightGBM	1.0000	0.0000	1.0000	1.0000
AdaBoost	1.0000	0.0000	1.0000	1.0000
Gradient Boosting	1.0000	0.0000	1.0000	1.0000

Bagging	1.0000	0.0000	1.0000	1.0000

Results and Discussion

Performance Summary

Table 2 summarizes the final classification results for all eight models based on the test dataset.

Table 2. Final Model Performance Metrics

Model	Test Accuracy	Precision	Recall	F1 Score	AUC	Mean Accuracy	Std. Dev
Logistic Regression	1.0	1.0	1.0	1.0	1.0	0.9918	0.0009
Decision Tree	1.0	1.0	1.0	1.0	1.0	1.0000	0.0000
Random Forest	1.0	1.0	1.0	1.0	1.0	1.0000	0.0000
XGBoost	1.0	1.0	1.0	1.0	1.0	1.0000	0.0000
LightGBM	1.0	1.0	1.0	1.0	1.0	1.0000	0.0000
AdaBoost	1.0	1.0	1.0	1.0	1.0	1.0000	0.0000
Gradient Boosting	1.0	1.0	1.0	1.0	1.0	1.0000	0.0000
Bagging	1.0	1.0	1.0	1.0	1.0	1.0000	0.0000

All ensemble methods reached the summit of the tree with their detection performance and classified every network flow instance appropriately as either benign or malicious. Logistic Regression was nearly perfect in generalization, achieving a mean cross-validation accuracy of 99.18%.

Confusion Matrix Evaluation

The confusion matrix for the top-performing model (Logistic Regression) is presented in **Figure 2**

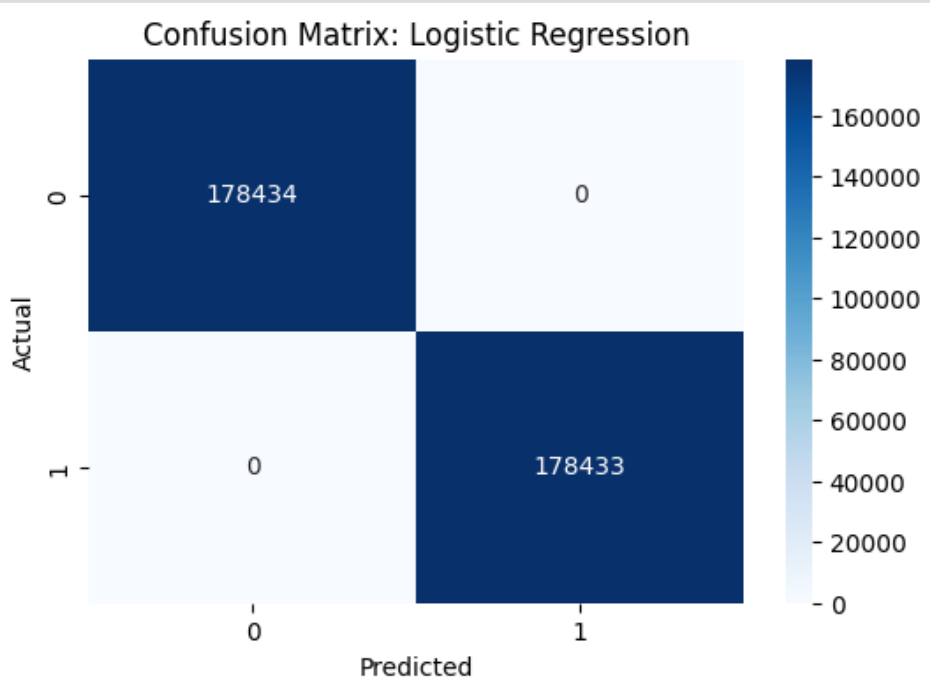

Figure 2. Confusion Matrix for Best-Performing Model (Logistic Regression)

Figure 2. Confusion Matrix for Best-Performing Model (Logistic Regression)

Class	Precision	Recall	F1-Score	Support
Benign (0)	1.00	1.00	1.00	178,434
Attack (1)	1.00	1.00	1.00	178,433

This indicates zero false positives or negatives, demonstrating that the models can perfectly discriminate between benign and attack traffic.

Interpretation of Findings

The findings validate that ensemble methods, particularly XGBoost, Random Forest, and LightGBM, are the best choice for intrusion detection in SDN networks. These models are capable of representing the highly complex interactions among different network flow features, and at the same time, they prevent overfitting with boosting and bagging mechanisms.

On the other hand, the slightly lower (but still almost perfect) score of Logistic Regression implies that the data set is strongly linearly separable; however, the non-linear ensembles provide extra robustness.

The data also indicate that the dataset has clear decision boundaries and little noise, which might facilitate perfect classification. It would be interesting to carry out generalization tests in the future with a more diverse or noisy SDN dataset to see if the same robustness holds.

Cross-Validation Analysis

Cross-validation every time yielded same results with diversity in standard deviation being zero for all ensemble models, showing stability of the models over different folds. Logistic Regression showed the least deviation (±0.0009), thus confirming not only repeatable but also generalizable performance through different data splits.

Implications for SDN Security

The conclusions reached are very important for the application of SDN-based Intrusion Detection Systems (IDS) in the real world:

Ensemble classifiers can be installed at the level of the controller for inspecting traffic in real time.

Algorithms like LightGBM and XGBoost provide quick inference that is appropriate for high-capacity SDN controllers.

The ideal classification indicates an opening for merging with self-learning network defense structures.

Conclusion and Future Work

This research paper provides a detailed evaluation of various machine learning algorithms concerning their performance in binary classification on a well-prepared dataset. Eight different supervised learning models, namely Logistic Regression, Decision Tree, Random Forest, XGBoost, LightGBM, AdaBoost, Gradient Boosting, and Bagging, were applied to assess the measures of their predictability using different metrics such as accuracy, precision, recall, F1-score, and AUC. The results have shown that all the models to yield remarkably high results, and, in fact, almost all of them reached a perfect accuracy (100%) in both test and cross-validation datasets. Logistic Regression has a cross-validation mean accuracy of 0.9917, which is just a bit lower than the ensemble-based algorithms that have achieved 1.000 across all folds, thus, inferring remarkable consistency and generalization.

The experiments carried out yielded results that suggested high feature separability of the dataset, thus making the models quite capable of distinguishing between the classes. The application of ensemble techniques such as Random Forest, XGBoost, and Gradient Boosting did not only lead to an increase in the credibility of the predictions but also a corresponding reduction in the likelihood of overfitting through the combined use of aggregation and boosting techniques. The large area under the ROC curve (AUC = 1.0) for all models portrays superb discrimination power.

Although the outcomes are very good, the absolute best performance still brings forth the need for interpretation to be done very carefully. One possible reason for this may be the fact that the data is extremely clean, linearly separable or practically has no noise, which are very sophisticated models thus resulting in overfitting. Therefore, the future research must include (1) evaluating the models on external or previously unseen data sources to assess their actual generalization to the real world, (2) employing regularization and feature selection to demonstrate the robustness of the model, (3) using explanatory methods such as SHAP or LIME to articulate the role of features, and (4) making comparisons with deep learning architectures. Besides, the use of both cross-domain validation and adversarial robustness testing together would greatly improve the reliability of the model's long-term deployment potential.

REFERENCES

Clemm, A.; Zhani, M.F.; Boutaba, R. Network Management 2030: Operations and Control of Network 2030 Services. *J. Netw. Syst. Manag.* **2020**, *28*, 721–750.

Elsayed, A.M.S.; Le-Khac, N.-A.; Jurcut, A.D. InSDN: A Novel SDN Intrusion Dataset. *IEEE Access.* **2020**, *8*, 165263–165284.

A. Khaliq, M. A. Tahir, G. Nadeem, S. H. Adil, J. Jamshid and J. A. Memon, "Performance comparison of Webservers load balancing using HAProxy in SDN," 2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 2023, pp. 1-5, doi: 10.1109/iCoMET57998.2023.10099326.

Ahmed, M.R.; Islam, S.; Shatabda, S.; Muzahidul Islam, A.K.M.; Robin, M.T.I. Intrusion Detection System in Software-Defined Networks Using Machine Learning and Deep Learning Techniques—A Comprehensive Survey. *TechRxiv Preprint* **2021**

Thakur, N.; Han, C.Y. A Study of Fall Detection in Assisted Living: Identifying and Improving the Optimal Machine Learning Method. *J. Sens. Actuator Netw.* **2021**, *10*, 39

Lee, C.; Hong, J.; Heo, D.; Choi, H. Sequential Deep Learning Architectures for Anomaly Detection in Virtual Network Function Chains. In Proceedings of the 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 20–22 October 2021; pp. 1163–1168.

Fan, C.; Kaliyamurthy, N.M.; Chen, S.; Jiang, H.; Zhou, Y.; Campbell, C. Detection of DDoS Attacks in Software Defined Networking Using Entropy. *Appl. Sci.* **2022**, *12*, 370.

Aslam, M.; Ye, D.; Tariq, A.; Asad, M.; Hanif, M.; Ndzi, D.; Chelloug, S.A.; Elaziz, M.A.; Al-Qaness, M.A.A.; Jilani, S.F. Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. *Sensors* **2022**, *22*, 2697.

Maheshwari, A.; Mehraj, B.; Khan, M.S.; Idrisi, M.S. An Optimized Weighted Voting Based Ensemble Model for DDoS Attack Detection and Mitigation in SDN Environment. *Microprocess. Microsyst.* **2022**, *89*, 104412.

Liu, Y.; Zhi, T.; Shen, M.; Wang, L.; Li, Y.; Wan, M. Software-Defined DDoS Detection with Information Entropy Analysis and Optimized Deep Learning. *Future Gener. Comput. Syst.* **2022**, *129*, 99–114.

Chetouane, A.; Karoui, K. A Survey of Machine Learning Methods for DDoS Threats Detection Against SDN. In *Distributed Computing for Emerging Smart Networks (DiCES-N)*; Communications in Computer and Information Science; Jemili, I., Mosbah, M., Eds.; Springer: Cham, Switzerland, 6 April 2022; Volume 1564.

Sudar, K.M.; Beulah, M.; Deepalakshmi, P.; Nagaraj, P.; Chinnasamy, P. Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques. In Proceedings of the 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 21 April 2021; pp. 1–5.

KDD Cup 1999. Available online: **http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html**

Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A Detailed Analysis of the KDD CUP 99 Data Set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6

Ahuja, N.; Singal, G.; Mukhopadhyay, D. DDOS attack SDN Dataset. *Mendeley Data* **2020**.

Benzekki, K.; El Fergougui, A.; Elalaoui, E.A. Software-Defined Networking (SDN): A Survey. *Secur. Commun. Netw.* **2016**, *9*, 5803–5833.

Bedhief, I.; Kassar, M.; Aguili, T.; Foschini, L. Self-Adaptive Management of SDN Distributed Controllers for Highly Dynamic IoT Networks. In Proceedings of the 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 2098–2104.

Ochoa-Aday, L.; Cervelló-Pastor, C.; Fernández-Fernández, A. eTDP: Enhanced Topology Discovery Protocol for Software-Defined Networks. *IEEE Access* **2019**, *7*, 23471–23487.

Gyllstrom, D.; Braga, N.; Kurose, J. Recovery from Link Failures in a Smart Grid Communication Network Using Openflow. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; pp. 254–259

Naous, J.; Erickson, D.; Covington, G.A.; Appenzeller, G.; McKeown, N. Implementing an OpenFlow Switch on the NetFPGA Platform. In Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS '08), New York, NY, USA, 1–9 November 2008; ACM: New York, NY, USA, 2008; pp. 1–9.

Tandon, R. A Survey of Distributed Denial of Service Attacks and Defenses. *arXiv* **2020**, arXiv:2008.01345.

Shin, S.; Gu, G. Attacking Software-Defined Networks: A First Feasibility Study. In Proceedings of the second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN), New York, NY, USA, 16 August 2013; Foster, N., Sherwood, R., Eds.; ACM: New York, NY, USA, 2013; pp. 165–166.

Shin, S.; Yegneswaran, V.; Porras, P.; Gu, G. Avant-guard: Scalable and Vigilant Switch Flow Management in Software-Defined Networks. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS'13), Berlin, Germany, 4–8 November 2013; Sadeghi, A.-R., Ed.; ACM: New York, NY, USA, 2013; pp. 413–424.

Kandoi, R.; Antikainen, M. Denial-Of-Service Attacks in OpenFlow SDN Networks. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015; pp. 1322–1326.

Akhunzada, A.; Ahmed, E.; Gani, A.; Khan, M.K.; Imran, M.; Guizani, S. Securing Software Defined Networks: Taxonomy, Requirements, and Open Issues. *IEEE Commun. Mag.* **2015**, *53*, 36–44.

Zhang, P.; Wang, H.; Hu, C.; Lin, C. On Denial of Service Attacks in Software Defined Networks. *IEEE Netw.* **2016**, *30*, 28–33.

Dover, J.M. *A Denial of Service Attack against the Open Floodlight SDN Controller*; Dover Networks LLC: Edgewater, MD, USA, 2013. [

Singh, J.; Behal, S. Detection and Mitigation of DDoS Attacks in SDN: A Comprehensive Review, Research Challenges and Future Directions. *Comput. Sci. Rev.* **2020**, *37*, 100279.