electronics

MDPI

Review

A Comprehensive Review of DDoS Detection and Mitigation in SDN Environments: Machine Learning, Deep Learning, and Federated Learning Perspectives

Sidra Batool [1,*], Muhammad Aslam [1], Edore Akpokodje [1] and Syeda Fizzah Jilani [2]

[1] Department of Computer Science, Aberystwyth University, Aberystwyth SY23 3FL, Wales, UK; mua19@aber.ac.uk (M.A.); eta@aber.ac.uk (E.A.)
[2] Department of Physics, Aberystwyth University, Aberystwyth SY23 3FL, Wales, UK; sfj7@aber.ac.uk
* Correspondence: sib54@aber.ac.uk

Abstract

Software-defined networking (SDN) has reformed the traditional approach to managing and configuring networks by isolating the data plane from control plane. This isolation helps enable centralized control over network resources, enhanced programmability, and the ability to dynamically apply and enforce security and traffic policies. The shift in architecture offers numerous advantages such as increased flexibility, scalability, and improved network management but also introduces new and notable security challenges such as Distributed Denial-of-Service (DDoS) attacks. Such attacks focus on affecting the target with malicious traffic and even short-lived DDoS incidents can drastically impact the entire network's stability, performance and availability. This comprehensive review paper provides a detailed investigation of SDN principles, the nature of DDoS threats in such environments and the strategies used to detect/mitigate these attacks. It provides novelty by offering an in-depth categorization of state-of-the-art detection techniques, utilizing machine learning, deep learning, and federated learning in domain-specific and general-purpose SDN scenarios. Each method is analyzed for its effectiveness. The paper further evaluates the strengths and weaknesses of these techniques, highlighting their applicability in different SDN contexts. In addition, the paper outlines the key performance metrics used in evaluating these detection mechanisms. Moreover, the novelty of the study is classifying the datasets commonly used for training and validating DDoS detection models into two major categories: legacy-compatible datasets that are adapted from traditional network environments, and SDN-contextual datasets that are specifically generated to reflect the characteristics of modern SDN systems. Finally, the paper suggests a few directions for future research. These include enhancing the robustness of detection models, integrating privacy-preserving techniques in collaborative learning, and developing more comprehensive and realistic SDN-specific datasets to improve the strength of SDN infrastructures against DDoS threats.

Keywords: software-defined networking; distributed denial-of-service attack; machine learning; deep learning; federated learning

check for updates

Academic Editor: Zbigniew Kotulski

Received: 26 August 2025
Revised: 14 October 2025
Accepted: 15 October 2025
Published: 29 October 2025

Citation: Batool, S.; Aslam, M.; Akpokodje, E.; Jilani, S.F.; Alnfiai, M.M. A Comprehensive Review of DDoS Detection and Mitigation in SDN Environments: Machine Learning, Deep Learning, and Federated Learning Perspectives. *Electronics* **2025**, *14*, 4222. https://doi.org/10.3390/electronics14214222

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction and Background

Traditional networks are static and based on the 7-layer OSI model, where the Data Link and Network layers enable networking functions. These networks comprise three planes: the management plane for monitoring, the control plane for device coordination,

and the data plane for handling traffic. Network changes must be made manually, making administration labor-intensive. In contrast, software-defined networking introduces a modern architecture that distinguishes the control plane from the data plane. A centralized controller manages network devices and directs traffic, making the network programmable and easier to manage [1,2]. This shift allows administrators to control resources more efficiently and automate network operations.

Software-Defined Networking is a revolutionary network architecture that overcomes the shortcomings of traditional networks by demultiplexing the control logic from the hardware [3]. Rather than controlling every network device separately, software-defined networking brings centralized control via software independent of the hardware devices. There are three chief layers that form the basis of software-defined networking [4,5]. (1) The Application Plane allows for instant development and instantiation of custom network services, in which developers have the ability to develop applications enhancing performance, security, and management of resources [6,7]. (2) The Control Plane is the centralized software-defined networking controller, overseeing traffic and policies [8]. It makes decisions of core forwarding devices based on global network information [9]. (3) The Data Plane includes the physical infrastructure for network communication, as shown in Figure 1 [10]. Packet forwarding in the data plane is controlled by rules that are defined by the control plane [11,12].

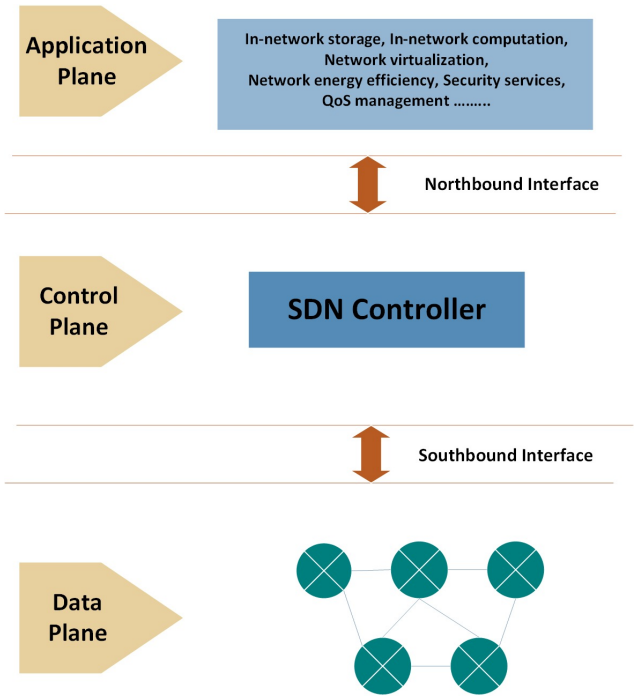


Figure 1. Software-Defined Network architecture.

Communication between these layers is performed through APIs. Southbound APIs link the software-defined networking controller to network devices [6] (e.g., routers and switches), whereas Northbound APIs help communication between the software-defined networking controller and applications [13,14]. This programmable and layered design provides more flexibility, automation, and scalability for advance network management.

Recent studies have shown that communication disruptions and partial observation losses in cyber–physical systems can weaken their resilience and compromise the effectiveness of attack detection mechanisms [15]. DDoS attacks flood applications and services, causing deviation from normal operations. These attacks use a botnet or zombie network of compromised systems to make coordinated attacks on a specific server or network

resource [16]. Through various vulnerabilities, they consume important system resources like memory, CPU, and bandwidth, essentially preventing legitimate traffic from entering the destination [17,18].

Figure 2 presents a taxonomy of DDoS attacks, which are generally classified into two primary categories. (1) Bandwidth depletion attacks (to flood the target network with malicious traffic, by blocking legitimate users from accessing network services). Its subtypes are Amplification Attack and Flood Attack. Amplification attack refers to sending requests to broadcast addresses or servers that respond with larger messages than the original request, amplifying traffic volume toward the victim, e.g., Fraggle (sending UDP packets to broadcast addresses), Smurf (sending ICMP packets to broadcast addresses), etc. Flood Attacks involve attackers using compromised devices/zombies to send massive volumes of traffic to a target system, e.g., ICMP, UDP, etc. [19,20]. (2) Resource depletion attacks (exhausting the victim system's computational, memory, or protocol-handling resources, making it incapable of processing legitimate user requests). Its subtypes are Protocol Exploitation Attack and Malformed Packet Attack. In Protocol Exploitation Attack, the attacker identifies and takes advantage of flaws/weaknesses in network protocol, e.g., TCP SYN Flood, PUSH + ACK Attack, etc. [21,22]. In Malformed Packet Attack, the attacker sends corrupted or incorrectly structured packets to confuse/crash the target, e.g., IP Address Attack (IP spoofing), IP Packet Options Field (putting unusual values in the IPv4 options field), etc.

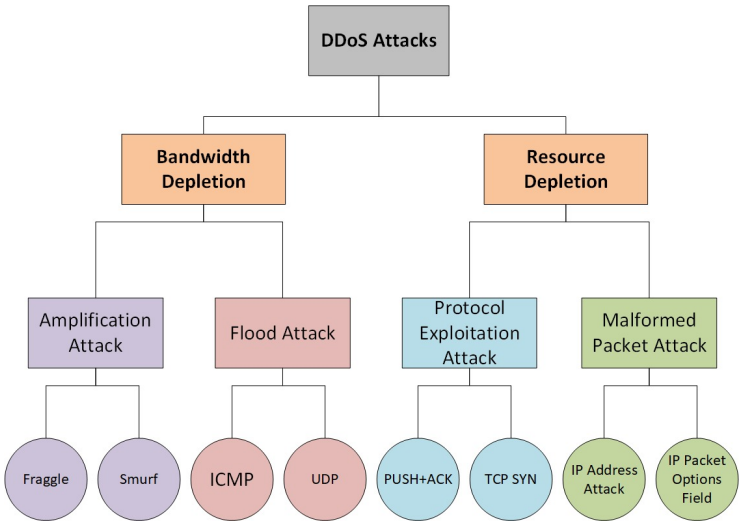


Figure 2. Taxonomy of DDoS attacks in SDN.

Machine learning (ML), deep learning (DL) and federated learning (FL) are the detection methods of DDoS attacks in SDN networks.

Through machine learning, computers learn automatically and improve themselves from experience without being programmed explicitly [23]. Machine learning is a highly effective tool for use in the domain of cybersecurity. It is capable of analyzing huge datasets and determining patterns that can reveal a cyber-attack. From past data, machine learning algorithms can learn to identify suspicious activity and anomalies in real time, enabling organizations to proactively take steps to safeguard against cyber attacks [24–26]. Machine learning entails examples like K-Nearest Neighbors (KNN) that classifies attacks according to traffic similarity, Support Vector Machine (SVM) that discriminates between benign and malicious traffic with the aid of hyperplanes, Decision Trees (DT)/Random Forest (RF) that use ensemble techniques for feature-based attack classification, Naïve Bayes (NB), which is a probabilistic approach for rapid anomaly detection, and Logistic Regression (LGR), which is the binary classification of attacks against normal traffic [1,27].

Figure 3 shows the machine learning workflow that begins with data collection. The next phase is preprocessing, which ensures that the dataset is clean and suitable for analysis. This phase often includes steps such as normalization, encoding, and, when necessary, data cleaning—which involves removing duplicate records and handling or discarding entries with missing values. The next step involves supervised learning and unsupervised learning. Supervised learning handles labeled data, while unsupervised learning treats unlabeled data. The process continues to split the data into training and testing sets by selecting an appropriate algorithm, trains the model, and then evaluates its performance on unseen data. For DDoS detection tasks, this typically involves classifying network traffic as either normal or attack. Then suitable actions such as alerts, blocking, or deploying mitigation strategies can be executed to safeguard the software-defined networking environment against identified attacks. For DDoS-specific datasets (e.g., CICDDoS2019, NSL-KDD, CAIDA), preprocessing usually entails flow reassembly to convert raw packet captures into bidirectional flow records, feature scaling (min–max or z-score normalization) to cope with skewed traffic volumes caused by attack floods, encoding categorical features such as protocol type, service, or flag values into numerical form, and removing artifacts, e.g., incomplete flows, duplicated SYN packets, or truncated attack bursts that occur during capture. These steps ensure that attack-induced anomalies (like high packet inter-arrival rates or abnormal byte counts) are kept intact as redundant data is minimized.

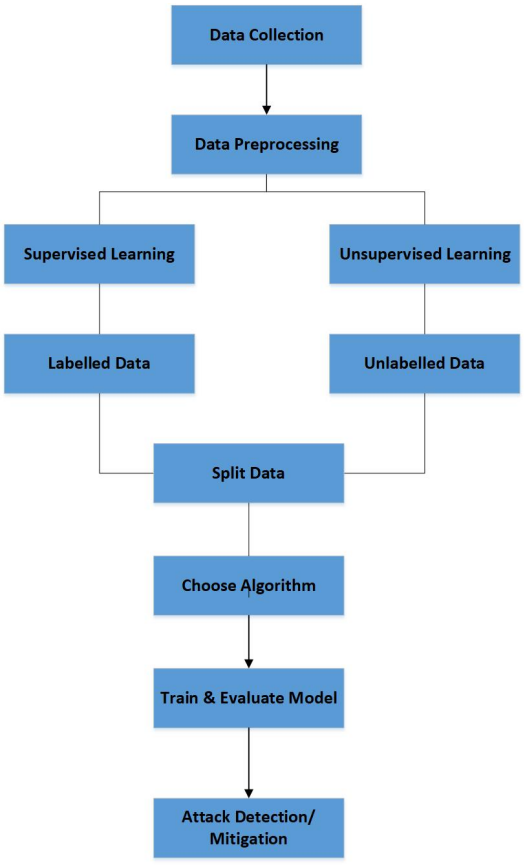


Figure 3. Machine learning workflow in cybersecurity.

Deep learning models have a high level of abstraction and learning layers [23]. These are widely utilized in the application of intrusion detection and malicious traffic identification, mostly due to their inherent benefits such as self-learning, self-organization, resistance, parallelism and excellent fault tolerance [28,29]. Examples of deep learning algorithms are Convolutional Neural Networks (CNNs), which evaluate spatial patterns within traffic data (e.g., packet headers, flow matrices), Recurrent Neural Networks (RNNs) that operate on

sequential traffic data (e.g., time-series flow statistics), Long Short-Term Memory (LSTM), an advanced form of RNN used to discover long-term dependencies within attack patterns, and Deep Neural Networks (DNNs), which comprise generic multi-layer perceptrons (MLPs) to handle high-dimensional feature learning [30,31].

Figure 4 presents a workflow of the Deep Learning method. The process begins with data collection and then the dataset is preprocessed to ensure compatibility with the learning algorithm [32,33]. Pre-processing cleans the dataset [34,35]. Unlike conventional machine learning, deep learning requires the design of a neural network architecture, which may include multiple layers like convolutional, recurrent, or fully connected layers. Then data is split into training, validation, and test sets. During training, the model learns through forward propagation and backpropagation and adjusts weights. Following training, the model is evaluated using validation or test data to assess its performance. For DDoS detection tasks, this typically involves classifying network traffic as benign or malicious and mitigating them. For DDoS-specific detection, data preparation usually requires transforming raw flow statistics into traffic matrices or time-series windows to preserve temporal dependencies of attack bursts, applying one-hot encoding for protocol or port numbers in CNNs, handling class imbalance using oversampling (SMOTE) or undersampling, and selecting features tailored to DDoS traffic like packet rate per flow, entropy of source/destination IPs, or SYN-to-ACK ratios. These considerations boost the performance of deep learning models to capture both volumetric attacks like UDP floods and stealthy low-rate attacks like slow HTTP.

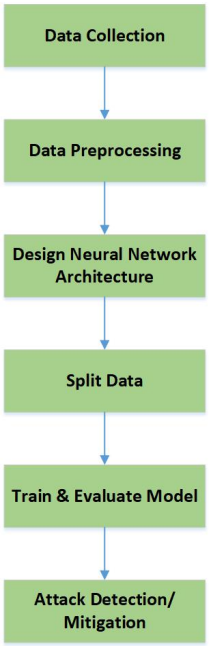


Figure 4. Deep learning workflow in cybersecurity.

In contrast to traditional machine learning approaches, where data created by the various devices needs to be collected and transported to a common location for training and processing [36], federated learning is distributed and collaborative, and it hinges on the notion of keeping data where it is created and training it there [17,37,38]. The federated learning component indicates an Aggregated Global Model, which is a centralized software-defined networking controller (or server) that collects updates from local models to improve the global detection model and distributed software-defined networking architecture, which underscores federated learning's place in scalable, privacy-conscious threat detection in geo-distributed networks [39].

Figure 5 depicts the federated learning workflow in which each local node (distributed SDN switches/controllers) collects local traffic data (packet headers, flow stats, etc.) and independently trains a local model using its own data. Instead of sharing raw data, these nodes transmit only the model updates to a central server for aggregation. The server then combines these updates to form an improved global model that reflects collective insights from all participants and then the update global model is sent back to the edge nodes for real-time DDoS detection. This process reduces the data leakage risk, preserves data privacy and minimizes bandwidth usage by avoiding the transfer of large datasets [40,41]. In DDoS-specific contexts, local nodes typically preprocess their traffic by removing redundant heartbeat packets, normalizing flow-level statistics, and filtering out incomplete TCP handshakes to preserve bandwidth before training. Aggregation strategies like FedAvg or weighted averaging are adjusted to account for heterogeneous traffic intensities across different software-defined networking domains, where some controllers may experience heavy attack loads, while others only experience normal traffic. Additionally, feature extraction at the edge often emphasizes DDoS-relevant metrics like packets-per-second, average flow duration, and source IP entropy, making sure that updates capture meaningful attack behavior. This enables federated learning-based detection to remain resilient against poisoning attacks, while reducing communication overhead and preserving local privacy.

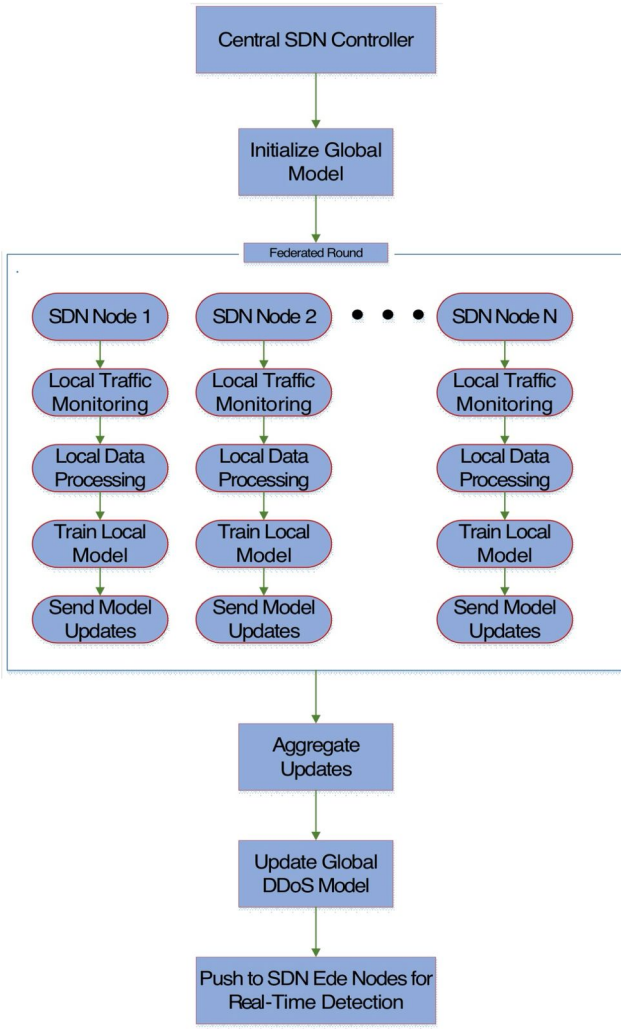


Figure 5. Flowchart of federated learning in cybersecurity.

Figure 6 illustrates the overall process of detecting and mitigating DDoS attacks in SDN. Malicious traffic first targets the SDN controller, where a defense module analyzes it using

Machine Learning, Deep Learning, and Federated Learning approaches. These methods enable the system to identify and mitigate the attack, restoring normal network traffic.

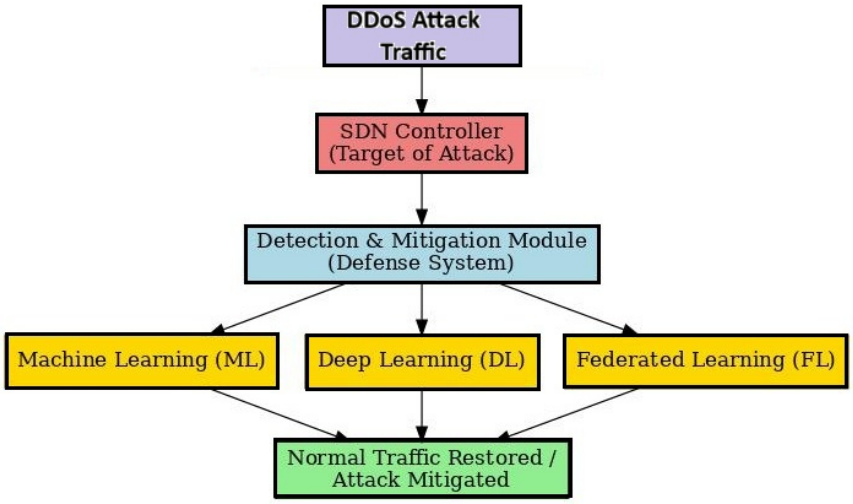


Figure 6. Conceptual framework for DDoS attack detection and mitigation in SDN using Machine Learning, Deep Learning, and Federated Learning.

The research methodology involves identify search terms, selecting sources and applying inclusion/exclusion criteria. Search terms were applied for finding DDoS attacks on software-defined networking; the core terms found were "DDoS attack", "SDN", "Distributed Denial of Service", "Software Defined Networking", "DDoS detection in SDN", "SDN-based DDoS detection techniques". Boolean operators were also applied to obtain better results: ("DDoS" OR "Distributed Denial of Service") AND ("SDN" OR "Software Defined Networking") AND ("detection" OR "prevention" OR "mitigation"), "DDoS attack" AND "SDN controller", "security challenges in SDN" AND "DDoS". Search sources are Google Scholar, SpringerLink, IEEE Xplore, ScienceDirect, ACM Digital Library, and Scopus. Recent work has been selected (2021–2025) to be on par with changing software-defined networking technology and concentrated on papers showing performance, either in simulations (e.g., Mininet) or actual implementations. Papers dealing with unrelated attacks or technologies have been removed, i.e., DNS attacks, phishing, ransomware, or attacks in conventional networks/cloud, unless specifically related to software-defined networking. Papers that are just duplicates of previous work with no significant new results or without complete content available for analysis have also been removed.

The novelty of this survey is the categorization of DDoS detection/mitigation models based on machine learning, deep learning and federated learning models into domain-specific SDN environments and general-purpose SDN environments. Their performance parameters have also been discussed. Another novel contribution is to categorize the datasets as Legacy Compatible Datasets and SDN-Contextual Datasets.

This paper has been arranged into the following order. Section 1 describes the 'Introduction and Background' in which fundamentals of software-defined networking architecture, DDoS attacks, machine learning, deep learning and federated learning architecture to identify DDoS attacks in software-defined networking and research methodology have been discussed. Then a comparative analysis of DDoS detection/mitigation approaches in domain-specific vs. general-purpose software-defined networking environment is given in Section 2. Section 3 contains a comparison of performance parameters of DDoS detection/mitigation techniques in a software-defined networking environment. Categorization of datasets as legacy compatible datasets and SDN-contextual datasets is discussed in Section 4. Finally, the future directions and conclusion have been presented in Section 5.

2. Comparative Analysis

Table 1 classifies and contrasts different machine learning techniques used in general-purpose software-defined networking settings for DDoS detection and prevention. Each row provides the algorithm(s) employed, dataset(s) utilized, strengths, limitations and simulation tools in the techniques.

Table 1. Machine Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Supervised/ Unsupervised—Reference	Algorithm	Dataset	Strength	Limitation	Simulation Tool
Supervised-[20]	Hybrid SVC-RF	Self-generated	Novel features	Computational overhead, needs tuning	Mininet, Ryu Controller
Supervised-[42]	LR, SVM, RF, KNN, XGBoost	Custom SDN	Realistic SDN attack scenarios	Synthetic traffic	Mininet, Ryu Controller, MGEN, hping3
Supervised-[43]	Ensemble Online ML (BernoulliNB, PA, SGD, MLP)	Custom SDN, CICDDoS2019, InSDN, slow-read-DDoS	Handles zero-day, low-rate attacks	High complexity due to ensemble methods	Mininet, Ryu Controller
Supervised-[44]	Hybrid Entropy + ML (SVR-RBF)	DARPA2009, RealCorp	Scalable, adaptive thresholding	Entropy-based features, manually selected attributes	Not mentioned
Supervised-[45]	Distributed ML (H2O: GBM, NBC, DRF)	CICDDoS2019	Classifies 5 attack types	Limited to CICDDoS2019 only	Mininet, CICFlowMeter
Supervised-[26]	LGR, SVM, BIRCH	Self-generated	Real-time detection of LDoS	Only LDoS covered	Mininet, Ryu Controller, TCPDump
Supervised-[46]	SVM, DT, GNB, RF, ETC	Self-generated	Real-time detection	Limited to SYN flood	Mininet, Ryu Controller
Supervised-[47]	LR, NB, KNN, SVM, DT, RF	Synthetic dataset	Real-time mitigation	Ineffective against IP spoofing	Mininet, Floodlight
Supervised-[48]	SVM, NB, MLP	Self-generated	Real-time, high accuracy with MLP	Only SYN flood evaluated	Mininet, Ryu Controller
Supervised-[49]	Rule-based classifier, SVM, cKNN	Custom simulated dataset	Detection and mitigation using policy-based automation	Mitigation limited to static policies	Mininet, hping3
Unsupervised-[50]	DenStream	Custom SDN datasets	Real-time detection	Less effective at detecting short-duration portscan attacks	Mininet, Floodlight
Unsupervised-[51]	SOM + k-NN	CAIDA "DDoS Attack 2007"	low processing time	Manual feature selection	Mininet, Floodlight

The authors in Ref. [20] develop a novel SDN-specific dataset using the Mininet emulator and Ryu controller, which capture both benign and malicious traffic. They extracted 23 flow and port-level features and label the traffic data accordingly. The core methodology involves training various machine learning classifiers, with a hybrid model achieving the best performance by combining RF and SVC. However, it has computational overhead due to the hybrid model and requires tuning for real-world deployment.

The main methodology employed in Ref. [42] to detect DDoS attacks involves a supervised machine learning framework integrated with a custom dataset. Mininet emulator was used. The authors first designed realistic SDN topologies and simulated both benign and attack-based traffic by using tools like MGEN and hping3. Network traffic was captured and processed into flow-level features. These features were then subjected to preprocessing.

Five machine learning classifiers, LR, SVM, RF, KNN and XGBoost, were trained and tested on the dataset. The proposed framework is shown in Figure 7. Performance evaluation demonstrated that the RF model achieved the highest detection accuracy with a low FAR. However, the dataset was generated using synthetic traffic, which may not fully capture real-world variability and computational overhead of hybrid models not evaluated.

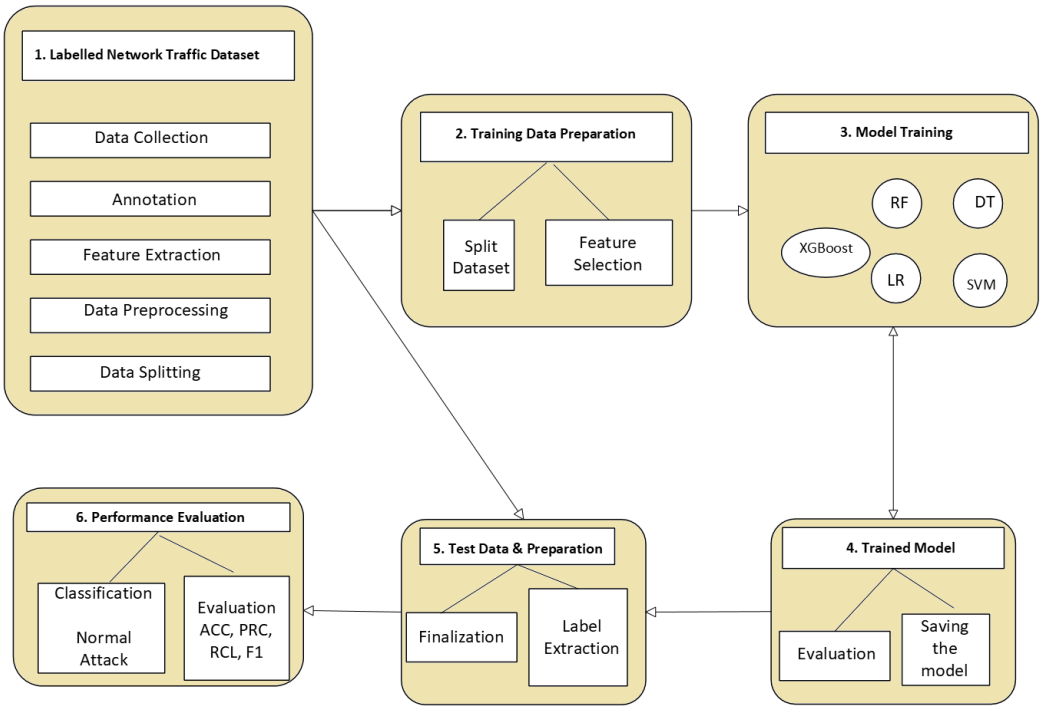


Figure 7. Proposed detection framework.

The study by Ref. [43] involves deploying a real-time, adaptive intrusion detection and prevention framework. The approach develops an ensemble of online machine learning classifiers—Bernoulli Naïve Bayes, Passive-Aggressive, SGD, and MLP—which were trained incrementally using streaming data from live network traffic which was collected via a custom-built Traffic Collector module in the Ryu software-defined networking controller. The system analyzes flow statistics continuously and dynamically selects the most suitable features using a refined Chi-squared feature selection technique. Although this architecture ensures effective handling of zero-day and low-rate DDoS attacks, it incurs higher computational complexity due to ensemble and online learning.

Previous research [44] involves a hybrid approach called ML-Entropy, which combines statistical entropy analysis with machine learning to enhance DDoS detection in a software-defined networking environment. The detection process is divided into three steps: (1) calculate the entropy of selected network traffic to detect anomalies; (2) use these entropy values as input for a SVR model with a Radial Basis Function (RBF) kernel to dynamically learn and adjust entropy thresholds; and (3) apply the learned thresholds to incoming traffic to identify and flag potential attacks. The evaluation relies on two datasets: the DARPA2009 synthetic dataset and a corporate network (RealCorp). Initial reliance on entropy-based features and manual selection of attributes could limit adaptability to emerging attack vectors.

In Ref. [45], authors introduce K-DDoS-SDN, a distributed methodology to detect DDoS attacks in software-defined networking environments, leveraging Apache Kafka and scalable machine learning. The approach comprises two modules; the NTClassification module employs H2O machine learning algorithms (e.g., Gradient Boosting Machine) trained on the Hadoop cluster using the CICDDoS2019 dataset, then deploys the optimized

model on a Kafka Streams cluster (KC-3) to identify network traffic in real time. Kafka clusters KC-1 and KC-2 preprocess incoming traffic by extracting 21 critical features (e.g., packet length, flow intervals) using CICFlowMeter, normalizes them, and replicates data to KC-3 for analysis. The NTStorage module persistently stores raw packets, network flows, and features in HDFS for model retraining and historical analysis. Evaluated across multiple attack scenarios, the system achieves high accuracy but it has limited evaluation to CICDDoS2019 and lacks diverse SDN-specific datasets.

According to Ref. [26], the machine learning-based methodology has been used to detect non-periodic Low-rate Denial of Service (LDoS) attacks in software-defined networking. The approach involves simulating both traditional and non-periodic LDoS attacks in a software-defined networking environment by using Mininet, Ryu controller, where non-periodic attacks employ random values for parameters like attack rate (R), duration (L), and period (T). Network traffic data is collected using tools like TCPDump, and key features such as TCP/UDP packet counts, standard deviation, skewness, and UDP-to-TCP ratio (UTR) are extracted. These features are preprocessed and used to train supervised (SVM, LR) and unsupervised (BIRCH) machine learning models. The LR model achieves the highest accuracy and fastest detection time. However, there is no real-world dataset validation and it focuses only on LDoS attacks (not all DDoS variants).

The experimental setup in Ref. [46] involves using the Mininet tool and Ryu controller to emulate a software-defined networking environment, where normal and attack traffic data are collected and stored to a CSV file. The machine learning algorithms, including DT, RF, GNB, SVM and ETC, are trained on this dataset, classifying traffic as normal or malicious. When an attack is detected, the Ryu controller dynamically blocks the offending port to mitigate the attack, then unblocks it after a specified time. Results showing high detection rates, particularly for the Extra Tree Classifier, which achieved high accuracy. However, limited to SYN flood attacks, emulation may not fully reflect real-world network conditions.

The methodology used in [47] involves four key modules: a Flow Collector to collect traffic data from switches, a Feature Extender to enhance native OpenFlow feature, an Anomaly Detection module using machine learning classifiers to identify flows as normal or malicious, and an Anomaly Mitigation module to block attack sources. The system was tested in a Mininet-emulated software-defined networking environment, with Random Forest achieving the highest accuracy. By extending flow features, the model improved detection performance compared to methods relying solely on native OpenFlow counters, enabling precise attack mitigation without disrupting legitimate traffic. Yet it may be ineffective in the case of IP spoofing because it relies on Ethernet address for mitigation.

The study by Ref. [48] involves simulating a software-defined networking environment by using the Ryu controller and Mininet, where normal and attack traffic flows are generated and labeled. Key flow statistics are collected and stored in a dataset. SVM, NB, and MLP (Multi-Layered Perceptron) are trained on this dataset to classify traffic. The MLP achieved the highest accuracy in real-time detection. Upon identifying an attack, the system blocks the malicious flow, leveraging SDN's programmability for rapid mitigation. Although the results illustrate the effectiveness of MLP in distinguishing and mitigating SYN flood attacks, only TCP-SYN flood attack is evaluated, and other attacks are not considered.

Authors in Ref. [49] introduce a three-stage structure to detect/mitigate DDoS attacks in software-defined networking. First, it accumulates traffic data using OpenFlow from the data plane to the control plane, where key features are quantified using Renyi's entropy. In the second stage, it frames the expected traffic status using a hybrid metaheuristic clustering algorithm combining Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), which forms clusters of normal traffic behaviors. Multinomial logistic regression is then used to classify incoming traffic based on deviations from these clusters.

Finally, the third stage implements mitigation policies that block, drop, or redirect traffic flows from suspicious IPs or ports. However, it lacks real-world dataset validation and the mitigation strategy is limited to static policies.

DenStream clustering algorithm is used in Ref. [50]. It identifies DDoS and portscan attacks in real time using entropy-based features extracted from source, destination IP addresses and ports. DenStream operates by creating micro-clusters dynamically from network traffic data, classifying new behaviors as potential anomalies based on deviations in entropy patterns. Attacks are recognized when a new potential micro-cluster (P-MC) forms within a defined "potential area" (PA) that significantly diverges from the core micro-clusters (C-MCs), which represent normal traffic. The system updates continuously without requiring labeled data or prior training and makes it suitable for evolving and high-speed software-defined networking environments. The method was tested on 48 simulated datasets with various attack intensities.

A hybrid machine learning approach used in [51] combines Self-Organizing Maps (SOM) and k-Nearest Neighbors (k-NN) to detect and mitigate DDoS attacks in a software-defined networking environment. The system comprises four key modules: statistics collection, feature extraction, attack detection, and defense. Network traffic features are collected from flow tables and preprocessed using SOM to reduce dimensionality and enhance pattern recognition. These preprocessed vectors are then classified with k-NN to determine whether an ongoing DDoS attack exists. Upon detection of attack flow, the system dynamically installs flow rules to drop malicious traffic and later restores blocked ports based on an adaptive recovery model. The system is evaluated using a regenerated CAIDA DDoS 2007 dataset.

Table 2 shows Machine Learning DDoS detection/mitigation policies in domain-specific SDN environment.

Table 2. Machine Learning DDoS detection/mitigation policies in domain-specific SDN environment.

Supervised/ Unsupervised- Domain- Reference	Algorithm	Dataset	Strength	Limitation	Simulation Tool
Supervised- IoT-[52]	NB, DT, SVM	Self-generated	Low resource usage	Limited to flooding-based attacks	Cooja simulator, Contiki
Supervised- IoT-[53]	RF, DT, GNB, BLR, kNN, SVM	Edge-IIoTset	Mitigation module for IoT nodes	Scalability concerns	Mininet-IoT, POX controller, sFlow-test and sFlow
Supervised- IoT-[54]	EV (SVM, NB, RF, kNN, LR)	Self-generated	Multilayered forwarding, real-time mitigation	Computationally heavy	Mininet dashboard Mininet, POX, Floodlight controllers, Sflow-RT
Supervised- Smart Cities-[55]	XGBoost	BoT-IoT	Binary accuracy, high multiclass precision/recall	Limited to BoT-IoT	RFID, ZigBee, sensors, and WSN
Supervised- Smart Homes-[56]	SVM, LGR, DT, KNN	UNSW-NB15, CICDDoS2019, Captured Dataset	Integrates SNORT for real-time mitigation	Limited to TCP SYN floods	Mininet, POX controller

The authors in Ref. [52] propose a feature selection mechanism using the Chi-square statistical test to identify the most relevant features from network traffic data. They then employ a Decision Tree classifier to train on these selected features for effective attack detection. The system collects flow statistics from the SDN controller and applies the trained model to identify the traffic as either benign or malicious, enabling DDoS attack identification. However, this is limited to flooding-based DDoS attacks and performance may vary with larger-scale IoT networks.

The paper in Ref. [53] introduces FMDADM, a multi-layer DDoS attack detection/mitigation framework designed for stateful SDN-based IoT networks. This process comprises four key modules: (1) an early detection module that uses a 32-packet window and entropy-based analysis for quick anomaly identification, (2) a novel Double-Check Mapping Function (DCMF) to detect spoofed IP/MAC addresses at the data plane level, (3) a machine learning-based detection module that employs feature engineering, including five newly computed features, to train and classify traffic using models like SVM, GNB, kNN, BLR, DT, and RF algorithms and (4) a mitigation module that dynamically installs high-priority flow rules to drop attack packets. Although it improves accuracy, performance may vary with larger-scale networks or different attack types.

An Adaptive Machine Learning-based SDN-enabled DDoS attack Detection and Mitigation (AMLSDM) framework has been proposed in Ref. [54], shown in Figure 8. This employs a multilayered feed-forward architecture that begins with training an ensemble of machine learning classifiers, including SVM, NB, RF, kNN, and LR. These models are aggregated using an Ensemble Voting (EV) mechanism to enhance classification accuracy. The trained model analyzes real-time traffic features such as the rate of source IPs, standard deviation of flow packets, standard deviation of flow bytes, rate of flow entries on switch, and ratio of pair-flow entries on switches. Detection occurs at the OpenFlow (OF) switches, and upon identification of a DDoS attack, mitigation is executed by the software-defined networking controller through reconfiguration of network paths and flow rules to isolate malicious traffic and maintain legitimate traffic flows. However, it requires environment-specific dataset training and computational overhead due to multiple classifiers.

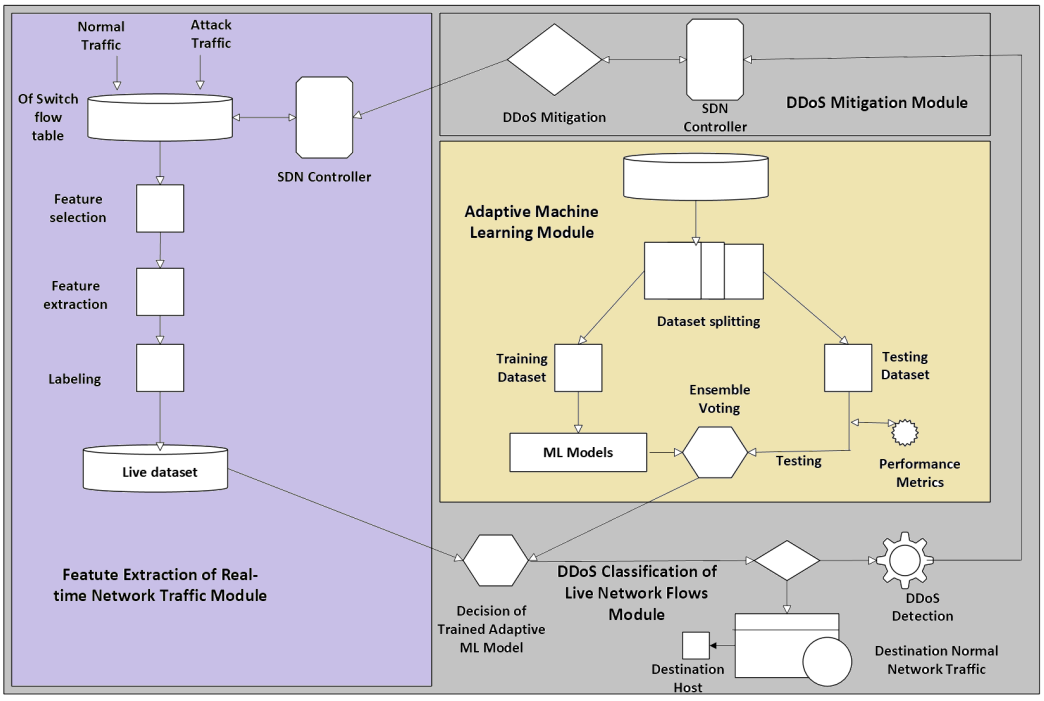


Figure 8. Proposed detection method.

The framework used in Ref. [55] operates in three layers: infrastructure, secure and intelligent software-defined networking, and service layers, leveraging SDN's centralized control to monitor and manage network traffic. XGBoost is employed for binary and multiclass classification, achieving high accuracy by analyzing features such as connection counts, packet rates, and traffic statistics to distinguish between normal and malicious traffic. The model is trained and validated using the BoT-IoT dataset to evaluate its effectiveness in mitigating DDoS attacks in smart city environments. Hence, it is limited to

the BoT-IoT dataset and may not generalize to all DDoS attack types. There is computational overhead due to XGBoost's complexity.

The main method used in [56] to detect DDoS attacks in SDN-enabled smart home networks combines signature-based detection with machine learning models. The framework employs SNORT, a signature-based intrusion detection system (IDS) for protecting the software-defined networking controller by matching traffic against predefined rules for known attacks. For smart devices, the system utilizes supervised machine learning algorithms DT, LR, SVM, and KNN trained on network traffic data to classify and detect DDoS attacks in real-time. Feature selection techniques like Principal Component Analysis (PCA) and feature selector class are applied. The framework operates within the SDN controller, leveraging its centralized control to monitor traffic and mitigating attacks dynamically. But it is limited to TCP SYN flood attacks in experiments and SNORT requires manual rule updates for new attacks.

Table 3 below shows Deep Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Table 3. Deep Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Reference	Algorithm	Dataset	Strength	Limitation	Simulation Tool
[57]	BRS-CNN	CICDDoS2019	mitigation, alerts	Scalability not explored	Mininet, POX Controller
[58]	CNN-MLP	CICDDoS2019, InSDN	Bayesian/ADAM tuning	Scalability concerns	Python, sci-kit-learn
[59]	Stacked LSTM + CNN	CICDDoS2019	Effective for low-rate/volumetric attacks	No mitigation explored	Python/Jupyter
[39]	DNN	InSDN, CICIDS2018, Kaggle DDoS	Real-time mitigation	No analysis of controller impact	Mininet, RYU, MGEN
[60]	GAN	Custom SDN Emulated Dataset, CICDDoS2019	Detects 12 DDoS types	High computational load	Mininet, Floodlight
[61]	SAE-MLP	SDN-DDoS	Hybrid models	Long processing time	Mininet
[62]	CNN	Custom, CTU-13, InSDN	Control/data plane detection	Evaluation cost, no deployment	Mininet, RYU controller
[63]	LSTM-based IDS + DRL-based IPS	Custom flow-based dataset	Detects/mitigates slow-rate DDoS	Only slow-rate attacks	Mininet, ONOS controller
[64]	LSTM-Autoencoder	InSDN, CICIDS2017, CICIDS2018	Lightweight DL model	Low performing InSDN when trained on CICIDS	Mininet, Ryu Controller
[65]	ID-CNN	Self generated	NSGA-II hyperparameter tuning, SMOTE class balancing	simulated SDN environment	Mininet, Ryu Controller
[66]	1D-CNN, GRU, DNN	CICDDoS2019, Self generated	Volumetric/low-rate DDoS attacks detection	Lacking real-time deployment	Mininet, Ryu Controller
[57]	BRS-CNN	CICDDoS2019	email alert system, real-time monitorin	Uses simulated data	Mininet, POX Controller
[67]	GAN-CL-STO	UNSW-NB15, CICIDS2017, NSL-KDD	STO feature selection, GAN data balancing	Simulation/testing offline	Python, Keras library
[68]	DBN-LSTM	CICDDoS2019	Integrated DBN for feature reduction, LSTM for temporal analysis	Offline evaluation, source code not publicly available	CICFlowMeter

The study by Ref. [57] involves a CNN-based deep learning model. The approach first collects traffic statistics from the SDN controller, then extracts flow-based features to construct a dataset. These features are normalized and reshaped into a format compatible with CNN input layers. The CNN architecture consisted of convolutional and pooling layers, which is followed by fully connected layers, and is trained to classify traffic as normal or DDoS. The model leverages its ability to automatically extract hierarchical spatial features from the data. However, the computational cost for large-scale networks has not been fully explored and reliance on pre-trained models may limit adaptability to new attack patterns.

In Ref. [58], the hybrid deep learning CNN-MLP architecture processes software-defined networking traffic features to automatically extract spatial characteristics (via CNN layers) and then classifies the traffic using fully connected MLP layers. The approach leverages flow-based traffic statistics collected from the software-defined networking controller, with preprocessing and normalization applied to the dataset. This model is trained and tested on a labeled dataset to identify DDoS attacks, demonstrating improved accuracy and performance compared to individual models. However, there are high computational complexity and potential scalability issues in large software-defined networking environments.

The proposed detection methodology in [59] is shown in Figure 9. It is a hybrid deep learning approach, combining CNN and Bidirectional Long Short-Term Memory (BiLSTM) model. Initially, network traffic data is collected and preprocessed using the CICDDoS2019 dataset. The CNN component is employed to extract local spatial features from the input traffic data, while the BiLSTM network captures temporal dependencies and contextual information from the sequence of network events. This combined architecture improves the model's ability to accurately identify patterns similar to DDoS attacks. Though limited to detection only, it does not explore real-world mitigation deployment and is only tested on one dataset.

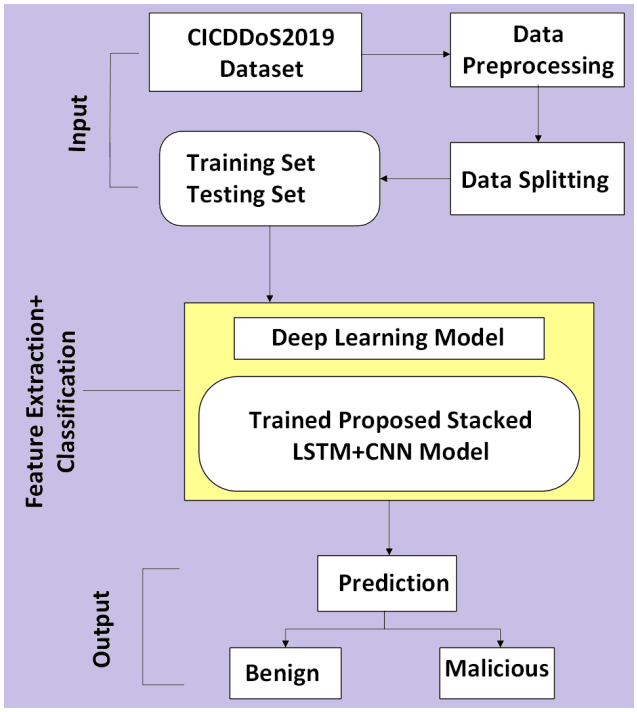


Figure 9. Proposed detection methodology.

The approach proposed in Ref. [39] involves a DNN model trained on network traffic data, which is integrated into the software-defined networking environment via the Ryu controller. The process begins with capturing real-time network traffic, preprocessing the data and feeding it into the pre-trained DNN model for classification. The model analyzes traffic patterns to differentiate between normal and malicious tasks, with a threshold-based mechanism to flag potential DDoS attacks. The model's performance is validated on datasets like InSDN, CICIDS2018, and Kaggle DDoS. Upon detection, the system mitigates the threat by blocking malicious traffic dynamically. But no mitigation mechanism is deeply evaluated beyond detection.

The authors of Ref. [60] proposed a technique based on a Generative Adversarial Network (GAN) framework. The system collects network traffic data in near-real time (every second) from OpenFlow switches, processes the data by extracting features such

as bit rates, packet rates, and entropy metrics for IP/port distributions, and feeds these features into a GAN-based anomaly detection model. The GAN consists of a generator that creates adversarial examples and a discriminator trained to distinguish between normal traffic and malicious DDoS attacks, making the system robust against adversarial perturbations. Upon detection, a mitigation module blocks malicious traffic using an Event-Condition-Action (ECA) model and a Safe List mechanism. The approach was validated on emulated software-defined networking environments and the CICDDoS 2019 dataset. Hence, adversarial training is required, which can result in high computational load.

In Ref. [61], a hybrid deep learning approach combining a Stacked Auto-Encoder (SAE) with a Multi-Layer Perceptron (MLP) is employed, termed SAE-MLP. The SAE is employed for dimensionality reduction and feature extraction, while the MLP performs the categorization of network traffic into normal or malicious classes. SDN-DDoS dataset has been used. The SAE-MLP model achieves a high accuracy, outperforming other deep learning models such as LSTM, CNN and CNN-LSTM. However, long processing time is required for SAE-MLP.

The technique used in Ref. [62] involves converting raw network traffic into RGB image data by splitting the traffic into 1 MB windows, preprocessing the data and transforming it into 3D arrays representing pixel values. These arrays are then converted into images using OpenCV2 and resized to a common dimension. A custom deep learning model based on CNN architecture is employed, consisting of multiple 2D-convolution, max-pooling, batch normalization, and dense layers, optimized with the Adam algorithm. The model classifies the images as normal or attack traffic. This approach leverages CNN's ability to automatically extract features from image-like representations of network data. However, evaluation cost is high and it lacks real-world deployment.

The architecture in Ref. [63] includes two main components: an Intrusion Detection System (IDS) based on LSTM neural networks, and a Deep Reinforcement Learning (DRL)-based Intrusion Prevention System (IPS). Traffic is monitored at the software-defined networking data plane via edge switch mirroring, processed using CICFlowMeter to generate bidirectional flow statistics, and sampled for lightweight processing. The LSTM-based IDS detects suspicious flows and informs the DRL-based IPS, which uses a deep Q-learning agent per bidirectional connection to make mitigation decisions in real time. However, it focuses only on slow-rate attacks (e.g., application layer).

Authors in Ref. [64] combined feature selection techniques with a deep learning model based on an LSTM-Autoencoder architecture. The method uses IG and RF to identify the most relevant SDN-specific traffic features from datasets including InSDN, CICIDS2017, and CICIDS2018. After preprocessing and normalization, the selected features are put into the deep learning model, where an unsupervised autoencoder extracts hierarchical representations, followed by supervised fine-tuning using LSTM layers to capture temporal traffic behavior. The model is merged into the software-defined networking controller and evaluated using the Cbench tool for performance impact. Although it achieves high accuracy, there is limited generalization across datasets and it is not deployed in a live software-defined networking environment.

The authors in Ref. [65] first generate a custom SDN-specific dataset in Mininet by launching ICMP, UDP, and TCP flooding attacks. A Ryu controller monitors OpenFlow statistics, which are logged and labeled to form the dataset. To address data imbalance, SMOTE is applied at the protocol level, and the model is trained using TensorFlow/Keras. The core model is a stacked 1D-CNN, enhanced with early stopping to prevent overfitting. NSGA-II, a multi-objective genetic algorithm, is employed to tune seven critical hyperparameters, optimizing for both accuracy and training time. Although this achieves high accuracy, its evaluation is limited to a simulated software-defined networking environment.

The paper in [66] proposes a hybrid deep learning model combining a 1D CNN, Gated Recurrent Unit (GRU), and Dense Neural Network (DNN) for detecting DDoS attacks in software-defined networking. This architecture is designed to capture both spatial and temporal patterns in network traffic, and is effective for identifying both high-rate and low-rate DDoS attacks. The model is trained and evaluated on both the CICDDoS2019 dataset and a custom-generated SDN dataset. The CNN extracts short-term features, the GRU captures long-term dependencies, and the DNN performs classification. But there is a lack of real-time deployment in a live SDN.

The CNN in Ref. [57] is used for both binary and multi-class classification of DDoS attack types. It employs pre-processing methods including feature elimination and information gain-based selection to enhance performance. Once the software-defined networking controller is deployed, the model monitors flow statistics from OpenFlow switches in real time, classifies traffic, and applies mitigation strategies if an attack is detected. An integrated monitoring module oversees blocked IPs to reduce false positives and ensure continued access for legitimate traffic. Additionally, a base64-encoded email alert system notifies administrators upon attack detection. However, simulated data is not validated in production networks and IP spoofing mitigation could cause false blocking.

The approach in Ref. [67] combines a 1D CNN and LSTM network, with the Siberian Tiger Optimization (STO) algorithm used for both optimal feature selection and hyperparameter tuning. To tackle dataset imbalance, a GAN is used to synthesize realistic attack samples. The processed and dimensionally reduced features are passed through the CNN-LSTM pipeline for classification for secure and efficient sharing of blacklists among distributed software-defined networking controllers. However, simulation and testing are not performed in a real software-defined networking environment and steganography for blacklist exchange is not benchmarked for speed or scalability.

An adversarial DDoS detection framework for software-defined networking is presented in Ref. [68] using a hybrid deep learning model that combines a Deep Belief Network (DBN) and LSTM network. The system incorporates GANs to generate perturbed adversarial samples, which are added to the training set. The DBN module performs dimensionality reduction on 88 flow features (extracted using CICFlowMeter), while the LSTM captures temporal dependencies in the flow data. Together, both models detect normal and adversarial DDoS traffic. The detection module is integrated into a software-defined networking application layer with a mitigation system based on Event-Condition-Action (ECA) rules, allowing automated packet drops at the controller level. However, no real software-defined networking testbed is used and evaluation is purely offline. Source code is also not publicly available.

Table 4 pivots its attention to deep learning DDoS detection/mitigation policies in domain-specific SDN deployments like IoT, Cloud, Edge Computing and Vehicular Networks where network requirements and constraints are quite different.

Table 4. Deep Learning DDoS detection/mitigation policies in domain-specific SDN environment.

Domain- Reference	Algorithm	Dataset	Strength	Limitation	Simulation Tool
Cloud [69]	MMEDRL-ADM	SDN-specific dataset	Metaheuristics optimize tuning	High computational complexity	Mininet, RYU controller
Vehicular [70]	OptTabNet-Adam	SDN-DDoS, InSDN	Interpretability, adaptive learning rate	Lacks adversarial testing	Mininet, RYU
IoT [71]	LSTM	Edge-IIoT	Detects low-rate DDoS	Only one dataset, LSTM is resource-intensive	SDN-IoT testbed, ONOS

Table 4. *Cont.*

Domain- Reference	Algorithm	Dataset	Strength	Limitation	Simulation Tool
Edge Computing [72]	LSTM	CIC-DDoS2019, CIC-IDS2017, UNSW_NB15, CSE-CIC-IDS2018, ISCX-IDS-2012	Supports SDN, high precision and accuracy	Does not represent real-world DDoS scenarios	Python 3.6 with PyTorch 1.6
IoT [73]	FeedForward–CNN (FFCNN)	CIC DoS 2017	7 features via iterative feature selection	No real-time deployment in SDN environment	Python, TensorFlow 2.5.0
Vehicle [74]	SSAE- Softmax Classifier	Self generated	decreases model training time	Lacking real-time responsiveness	MATLAB

The paper in [69] involves the Metaheuristic with Multi-Layer Ensemble Deep Reinforcement Learning for DDoS Attack Detection and Mitigation (MMEDRL-ADM) technique. The approach begins with preprocessing network data to clean and transform it, followed by African Buffalo Optimization-based Feature Selection (ABO-FS) to reduce computational complexity and enhance detection accuracy. The core detection mechanism employs a Multi-Layer Ensemble Deep Reinforcement Learning (MEDRL) model, which combines deep learning and reinforcement learning to classify attacks. To optimize performance, an Improved Grasshopper Optimization Algorithm (IGOA) fine-tunes the hyperparameters of the MEDRL model. The system is tested on an SDN-specific dataset. However, it has computational complexity due to ensemble and optimization layers.

The study by Ref. [70] proposed the technique for detection of DDoS attacks in SDN-based VANETs, shown in Figure 10. It involves an optimized deep learning model called TabNet, enhanced with Adam optimization and hyperparameter tuning by using Grid Search Cross-Validation (GSCV). This approach processes tabular network traffic data, leveraging TabNet's feature selection and sequential decision-making capabilities to identify attack patterns. The model dynamically adjusts learning rates and employs limited attention mechanisms to focus on suitable features and achieve high accuracy. However, datasets might not fully represent real-world VANET diversity and limited adversarial testing.

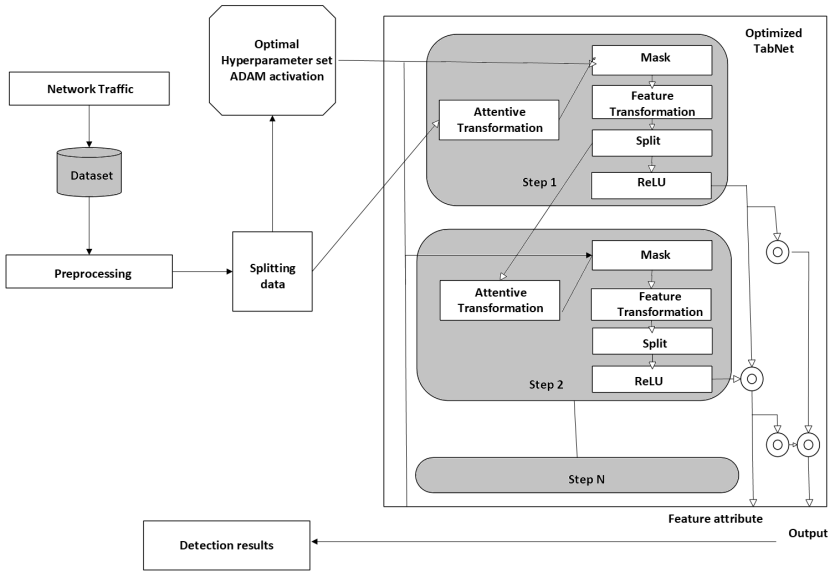


Figure 10. Proposed methodology.

The work in Ref. [71] introduces the technique for detecting low-rate DDoS (LDDoS) attacks in SDN-enabled IoT networks based on LSTM. The approach involves collecting and preprocessing network traffic data from the Edge-IIoTset dataset, normalizing features,

and training the LSTM model with an Adam optimizer and dropout regularization to prevent overfitting. The model analyzes time-series traffic patterns to distinguish LDDoS attacks from normal traffic, achieving accuracy. Key steps include feature extraction from software-defined networking controller logs, dynamic thresholding, and real-time prediction to mitigate malicious flows, outperforming traditional methods like SVM and decision trees in IoT-specific software-defined networking environments. However, only one dataset is evaluated, and LSTM training requires high resources.

Authors in Ref. [72] proposed CoWatch, a collaborative framework for predicting and detecting DDoS attacks in edge computing (EC) using distributed SDN. An optimal threshold model was introduced to balance collaboration efficiency and prediction accuracy, effectively filtering suspicious flows. A collaborative prediction algorithm based on LSTM was designed, and experiments on five datasets with three attack types showed CoWatch achieved high precision and accuracy. However the paper relies on limited attack types and public datasets, which may not fully represent real-world DDoS scenarios.

The paper in [73] proposes an intrusion detection system (anomaly-based) which is called FeedForward–Convolutional Neural Network (FFCNN) to detect LR-DoS attacks in IoT-enabled software-defined networking environments. The system uses hybrid deep learning that combines a feedforward neural network and 2D convolutional layers to deeply mine traffic flow features. Using the CIC DoS 2017 dataset, the raw packets are first converted to bidirectional flow statistics by using the CIC Flow Meter. The authors perform iterative SVM-based feature selection, reducing the feature set to seven critical attributes to speed up detection. FFCNN is trained on part of the data to establish a threshold for benign flows, and then tested against unseen attack traces. However, there is no real-time deployment in a software-defined networking environment and it is evaluated in offline mode on CIC data.

DDoS attack detection in software-defined networking-based VANET networks was carried out in Ref. [74] by using SSAE and Softmax classifier deep network models. Communication time between the the vehicles and main controller was increased using edge controller. Four different models of SSAE and Softmax classifier deep network were designed, i.e., single-, two-, three- and four-layer models. Higher accuracy was achieved via the four-layer SSAE and Softmax classifier deep network model. However, real-time responsiveness is required in VANET-based SDN systems.

Table 5 shows Federated Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Table 5. Federated Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Reference	Algorithm	Dataset	Strength	Limitation	Simulation Tool
[75]	FedLAD (Federated Learning with XGBoost)	CICDoS2017, CICDDoS2019, InSDN	Scalable, preserves private data	Vulnerable to data poisoning	Mininet, Ryu Controller
[17]	FL (DNN, CNN, LSTM)	CICIDS2019	Near-centralized high accuracy, privacy-preserving	Limited attack types evaluated	Mininet, Ryu Controller

The technique used in Ref. [75] is about the Federated Learning-based approach named "FedLAD," which leverages a decentralized machine learning framework to train models across multiple local controllers without exchanging raw data. Each local controller monitors network traffic, collects local data, and trains a model using the XGBoost algorithm, which is then aggregated by a root controller to make a global model. This hierarchical structure reduces controller overhead and improves scalability, achieving high detection accuracy while preserving data privacy. The system was evaluated using datasets

CICDoS2017, CICDDoS2019 and InSDN. As the model accuracy depends on aggregation technique, there is potential vulnerability to data poisoning threats.

The federated learning-based approach shown in Figure 10 has been used in Ref. [17]. Three deep learning classifiers, DNN, CNN, and LSTM, are trained locally on distributed SDN controllers using the CICIDS 2019 dataset. Instead of centralizing data, each controller trains its model on local traffic data and shares only the model parameters with a central server, which aggregates them by using the FedAvg algorithm to create a global model. Although the decentralized approach achieves high detection accuracy, evaluation is limited to only three DDoS attack types, i.e., UDP Flood, TCP SYN and DNS Flood, while relying solely on the CICIDS 2019 dataset.

Table 6 contains different Federated Learning DDoS detection/mitigation policies in a domain-specific SDN environment.

Table 6. Federated Learning DDoS detection/mitigation policies in domain-specific SDN environment.

Domain- Reference	Algorithm	Dataset	Strength	Limitation	Simulation Tool
IoT [76]	WFL-ANN	CAIDA	Privacy-preserving, lightweight	Dependent on CAIDA dataset, empirical weight tuning	MATLAB
IoT [41]	FL-DAD-CNN	CICIDS2017	scalable, privacy ensured	High aggregation cost, integration complexity	TensorFlow Federated, Keras
Industrial IoT [77]	FL (CNN-MLP)	CICDDoS2019	Fast execution, privacy-preserving	Needs broader IIoT testing	Filter-based PCC

The research in Ref. [76] presents a Weighted Federated Learning (WFL)-based approach to detect LR-DDoS attacks in SDN-controlled IoT networks. The method involves training local Artificial Neural Network (ANN) models using three algorithms, Bayesian Regularization (BR), Levenberg–Marquardt (LM) and Scaled Conjugate Gradient (SCG), on distributed edge devices. These locally trained models share their optimized weights with a federated server, which aggregates them by assigning dynamic preferences based on each model's accuracy. The federated server then forms a global model, which is validated and redistributed to edge devices for real-time attack detection. But dependency on the CAIDA dataset may not generalize to all IoT/SDN scenarios, with a preference for weights assigned via hit-and-trial, lacking theoretical optimization.

Federated Learning for Decentralized DDoS Attack Detection (FL-DAD) is introduced in Ref. [41], which employs a Convolutional Neural Network (CNN) trained in a federated learning framework, where local models are trained on distributed IoT devices without centralizing raw data. The federated learning process in IoT networks is shown in Figure 11. Model updates are aggregated at a central server to create a global model, ensuring data privacy and reducing communication overhead. The system was evaluated using the CICIDS2017 dataset, demonstrating high accuracy. However, it has high computational load during aggregation with nodes, causing integration challenges.

According to Ref. [77], the low-complexity CNN-MLP model is combined with filter-based Pearson Correlation Coefficient (PCC) feature selection technique. The system trains local models on distributed software-defined networking controllers (industrial agents) using private traffic data, then aggregates model updates via a central server using the Federated Averaging (FedAvg) algorithm to preserve data privacy. The CNN-MLP architecture employs residual connections and factorized convolutions to enhance feature extraction and mitigate gradient vanishing, while the PCC technique reduces feature dimensionality by selecting highly correlated features. Evaluated on the CICDDoS2019 dataset, FedDDoS achieves accuracy and low computational time in classifying DNS, UDP, and SYN flooding attacks. However, additional verification is required across varied IIoT environments.

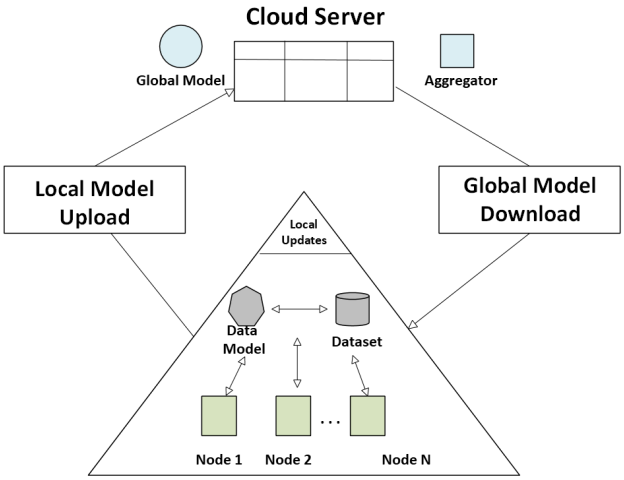


Figure 11. Federated learning process in IoT networks.

The surveyed literature indicates that machine learning, deep learning, and federated learning approaches hold significant potential for countering DDoS threats in both general-purpose and SDN-specific contexts. Supervised algorithms generally deliver strong detection accuracy with relatively straightforward designs; however, they often depend on artificially generated or narrow datasets, which restrict their ability to generalize to real-world traffic, and they tend to be optimized for only a limited range of attacks such as SYN floods or low-rate DoS. Unsupervised strategies lessen the reliance on labeled data and can adapt to dynamic traffic conditions, but they are prone to higher false alarm rates and struggle with subtle or stealthy attacks. Deep learning-based architectures including CNNs, LSTMs, GANs, and hybrid models excel at identifying spatial–temporal traffic patterns and consistently outperform traditional methods; though their computational cost is high and large-scale, real-time deployment in SDN testbeds remains rare. Efforts targeted at IoT, VANETs, cloud, and edge networks underscore the versatility of machine learning/deep learning, yet these solutions are still limited by heterogeneous datasets and the resource constraints of such environments. Recently, federated learning-based techniques have been explored for their scalability and privacy-preserving capabilities, achieved by distributing training across SDN controllers. Nevertheless, these methods are still at an early stage, facing issues such as vulnerability to poisoning, expensive aggregation, and insufficient validation under diverse attack settings. Overall, the common gaps across current studies include heavy dependence on small or synthetic datasets, the absence of standardized SDN benchmarks, limited evaluation in adversarial or real-time conditions, and a lack of lightweight, adaptive, and deployment-ready solutions suitable for production SDN systems.

3. Performance Parameters

Performance parameters help assess the effectiveness, efficiency, and practicality of the proposed models across various network scenarios [78]. They offer insights into how well a technique detects attacks, minimizes errors, scales with traffic, preserves privacy, and integrates into real-world software-defined networking infrastructures. Table 7 summarizes the core performance parameters, their descriptions, and references from the reviewed literature.

Table 7. Key parameters and descriptions.

Parameter	Description	References
Accuracy	Measures the percentage of correctly predicted traffic (benign vs. malicious).	[17,20,57]
False Alarm Rate (FAR)	Rate at which benign traffic is incorrectly classified as malicious traffic.	[47,59,79]
Detection Time/Latency	Time taken to detect an attack after it starts.	[26,46,54]
Computational Overhead	Resource consumption (CPU, memory) introduced by the detection mechanism.	[20,53,58]
Scalability	Ability to handle large traffic volumes or operate in large networks without performance degradation.	[17,44,76]
Adaptability	Ability to detect new, unseen attack types (zero-day).	[26,43,60]
Model Robustness	Resistance to adversarial attacks, poisoning attempts, or errors in training data.	[17,75,76]
Privacy Preservation	Ensures sensitive data isn't exposed, only updates are shared.	[17,41,75]
Detection Rate/Recall	Percentage of actual attacks correctly identified.	[55,59,60]
Precision	Ratio of true positive detections to all positive detections made.	[20,41,45]
F1-Score	Harmonic mean of recall and precision, balancing false negatives and false positives.	[39,53,76]
Mitigation Effectiveness	Ability to stop or neutralize the DDoS attack once detected.	[54,56,57]
Real-time Capability	Indicates whether detection and response are within operational timeframes.	[46,56,57]
Resource Utilization	Efficient use of controller memory, bandwidth and processing power.	[44,45,53]
Generalization	Performance consistency across environments or attack types.	[56,58,70]
Integration Complexity	Difficulty of integrating the solution into existing SDN setups and controllers.	[20,56,59]
Energy Efficiency	Model's power consumption, especially for IoT and edge environments.	[41,53,76]

Comparison of Common Performance Parameters

Some of the common performance parameters have been used to compare the techniques of DDoS detection/mitigation in a -defined networking environment in Tables 8–13.

In addition to presenting results, some broader insights can be drawn. Machine learning techniques in software-defined networking often perform well when feature engineering is tailored to specific attack types, but they tend to struggle with unseen or zero-day attacks. Deep learning approaches, such as CNN and LSTM, generally outperform traditional machine learning by automatically extracting temporal and spatial features from flows, though at the cost of higher computation. Federated learning introduces an advantage in preserving privacy and distributing the load, but its effectiveness depends heavily on aggregation strategies and resistance to poisoning. Notably, no single category dominates across all scenarios: machine learning is lightweight and suited for real-time applications with constrained resources, deep learning provides superior accuracy in complex traffic environments, and federated learning addresses data-sharing constraints but still requires stronger robustness in adversarial conditions.

Table 8. Performance parameters of Machine Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Supervised/ Unsupervised—Reference	Algorithm	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Supervised-[20]	Hybrid SVC-RF	Self-generated	98.8	98.27	97.07	97.65
Supervised-[42]	RF	Custom SDN	98.97	98.33	96.37	97.34
Supervised-[43]	Ensemble	Custom SDN	99.26	99.10	99.60	98.17
Supervised-[44]	RF	RealCorp	99.93	99.88	100	99.94
Supervised-[45]	Gradient Boosting Machine (GBM)	CICDDoS2019	99.22	96.00	99.00	99.00
Supervised-[26]	LGR	Self-generated	99.92	99.93	99.91	99.92
Supervised-[46]	Extra Tree Classifier (ETC)	Self-generated	100	100	100	100
Supervised-[47]	DT	Synthetic dataset	86.74	99.62	54.27	99.91
Supervised-[48]	MLP	Self-generated	99.75	99.59	100	99.79
Supervised-[49]	Rule-based classifier	Custom simulated dataset	96.20	100	75.00	85.70
Unsupervised-[50]	DenStream	Custom SDN datasets	98.00	98.00	99.00	100
Unsupervised-[51]	SOM + kNN	CAIDA "DDoS Attack 2007"	98.90	–	98.70	–

It is also important to reflect on unusually high reported performance values, such as the 100% accuracy result in Table 8. Perfect detection is rarely achievable in real-world SDN networks and often indicates evaluation on limited or imbalanced datasets, overfitting to training conditions, or insufficient adversarial testing. While such results demonstrate the potential of optimized models in controlled environments, they should be interpreted with caution, as they may not generalize to noisy, heterogeneous, and evolving traffic patterns. Future work should validate these claims using diverse datasets and adversarial stress testing to ensure robustness.

Table 9. Performance parameters of Machine Learning DDoS detection/mitigation policies in domain-specific SDN environment.

Supervised/ Unsupervised–Domain–Reference	Algorithm	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Supervised–IoT–[52]	DT	Self-generated	98.10	–	–	–
Supervised–IoT–[53]	RF	Edge-IIoTset	99.79	99.09	99.77	99.43
Supervised–IoT–[54]	Ensemble Voting (SVM, NB, RF, kNN, LR, EV) called AMLSDM-EV	Self-generated	98.80	98.30	96.30	95.20
Supervised–Smart Cities–[55]	XGBoost	BoT-IoT	99.90	97.00	99.00	98.00
Supervised–Smart Homes–[56]	DT	Captured Dataset	99.57	100.00	100.00	100.00

Table 10. Performance parameters of Deep Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Reference	Algorithm	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
[57]	BRS-CNN	CICDDoS2019	98.64	99.00	99.00	99.00
[58]	CNN-MLP	CICDDoS2019, InSDN	99.90	99.90	99.90	99.90
[59]	Stacked LSTM + CNN	CICDDoS2019	99.76	99.64	100.00	–
[39]	DNN	CICIDS2018	100.00	–	–	–
[60]	GAN	CICDDoS2019	99.78	99.76	99.99	99.87
[61]	SAE-MLP	SDN-DDoS	99.75	–	–	–
[62]	CNN	Custom	90.69	92.06	90.69	90.62
[63]	Deep Reinforcement Learning (DRL)-based IPS	Custom flow-based dataset	98.00	98.00	–	–
[64]	LSTM-Autoencoder	CICIDS2018	99.64	99.00	100.00	99.00
[65]	ID-CNN	Self-generated	99.00	99.00	–	99.00
[66]	GRU	Self-generated	99.80	99.90	99.90	99.90
[57]	BRS-CNN	CICDDoS2019	98.64	99.00	99.00	99.00
[67]	GAN-CL-STO	NSL-KDD	99.91	–	–	–
[68]	DBN-LSTM	CICDDoS2019	96.50	96.40	98.50	97.40

Table 11. Performance parameters of Deep Learning DDoS detection/mitigation policies in domain-specific SDN environment.

Domain–Reference	Algorithm	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Cloud [69]	MMEDRL-ADM	SDN-specific dataset	99.00	99.00	99.00	98.50
Vehicular [70]	OptTabNet-Adam	SDN-DDoS	99.41	98.50	99.90	99.20
IoT [71]	LSTM	Edge-IIoT	98.90	97.40	96.60	96.90
Edge Computing [72]	LSTM	CICDDoS2019	93.30	97.98	91.01	94.36
IoT [73]	FeedForward–Convolutional Neural Network (FFCNN)	CICDoS2017	99.00	97.50	96.30	96.90
Vehicular [74]	Four-layer (SSAE + Softmax Classifier)	Self-generated	96.90	96.22	–	96.19

Table 12. Performance parameters of Federated Learning DDoS detection/mitigation policies in general-purpose SDN environment.

Reference	Algorithm	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
[75]	FedLAD (Federated Learning with XGBoost)	CICDoS2017	99.92	99.95	99.95	99.97
[17]	FL-LSTM	CICIDS2019	99.99	100.00	99.99	99.99

Table 13. Performance parameters of Federated Learning DDoS detection/mitigation policies in domain-specific SDN environment.

Domain–Reference	Algorithm	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
IoT [76]	WFL-ANN	CAIDA	98.85	–	–	94.21
IoT [41]	FL-DAD-CNN	CICIDS2017	98.70	98.80	98.90	98.90
Industrial IoT [77]	FL (CNN-MLP)	CICDDoS2019	98.37	–	–	–

4. Datasets

A publicly available dataset plays a necessary role for the advancement of research. These are a shared foundation which enable researchers to enhance the reproducibility by utilizing the existing knowledge. Various public datasets for DDoS detection have been surveyed. Datasets can be categorized into two broader categories.

4.1. Legacy Compatible Datasets

Table 14 contains the datasets which do not focus on software-defined networking. They contain features from legacy networks. They do not have SDN-specific features (like OpenFlow messages or controller logs).

Table 14. Legacy compatible datasets.

Dataset Name	Year	No. of Features	Source/Repository	References
DARPA	1998–2009	(1998, 1999) 41; (2000, 2009) Not specified	MIT Lincoln Laboratory	[80–83]
KDD-CUP99	1999	41	Lincoln Laboratory	[84]
MAWI	2001+	Not specified	MAWI Working Group	[85]
Kyoto 2006+	2006+	24	Kyoto University	[86]
CAIDA	2007, 2016	Not specified	CAIDA	[87,88]
NSL-KDD	2009	41	University of New Brunswick (CIC)	[89,90]
CTU-13	2011	14	Czech Polytechnic University	[91]
ISCX2012	2012	20	Canadian Institute for Cybersecurity (CIC)	[92]
UNSW-NB15	2015	49	Australian centre for Cyber Security (ACCS)	[93]
CICIDS 2017	2017	80	Canadian Institute for Cybersecurity (CIC)	[94,95]
CSE-CIC-IDS 2018	2018	80	CSE and CIC	[96]
BOT-IOT	2018	35	UNSW Canberra IoT Security Lab	[97]
CICDDoS2019	2019	Not specified	Canadian Institute for Cybersecurity (CIC)	[98]
TON-IOT	2020	44	UNSW Canberra IoT Security Lab	[99]

Description

DARPA Dataset [80–83]: Lincoln Laboratory developed the DARPA intrusion detection datasets in 1998 and 1999 using simulated network environments to evaluate IDS performance. The dataset included both real-time and offline components, with batch-mode analysis of network traffic and audit logs. It featured typical services such as IRC, email, FTP, and Telnet, along with attack types like rootkits, DoS, buffer overflows, and Nmap scans. However, it lacked realism, suffered from outdated attack patterns, and included few false positives, limiting its relevance to modern networks. In 1999, DARPA 2000 extended this effort with a five-week dataset featuring two attack scenarios, i.e., LLDoS 1.0 and LLDoS 2.0.2. They cover DoS, U2R, R2L, probe attacks, and data exfiltration. Later, the DARPA 2009 dataset introduced more realistic, synthesized traffic between a/16 subnet

and the Internet. Collected over 10 days, it included 7000 pcap files (totaling 6.5 TB) and covered modern threats such as DDoS attacks and worms, simulating more diverse and contemporary attack behaviors.

KDD-CUP99 Dataset [84]: The KDD'99 dataset was developed at MIT's Lincoln Laboratory between 1998 and 1999 using DARPA packet trace data. Collected over a five-week period within a simulated military network environment, this dataset is provided in tcpdump format and comprises 41 traffic features grouped into three categories: basic (fundamental), traffic-based and content-based characteristics. Beyond standard network statistics, it includes labeled attack types classified into four main types: User to Root (U2R), Denial of Service (DoS), Remote to Local (R2L) and probing attacks. It has a challenge of redundancy, i.e., approximately 78% of training data and 75% of testing data consisting of duplicate records. This variance heavily biases detection algorithms toward frequent, high-volume attack types like DoS, and significantly reduces detection performance for less frequent but critical attacks such as U2R and R2L.

MAWI Dataset [85]: The MAWI (Measurement and Analysis on the WIDE Internet) dataset contains real-world traffic traces collected from the WIDE backbone, a major Japanese Internet exchange point. It serves as a valuable resource for analyzing large-scale network behavior, including traffic pattern recognition and the characterization of DDoS attacks.

Kyoto Dataset [86]: The Kyoto 2006+ dataset was developed using a honeypot at Kyoto University, capturing real network traffic from November 2006 to August 2009. It includes 24 statistical features, 14 from the KDD Cup99 dataset and 10 additional ones. To improve realism, normal traffic was generated alongside attack traffic using a secondary server in the same honeypot network. However, the dataset has several limitations. (1) Most traffic is malicious, with normal traffic making up only 3–4% (limited to email and DNS traces). (2) The lack of labeled attack types skews intrusion detection evaluations. (3) Normal and malicious traffic was generated separately, creating an uncorrelated dataset. Despite using real-world traffic, the absence of attack details makes it difficult to assess their impact on SDN infrastructure.

CAIDA Dataset [87,88]: Cooperative Association for Internet Data Analysis (CAIDA) has developed several network traffic datasets for research. The CAIDA OC48 dataset captures around 100 GB of raw traffic observed over an OC48 link in San Jose. The CAIDA DDoS dataset provides one hour of DDoS attack traffic, segmented into five-minute pcap files. Additionally, the 2016 CAIDA Internet Traces dataset offers passive traffic data collected via the Equinix-Chicago monitor on high-speed infrastructure. While these datasets are valuable for analyzing specific attack scenarios, their heavy anonymization, masking protocol, payload, and destination data limits their usefulness as comprehensive benchmarks for intrusion detection research.

NSL-KDD Dataset [89,90]: The NSL-KDD dataset was introduced as an updated version of the original KDDCup'99 dataset by removing redundant/duplicate records from both training and testing sets and is divided into two segments: training and testing. Notably, the test set contains a wider range of attack types than the training set, with approximately 17 attack types present in testing but absent from training. Both the KDD'99 and NSL-KDD datasets were developed two decades ago and are now outdated, no longer accurately representing modern network traffic or current attack vectors. Additionally, many of the features included in these datasets are irrelevant for SDN environments. For example, previous studies using NSL-KDD in software-defined networking contexts often relied on only six of the original 41 features. Detecting sophisticated attacks, i.e., User to Root (U2R) and Remote to Local (R2L), requires deeper packet inspection and content-based features,

which are limited or absent in NSL-KDD. As a result, the NSL-KDD dataset also falls short in representing realistic, dynamic network traffic found in modern systems.

The CTU-13 dataset [91]: The CTU-13 dataset, developed by Czech Technical University in 2011, contains real network traffic from 13 distinct botnet scenarios. Each scenario involves the execution of specific malware using various protocols to simulate different botnet behaviors, with corresponding pcap files. Scenario 10 captures UDP-based DDoS traffic, while Scenario 11 records ICMP-based DDoS activity. This dataset provides a valuable resource for analyzing botnet-related threats in diverse operational contexts.

ISCX2012 Dataset [92,94]: The ISCX dataset, developed by the Canadian Cyber Security Institute, consists of real-world network traffic generated in a simulated environment. Traffic was produced using two profiles: Alpha for attack traffic and Beta for normal traffic. It includes full-payload data for protocols such as IMAP, SMTP, HTTP, SSH, FTP, and POP3, along with 20 packet-level features focused on two attack types—DoS and brute force. However, the dataset presents limited diversity in DoS attacks and lacks representation across multiple OSI layers. Additionally, its exclusive use of HTTP traffic limits its relevance, as modern network traffic predominantly uses HTTPS.

UNSW-NB15 Dataset [93]: The UNSW-NB15 dataset was developed in 2015 by the Australian Centre for Cyber Security (ACCS). tcpdump was used to capture traffic and processed with Argus and Bro tools later. It extracted 49 features, including flow-based, time-based and content-based attributes. It is widely used because it reflects realistic network behavior.

CICIDS2017 Dataset [94,95]: The CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity in 2017, contains both real attack scenarios and normal traffic flows. Network data was captured and analyzed using CICFlowMeter, which extracts flow-based features from source/destination IPs, protocols, and traffic records. The dataset includes a variety of common attack types, such as Brute Force, Heartbleed, Botnet, DDoS, DoS, Web-based, and Infiltration attacks. It features 80 flow-based attributes designed to reflect real-world traffic behavior. However, the dataset is large and includes numerous duplicate entries, which may introduce redundancy in IDS training.

CSE-CIC-IDS2018 Dataset [96]: The CICIDS2018 dataset was developed through a collaboration between the Canadian Institute for Cybersecurity (CIC) and Communications Security Establishment (CSE). Similar to CICIDS2017, it includes over 80 flow-based features drawn out using CICFlowMeter-V3. The dataset is systematically generated using B-profiles for normal traffic and M-profiles for attack traffic, covering the same attack types as CICIDS2017. However, it inherits similar limitations, including data redundancy and challenges in accurately reflecting real-world network variability.

BOT-IOT Dataset [97]: The BoT-IoT dataset, developed by Canberra IoT Security Lab, is tailored for IoT environments and features five testbed scenarios, including weather stations, smart refrigerators, motion sensors, remote garage doors, and smart thermostats. It contains over 72 million records covering attack types such as DDoS, DoS, OS and service scans, keylogging, and data breaches. The dataset includes 35 features, offering a rich resource for evaluating intrusion detection in IoT systems.

CICDDoS2019 Dataset [98]: Created by the Canadian Institute for Cybersecurity in 2019, the CICDDoS2019 dataset captures realistic and contemporary DDoS attack scenarios in pcap format. It includes both benign and attack traffic, with 80 flow-based features (extracted using CICFlowMeter-V3). Key attributes include source/destination IPs, ports, protocols, timestamps, and attack types. Simulated behavior from 25 users interacting over FTP, SSH, HTTP, HTTPS and email protocols supports a diverse set of modern DDoS attacks, such as NTP, LDAP, PortMap, NetBIOS, MSSQL, UDP-Lag, SNMP, SYN and DNS.

TON-IOT Dataset [99]: The TON-IoT dataset, developed by the Canberra IoT Security Lab, includes 44 features and captures both normal and malicious traffic from multiple sources: network traffic, IoT telemetry data and Windows/Linux audit logs. It covers nine key attack types, making it one of the most comprehensive IoT intrusion detection datasets available. These attacks include DoS, DDoS, Scanning, backdoor, Ransomware, injection, Cross-site scripting (XSS), man-in-the-middle (MITM) attacks and password attacks. Due to its diverse attack coverage and multi-source data collection, TON-IoT serves as a representative benchmark for IoT security research.

4.2. SDN-Contextual Datasets

Datasets shown in Table 15 are specifically designed SDN environments and they have all SDN-specific features. They are collected in emulated SDN environments or real SDN testbeds.

Table 15. SDN -contextual datasets.

Dataset Name	Year	No. of Features	Source/Repository	References
SDN-DDoS Dataset	2020	26	GitHub/Research Publication Dataset	[79]
DDoS SDN Dataset	2020	23	Bennett University/Kaggle	[100,101]
InSDN Dataset	2021	83	GitHub/Institutional Laboratory	[102]
SDN-DDoS-TCP-SYN Dataset	2021	84	Shri Mata Vaishno Devi University	[103]
SDN-SlowRate-DDoS Dataset	2023	13	University of the Basque Country and LACNIC	[104,105]
HLD-DDoSDN Dataset	2024	71	PLOS ONE	[106]
Tree-SDN-DDoS Defense Dataset	2024	35	Mendeley Data/Research Publication Dataset	[107]
SNT DDoS Detection Dataset	2025	22	Mendeley Data/Research Publication Dataset	[108]

Description

SDN-DDoS Dataset [79]: This dataset is a vital asset for evaluating AI-driven DDoS detection, particularly in machine and deep learning. It features diverse attack scenarios across two SDN topologies—each with 12 switches and 24 hosts—managed by a Ryu controller. Realistic traffic patterns simulate real-world deployments, with over 1 million labeled records capturing essential features like IP addresses, packet counts, durations, protocols, port numbers, delays, jitter, and packet loss. The dataset supports robust analysis, preserves user privacy through anonymization, and adheres to ethical data handling standards.

DDoS SDN Dataset [100]: This dataset contains 104,345 rows and 23 features, including one target variable, Label, which indicates if the network traffic is attack (1) or benign (0). The objective is the classification of traffic as normal or attack-related using classical machine learning algorithms. The feature set includes three categorical and twenty numerical features, with the Label column included among the numeric features.

InSDN Dataset [102]: Elsayed et al. developed a specialized dataset for validating intrusion detection systems within software-defined networking environments. The dataset includes both benign traffic and attack traffic, such as DoS, DDoS, web-based attacks (e.g., XSS, SQL injection), password guessing, probing, and botnet activity. Using the CICFlowMeter tool, the authors extracted over 80 features representing 56 distinct attack categories. The captured network traffic spans various application-layer protocols, including DNS, HTTP, FTP, SSH, and email services.

The SDN-DDoS-TCP-SYN Dataset [103]: This dataset targets TCP SYN flood attacks in SDN environments, simulating DDoS traffic that overwhelms the controller or data plane by exploiting the TCP handshake. Created using Mininet and monitored via SDN controllers like Ryu or POX, it contains labeled benign and malicious traffic with features such as packet size, inter-arrival time, and flow duration—ideal for training intrusion detection models.

SDN-SlowRate-DDos Dataset [104,105]: This benchmark dataset is designed to detect and analyze low-rate DDoS attacks in SDN environments. Unlike high-volume attacks, slow-rate DDoS exploits protocol timeouts by sending traffic at low rates to avoid detection. Simulated using Mininet and SDN controllers like POX or ONOS, the dataset includes labeled normal and attack traffic with features such as byte count, flow duration, inter-arrival time and packet rate. It is well-suited for training machine learning models for intrusion detection.

The HLD-DDoSDN Dataset [106]: The High-Level Dataset for DDoS Attacks in SDN (HLD-DDoSDN) is designed to support advanced research on DDoS detection in SDN. It captures both benign and attack traffic, focusing on how software-defined networking controllers and switches respond under various DDoS scenarios. With features like flow duration, packet count, inter-packet intervals, and protocol types, the dataset is optimized for machine learning- and deep learning-based IDS development. Its high-level abstraction minimizes noise, highlighting patterns critical to detecting DDoS behavior in SDN environments.

Tree-SDN-DDoS Defense Dataset [107]: This dataset is developed for detecting DDoS attacks in software-defined networking, specifically on Tree-topology SDN for environments such as WANs or hierarchical enterprise networks. It was collected using Mininet simulations and an SDN controller and includes both benign traffic and DDoS traffic.

SNT DDoS Detection Dataset [108]: This dataset comprises 1,034,669 flow records captured in a realistic software-defined networking testbed setup using Mininet and the Ryu controller and covering attack types such as ICMP floods, UDP floods, TCP SYN floods, and LAND attacks. It includes 22 flow-level features.

Publicly available datasets such as CICDDoS2019 and CAIDA have been instrumental in pushing forward intrusion detection research, yet they present notable shortcomings that limit their effectiveness in practical SDN deployments. CICDDoS2019, for example, covers a wide range of contemporary DDoS attack scenarios, but its reliance on synthetic traffic traces and predominantly flow-level attributes reduces its ability to generalize to complex, real-world environments where encrypted communications, adaptive adversaries, and diverse protocols are common. Likewise, CAIDA traces remain a key resource for traffic studies but suffer from extensive anonymization, which removes payload content and conceals key protocol fields, thereby restricting their use for detailed intrusion detection analyses. More generally, commonly adopted datasets tend to exhibit imbalanced class distributions, duplicate records, and insufficient representation of varied attack behaviors. These limitations can lead to biased machine learning models that perform well in controlled settings but fail to adapt to operational networks. Hence, while such datasets remain essential benchmarks, they should not be viewed as sufficient on their own for assessing the resilience of detection methods in programmable and evolving SDN ecosystems.

5. Future Directions and Conclusions

The fast-paced development of DDoS attacks and the growing sophistication of contemporary networks require ongoing improvements in detection and mitigation methods. The convergence of machine learning, deep learning and federated learning for DDoS detection in domain-specific and general-purpose SDN settings opens up a number of promising future research avenues.

Adaptive and real-time detection systems: One of the areas of future research is the creation of adaptive and real-time detection systems. Existing machine learning and deep learning models tend to be ineffective against zero-day attacks or quickly evolving attack patterns. Future work may involve combining reinforcement learning and online learning methods to develop self-evolving models which can adapt to new threats dynamically.

Furthermore, using lightweight federated learning models at the edge of the network may allow for quicker detection and mitigation, especially in IoT-based SDN scenarios where latency is a significant issue. To support deployment in production-scale environments, these adaptive models must also be designed with scalability in mind, ensuring they can handle high-throughput traffic without compromising accuracy.

Explainability and trustworthiness: Another significant direction is improving the explainability and trustworthiness of AI-based DDoS detection solutions. As deep learning and federated learning models increase in complexity, their "black-box" nature increasingly becomes an impediment to adoption in production environments. Future research should investigate explainable AI (XAI) methods to render model choices understandable to network administrators, with transparency in threat detection. In addition, using blockchain or secure aggregation protocols in federated learning would enhance auditability so that organizations can validate model updates without violating data privacy. Integrating privacy-preserving collaborative mechanisms, such as differential privacy or secure multi-party computation, will be key to enabling inter-organizational cooperation without exposing sensitive network data.

Cross-domain generalization: Most current methods are designed for particular network environments, restricting their generality. Future work must explore transfer learning and meta-learning methods to create generalized models that can work across various SDN deployments. Pre-trained models, fine-tuned for various domains, might minimize the necessity for heavy retraining while preserving high accuracy. Addressing scalability challenges in operational settings will also require models that can generalize across heterogeneous traffic patterns while maintaining robustness against unseen adversarial behaviors.

Scalable federated learning architectures: Existing federated learning frameworks typically make the homogeneity assumption for data distributions, which might not be the case in real-world multi-organization SDN applications. Future work can investigate hybrid federated learning schemes, integrating horizontal and vertical federated learning to support heterogeneous data structures among participants. Privacy-preserving collaborative detection approaches, such as encrypted model updates or anomaly-resistant aggregation, can further ensure secure large-scale deployments across multiple administrative domains.

Robustness and energy efficiency: Adversaries can try to poison federated learning models or go undetected with adversarial samples, which requires strong defenses like anomaly-resistant aggregation and adversarial training. At the same time, the computational requirements of deep learning and federated learning pose sustainability concerns. Future research should pre-optimize models for energy efficiency using methods like quantization, pruning, and edge-computing-aware federated learning to minimize overhead. Strengthening robustness against adversarial attacks while maintaining efficiency will be critical for practical adoption in production SDN environments.

This review paper analyzed SDN-specific security challenges, particularly DDoS attacks. The novel contributions of this survey are a comparison and classification of DDoS detection/mitigation models based on machine learning, deep learning and federated learning models for domain-specific software-defined networking environments and general-purpose software-defined networking environments. We also outlined the common performance metrics used in evaluating detection/mitigation mechanisms. Another novel contribution is to categorize the datasets as Legacy Compatible Datasets (adapted from traditional network environments) and SDN-Contextual Datasets (generated to reflect the characteristics of modern software-defined networking systems). Finally, we suggested a few directions for future research. Advancing software-defined networking security

through adaptive, explainable, and scalable DDoS detection methods remains an exciting and vital research area for the coming years.

Author Contributions: S.B. carried out the primary literature review by analyzing the existing research on DDoS attack detection/mitigation in SDN, formulated the paper's goal and added novelty to it. M.A. supervised this research process and provided guidance on the scope and structure of the review. E.A. co-supervised this work and contributed in organization and improved the clarity of the manuscript. S.F.J. assisted in collecting relevant references and writing the initial draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Taif University, Saudi Arabia, Project No. (TU-DSPP-2024-41).

Acknowledgments: The authors extend their appreciation to Taif University, Saudi Arabia, for supporting this work through project number (TU-DSPP-2024-41).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cook, S. 20+ DDoS Attack Trends and Statistics in 2024: The Rising Threat. Available online: https://www.comparitech.com/blog/information-security/ddos-statistics-facts/ (accessed on 8 July 2025).
2. Ahmed, F.; Sumra, I.A.; Jamil, U. A comprehensive review on ddos attack in software-defined network (sdn): Problems and possible solutions. *J. Comput. Biomed. Inform.* **2024**, *7*, 353–363.
3. Zoraida, B.S.E.; Indumathi, G. A comparative study on software-defined network with traditional networks. *TEM J.* **2024**, *13*, 167. [CrossRef]
4. Ko, K.M. A ddos attack detection technique through cnn model in software define network. *J. Korea Inst. Inf. Electron. Commun. Technol.* **2020**, *13*, 605–610.
5. Ko, K.M.; Baek, J.M.; Seo, B.S.; Lee, W.B. Comparative study of ai-enabled ddos detection technologies in sdn. *Appl. Sci.* **2023**, *13*, 9488. [CrossRef]
6. Bahashwan, A.A.; Anbar, M.; Manickam, S.; Al-Amiedy, T.A.; Aladaileh, M.A.; Hasbullah, I.H. A systematic literature review on machine learning and deep learning approaches for detecting ddos attacks in software-defined networking. *Sensors* **2023**, *23*, 4441. [CrossRef] [PubMed]
7. Ahmad, S.; Mir, A.H. Sdn interfaces: Protocols, taxonomy and challenges. *Int. J. Wirel. Microw. Technol.* **2022**, *12*, 11–32. [CrossRef]
8. Carvalho, R.N.; Costa, L.R.; Bordim, J.L.; Alchieri, E.A. Detecting ddos attacks on sdn data plane with machine learning. In Proceedings of the International Symposium on Computing and Networking Workshops (CANDARW), Matsue, Japan, 23–26 November 2021; pp. 138–144.
9. Aslam, N.; Srivastava, S.; Gore, M.M. A comprehensive analysis of machine learning-and deep learning-based solutions for ddos attack detection in sdn. *Arab. J. Sci. Eng.* **2024**, *49*, 3533–3573. [CrossRef]
10. Maleh, Y.; Qasmaoui, Y.; Gholami, K.E.; Sadqi, Y.; Mounir, S. A comprehensive survey on sdn security: Threats, mitigations, and future directions. *J. Reliab. Intell. Environ.* **2023**, *9*, 201–239. [CrossRef]
11. Wang, H.; Li, Y. Overview of ddos attack detection in software-defined networks. *IEEE Access* **2024**, *12*, 38351–38381. [CrossRef]
12. Hirsi, A.; Alhartomi, M.A.; Audah, L.; Salh, A.; Sahar, N.b.M.; Ahmed, S.; Ansa, G.O.; Farah, A. Comprehensive analysis of ddos anomaly detection in software-defined networks. *IEEE Access* **2025**, *13*, 23013–23071. [CrossRef]
13. Hood, D. *Sdn Architecture Issue 1.1.*; ONF TR-521 TR-521; Open Networking Foundation: Palo Alto, CA, USA, 2016.
14. Ayodele, B.; Buttigieg, V. Sdn as a defence mechanism: A comprehensive survey. *Int. J. Inf. Secur.* **2024**, *23*, 141–185. [CrossRef]
15. Cong, X.; Zhu, H.; Cui, W.; Zhao, G.; Yu, Z. Critical observability of stochastic discrete event systems under intermittent loss of observations. *Mathematics* **2025**, *13*, 1426. [CrossRef]
16. Zhao, Z.; Li, Z.; Zhou, Z.; Yu, J.; Song, Z.; Xie, X.; Zhang, F.; Zhang, R. Ddos family: A novel perspective for massive types of ddos attacks. *Comput. Secur.* **2024**, *138*, 103663. [CrossRef]
17. Mateus, J.; Zodi, G.A.L.; Bagula, A. Federated learning-based solution for ddos detection in sdn. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 19–22 February 2024; pp. 875–880.
18. Kalambe, D.; Sharma, D.; Kadam, P.; Surati, S. A comprehensive plane-wise review of ddos attacks in sdn: Leveraging detection and mitigation through machine learning and deep learning. *J. Netw. Comput. Appl.* **2024**, *235*, 104081. [CrossRef]
19. Ahuja, N.; Singal, G. Ddos attack detection & prevention in sdn using openflow statistics. In Proceedings of the 2019 IEEE 9th International Conference on Advanced Computing (IACC), Tiruchirappalli, India, 13–14 December 2019; pp. 147–152.

20. Ahuja, N.; Singal, G.; Mukhopadhyay, D.; Kumar, N. Automated ddos attack detection in software defined networking. *J. Netw. Comput. Appl.* **2021**, *187*, 103108. [CrossRef]
21. Hill, W.; Acquaah, Y.T.; Mason, J.; Limbrick, D.; Teixeira-Poit, S.; Coates, C.; Roy, K. Ddos in sdn: A review of open datasets, attack vectors and mitigation strategies. *Discov. Appl. Sci.* **2024**, *6*, 472. [CrossRef]
22. Sarica, A.K.; Angin, P. Explainable security in sdn-based iot networks. *Sensors* **2020**, *20*, 7326. [CrossRef]
23. Sharifani, K.; Amini, M. Machine learning and deep learning: A review of methods and applications. *World Inf. Technol. Eng. J.* **2023**, *10*, 3897–3904.
24. Hussein, A.; Chadad, L.; Adalian, N.; Chehab, A.; Elhajj, I.H.; Kayssi, A. Software-defined networking (sdn): The security review. *J. Cyber Secur. Technol.* **2020**, *4*, 1–66. [CrossRef]
25. Rios, V.D.M.; Inacio, P.R.; Magoni, D.; Freire, M.M. Detection and mitigation of low-rate denial-of-service attacks: A survey. *IEEE Access* **2022**, *10*, 76648–76668. [CrossRef]
26. Yousef, D.; Maala, B.; Skvortsova, M.; Pokamestov, P. Detection of non-periodic low-rate denial of service attacks in software defined networks using machine learning. *Int. J. Inf. Technol.***2024**, *16*, 2161–2175. [CrossRef]
27. Al-Eryani, A.M.; Hossny, E.; Omara, F.A. Efficient machine learning algorithms for ddos attack detection. In Proceedings of the 6th International Conference on Computing and Informatics (ICCI), Cairo, Egypt, 6–7 March 2024; pp. 174–181.
28. Chen, J.; Zheng, X.; Rong, C. Survey on software-defined networking. In Proceedings of the Second International Conference on Cloud Computing and Big Data in Asia, Huangshan, China, 17–19 June 2015; Springer: Cham, Switzerland, 2015; pp. 115–124.
29. Su, Y.; Xiong, D.; Qian, K.; Wang, Y. A comprehensive survey of distributed denial of service detection and mitigation technologies in software-defined network. *Electronics* **2024**, *13*, 807. [CrossRef]
30. Mahdavifar, S.; Ghorbani, A.A. Application of deep learning to cybersecurity: A survey. *Neurocomputing* **2019**, *347*, 149–176. [CrossRef]
31. Dixit, P.; Silakari, S. Deep learning algorithms for cybersecurity applications: A technological and status review. *Comput. Sci. Rev.* **2021**, *39*, 100317. [CrossRef]
32. Xin, Y.; Kong, L.; Liu, Z.; Chen, Y.; Li, Y.; Zhu, H.; Gao, M.; Hou, H.; Wang, C. Machine learning and deep learning methods for cybersecurity. *IEEE Access* **2018**, *6*, 35365–35381. [CrossRef]
33. Ali, T.E.; Chong, Y.W.; Manickam, S. Machine learning techniques to detect a ddos attack in sdn: A systematic review. *Appl. Sci.* **2023**, *13*, 3183. [CrossRef]
34. Ali, T.E.; Abdala, M.A.; Morad, A.H. Sdn implementation in data center network. *J. Commun.* **2019**, *14*, 223–228. [CrossRef]
35. Ali, T.E.; Chong, Y.W.; Manickam, S. Comparison of ml/dl approaches for detecting ddos attacks in sdn. *Appl. Sci.* **2023**, *13*, 3033. [CrossRef]
36. Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A survey on federated learning: Challenges and applications. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 513–535. [CrossRef]
37. Mammen, P.M. Federated learning: Opportunities and challenges. *arXiv* **2021**, arXiv:2101.05428. [CrossRef]
38. Neto, H.N.C.; Hribar, J.; Dusparic, I.; Mattos, D.M.F.; Fernandes, N.C. A survey on securing federated learning: Analysis of applications, attacks, challenges, and trends. *IEEE Access* **2023**, *11*, 41928–41953. [CrossRef]
39. Hnamte, V.; Najar, A.A.; Nhung-Nguyen, H.; Hussain, J.; Sugali, M.N. Ddos attack detection and mitigation using deep neural network in sdn environment. *Comput. Secur.* **2024**, *138*, 103661. [CrossRef]
40. Ghimire, B.; Rawat, D.B. Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. *IEEE Internet Things J.* **2022**, *9*, 8229–8249. [CrossRef]
41. Alhasawi, Y.; Alghamdi, S. Federated learning for decentralized ddos attack detection in iot networks. *IEEE Access* **2024**, *12*, 42357–42368. [CrossRef]
42. Hirsi, A.; Audah, L.; Salh, A.; Alhartomi, M.A.; Ahmed, S. Detecting ddos threats using supervised machine learning for traffic classification in software defined networking. *IEEE Access* **2024**, *12*, 166675–166702. [CrossRef]
43. Alashhab, A.A.; Zahid, M.S.; Isyaku, B.; Elnour, A.A.; Nagmeldin, W.; Abdelmaboud, A.; Abdullah, T.A.A.; Maiwada, U.D. Enhancing ddos attack detection and mitigation in sdn using an ensemble online machine learning model. *IEEE Access* **2024**, *12*, 51630–51649. [CrossRef]
44. Santos-Neto, M.J.; Bordim, J.L.; Alchieri, E.A.; Ishikawa, E. Ddos attack detection in sdn: Enhancing entropy-based detection with machine learning. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e8021. [CrossRef]
45. Kaur, A.; Krishna, C.R.; Patil, N.V. K-ddos-sdn: A distributed ddos attacks detection approach for protecting sdn environment. *Concurr. Comput. Pract. Exp.* **2024**, *36*, e7912. [CrossRef]
46. Kannan, C.; Muthusamy, R.; Srinivasan, V.; Chidambaram, V.; Karunakaran, K. Machine learning based detection of ddos attacks in software defined network. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, *32*, 1503–1511. [CrossRef]
47. Khashab, F.; Moubarak, J.; Feghali, A.; Bassil, C. DDoS Attack Detection and Mitigation in SDN Using Machine Learning. In Proceedings of the 2021 IEEE 7th International Conference on Network Softwarization (NetSoft), Tokyo, Japan, 28 June–2 July 2021; pp. 395–401.

48. Karthika, P.; Arockiasamy, K. Simulation of sdn in mininet and detection of ddos attack using machine learning. *Bull. Electr. Eng. Inform.* **2023**, *12*, 1797–1805. [CrossRef]
49. Ramprasath, J.; Seethalakshmi, V. Improved network monitoring using software-defined networking for ddos detection and mitigation evaluation. *Wirel. Pers. Commun.* **2021**, *116*, 2743–2757. [CrossRef]
50. Scaranti, G.F.; Carvalho, L.F.; Junior, S.B.; Lloret, J.; Proença, M.L., Jr. Unsupervised online anomaly detection in software defined network environments. *Expert Syst. Appl.* **2022**, *191*, 116225. [CrossRef]
51. Zhao, K.; Lu, B.; Shi, H.; Ren, G.; Zhang, Y. A ddos attack detection and defense mechanism based on the self-organizing mapping in sdn. *Internet Technol. Lett.* **2024**, *7*, e305. [CrossRef]
52. Bhayo, J.; Shah, S.A.; Hameed, S.; Ahmed, A.; Nasir, J.; Draheim, D. Towards a machine learning-based framework for ddos attack detection in software-defined iot (sd-iot) networks. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106432. [CrossRef]
53. Khedr, W.I.; Gouda, A.E.; Mohamed, E.R. Fmdadm: A multi-layer ddos attack detection and mitigation framework using machine learning for stateful sdn-based iot networks. *IEEE Access* **2023**, *11*, 28934–28954. [CrossRef]
54. Aslam, M.; Ye, D.; Tariq, A.; Asad, M.; Hanif, M.; Ndzi, D.; Chelloug, S.A.; Elaziz, M.A.; Al-Qaness, M.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. [CrossRef]
55. Alshahrani, M.M. A secure and intelligent software-defined networking framework for future smart cities to prevent ddos attack. *Appl. Sci.* **2023**, *13*, 9822. [CrossRef]
56. Garba, U.H.; Toosi, A.N.; Pasha, M.F.; Khan, S. Sdn-based detection and mitigation of ddos attacks on smart homes. *Comput. Commun.* **2024**, *221*, 29–41. [CrossRef]
57. Najar, A.A.; Naik, S.M. Cyber-secure sdn: A cnn-based approach for efficient detection and mitigation of ddos attacks. *Comput. Secur.* **2024**, *139*, 103716. [CrossRef]
58. Mehmood, S.; Amin, R.; Mustafa, J.; Hussain, M.; Alsubaei, F.S.; Zakaria, M.D. Distributed denial of services (ddos) attack detection in sdn using optimizer-equipped cnn-mlp. *PLoS ONE* **2025**, *20*, e0312425. [CrossRef] [PubMed]
59. Dandotiya, M.; Makwana, R.R.S. Ddos attack detection and mitigation in sdn environment: A deep learning perspective. In Proceedings of the IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 14–16 March 2024; pp. 1–6.
60. Novaes, M.P.; Carvalho, L.F.; Lloret, J.; Proença, M.L., Jr. Adversarial deep learning approach detection and defense against ddos attacks in sdn environments. *Future Gener. Comput. Syst.* **2021**, *125*, 156–167. [CrossRef]
61. Ahuja, N.; Mukhopadhyay, D.; Singal, G. Ddos attack traffic classification in sdn using deep learning. *Pers. Ubiquitous Comput.* **2024**, *28*, 417–429. [CrossRef]
62. Clinton, U.B.; Hoque, N.; Robindro Singh, K. Classification of ddos attack traffic on sdn network environment using deep learning. *Cybersecurity* **2024**, *7*, 23. [CrossRef]
63. Yungaicela-Naula, N.M.; Vargas-Rosales, C.; Pérez-Díaz, J.A.; Carrera, D.F. A flexible sdn-based framework for slow-rate ddos attack mitigation by using deep reinforcement learning. *J. Netw. Comput. Appl.* **2022**, *205*, 103444. [CrossRef]
64. Sayed, M.S.E.; Le-Khac, N.A.; Azer, M.A.; Jurcut, A.D. A flow-based anomaly detection approach with feature selection method against ddos attacks in sdns. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1862–1880. [CrossRef]
65. Al-Dunainawi, Y.; Al-Kaseem, B.R.; Al-Raweshidy, H.S. Optimized artificial intelligence model for ddos detection in sdn environment. *IEEE Access* **2023**, *11*, 106733–106748. [CrossRef]
66. Elubeyd, H.; Yiltas-Kaplan, D. Hybrid deep learning approach for automatic dos/ddos attacks detection in software-defined networks. *Appl. Sci.* **2023**, *13*, 3828. [CrossRef]
67. Al-Sarray, N.H.S.; Rahebi, J.; Demirhan, A. Detection of ddos attacks in sdn with siberian tiger optimization algorithm and deep learning. *Res. Sq.* 2024, *in press*. [CrossRef]
68. Chen, L.; Wang, Z.; Huo, R.; Huang, T. An adversarial dbn-lstm method for detecting and defending against ddos attacks in sdn environments. *Algorithms* **2023**, *16*, 197. [CrossRef]
69. Paidipati, K.K.; Kurangi, C.; Uthayakumar, J.; Padmanayaki, S.; Pradeepa, D.; Nithinsha, S. Ensemble of deep reinforcement learning with optimization model for ddos attack detection and classification in cloud based software defined networks. *Multimed. Tools Appl.* **2024**, *83*, 32367–32385. [CrossRef]
70. Setitra, M.A.; Fan, M. Detection of ddos attacks in sdn-based vanet using optimized tabnet. *Comput. Stand. Interfaces* **2024**, *90*, 103845. [CrossRef]
71. Alashhab, A.; Zahid, M.S.; Muneer, A.; Abdullahi, M. Low-rate ddos attack detection using deep learning for sdn-enabled iot networks. *Authorea Prepr.* 2022, *in press*. [CrossRef]
72. Zhou, H.; Zheng, Y.; Jia, X.; Shu, J. Collaborative prediction and detection of ddos attacks in edge computing: A deep learning-based approach with distributed sdn. *Comput. Netw.* **2023**, *225*, 109642. [CrossRef]
73. Ilango, H.S.; Ma, M.; Su, R. A feedforward–convolutional neural network to detect low-rate dos in iot. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105059. [CrossRef]

74. Polat, H.; Turkoglu, M.; Polat, O. Deep network approach with stacked sparse autoencoders in detection of ddos attacks on sdn-based vanet. *IET Commun.* **2020**, *14*, 4089–4100. [CrossRef]
75. Fotse, Y.S.N.; Tchendji, V.K.; Velempini, M. Federated learning based ddos attacks detection in large scale software-defined network. *IEEE Trans. Comput.* **2024**, *74*, 101–115. [CrossRef]
76. Ali, M.N.; Imran, M.; Din, M.S.U.; Kim, B.S. Low rate ddos detection using weighted federated learning in sdn control plane in iot network. *Appl. Sci.* **2023**, *13*, 1431. [CrossRef]
77. Zainudin, A.; Akter, R.; Kim, D.S.; Lee, J.M. Fedddos: An efficient federated learning-based ddos attacks classification in sdn-enabled iiot networks. In Proceedings of the 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 19–21 October 2022; pp. 1279–1283.
78. Ma, J.; Su, W.; Li, Y.; Peng, Y. Synchronizing ddos detection and mitigation based graph learning with programmable data plane, sdn. *Future Gener. Comput. Syst.* **2024**, *154*, 206–218. [CrossRef]
79. Hirsi, A.; Audah, L.; Salh, A. *SDN-DDoS Traffic Dataset*; Mendeley Data, V1; Elsevier: Amsterdam, The Netherlands, 2024.
80. MIT Lincoln Laboratory. 1998 Darpa Intrusion Detection Evaluation Dataset. 1998. Available online: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset (accessed on 20 June 2025).
81. MIT Lincoln Laboratory. *1999 DARPA Intrusion Detection Evaluation Dataset*; Massachusetts Institute of Technology, Lincoln Laboratory: Cambridge, MA, USA, 1999.
82. Lippmann, R.P.; Haines, J.W.; Fried, D.J.; Korba, J.; Das, K. The 1999 DARPA off-line intrusion detection evaluation. *Comput. Netw.* **2000**, *34*, 579–595. [CrossRef]
83. ISI Ant Lab. Darpa 2009 Ddos Attack Dataset. Available online: https://ant.isi.edu/datasets/readmes/DARPA_2009_DDoS_attack-20091105.README.txt (accessed on 22 June 2025).
84. KDD Cup 1999. The Third International Knowledge Discovery and Data Mining Tools Competition Dataset. 1999. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 25 July 2025).
85. Mittal, A.; Jain, P.; Sharma, S. Characterization of probabilistic structure of network traffic during COVID-19: A study based on mawi data. *Authorea Prepr.* 2023, *in press*.
86. Song, J.; Takakura, H.; Okabe, Y. Description of Kyoto University Benchmark Data (2006). Available online: http://www.takakura.com/Kyoto_data/BenchmarkData-Description-v5.pdf (accessed on 28 July 2025).
87. CAIDA. Caida Ddos Attack Dataset (2007). 2007. Available online: https://www.caida.org/catalog/datasets/ddos-20070804_dataset/ (accessed on 5 August 2025).
88. CAIDA. Caida: The Cooperative Association for Internet Data Analysis. 2021. Available online: https://www.caida.org/ (accessed on 11 September 2022).
89. Dhanabal, L.; Shantharajah, S.P. A study on nsl-kdd dataset for intrusion detection system based on classification algorithms. *Int. J. Adv. Res. Comput. Commun. Eng.* **2015**, *4*, 446–452.
90. University of New Brunswick. Nsl-kdd Dataset. Available online: https://www.unb.ca/cic/datasets/nsl.html (accessed on 10 August 2025).
91. Stratosphere IPS. Ctu-13 Dataset. 2011. Available online: https://www.stratosphereips.org/datasets-ctu13 (accessed on 25 July 2025).
92. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
93. Unsw-nb15 Dataset. 2017. Available online: https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 12 August 2025).
94. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Madeira, Portugal, 22–24 January 2018; Volume 1, pp. 108–116.
95. Canadian Institute for Cybersecurity. Cic-ids2017 Dataset. Available online: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 7 August 2025).
96. Canadian Institute for Cybersecurity. Cse-cic-ids2018 Dataset. 2018. Available online: https://www.unb.ca/cic/datasets/ids-2018.html (accessed on 16 August 2025).
97. The bot-iot Dataset. Available online: https://research.unsw.edu.au/projects/bot-iot-dataset (accessed on 16 August 2025).
98. Canadian Institute for Cybersecurity. Cic-ddos2019 Dataset. 2019. Available online: https://www.unb.ca/cic/datasets/ddos-2019.html (accessed on 18 August 2025).
99. The ton_iot Datasets. Available online: https://research.unsw.edu.au/projects/toniot-datasets (accessed on 19 August 2025).
100. Ddos sdn dataset. 2020. Available online: https://data.mendeley.com/datasets/jxpfjc64kr/ (accessed on 20 August 2025).
101. Ahuja, N.; Singal, G.; Mukhopadhyay, D. Ddos attack sdn dataset. *Mendeley Data* **2020**, *1*, 17632.
102. Elsayed, M.S.; Le-Khac, N.-A.; Jurcut, A.D. Insdn: A novel sdn intrusion dataset. *IEEE Access* **2020**, *8*, 165263–165284. Available online: https://www.kaggle.com/datasets/badcodebuilder/insdn-dataset (accessed on 20 August 2025). [CrossRef]

103. Shivam, K.S.G.; Karan, S. *Sdn-ddos-tcp-syn Dataset*; Mendeley Data: Amsterdam, The Netherlands, 2021. Available online: https://data.mendeley.com/datasets/8nb4cdwc9h/1 (accessed on 21 August 2025).
104. Yungaicela-Naula, N.M.; Vargas-Rosales, C.; Pérez-Díaz, J.A.; Jacob, E.; Martinez-Cagnazzo, C. Physical assessment of an sdn-based security framework for ddos attack mitigation: Introducing the sdn-slowrate-ddos dataset. *IEEE Access* **2023**, *11*, 46820–46831. [CrossRef]
105. Yungaicela-Naula, N.M.; Vargas-Rosales, C.; Pérez-Díaz, J.A.; Jacob, E.; Martinez Cagnazzo, C. Sdn-slowrate-ddos Dataset. 2023. Available online: https://ieee-dataport.org/documents/sdn-slowrate-ddos-dataset (accessed on 12 August 2025). [CrossRef]
106. Bahashwan, A.A.; Anbar, M.; Manickam, S.; Issa, G.; Aladaileh, M.A.; Alabsi, B.A.; Rihan, S.D.A. Hld-ddosdn: High and low-rates dataset-based ddos attacks against sdn. *PLoS ONE* **2024**, *19*, e0297548. [CrossRef] [PubMed]
107. Alhamamy, K.; Albermany, S. *Tree-sdn-ddos Defense Dataset*; Mendeley Data: Amsterdam, The Netherlands, 2024. Available online: https://data.mendeley.com/datasets/m4bp9wjnf9/1#:~:text=The%20%22TreeSDN%2DDDoS%20Defense%20Dataset,focus%20on%20Tree%20topology%20configurations (accessed on 22 August 2025).
108. Alawadi, A. *Snt (Simulated Network Traffic Using Mininet and Ryu) Ddos Detection Dataset*; Mendeley Data: Amsterdam, The Netherlands, 2025. Available online: https://data.mendeley.com/datasets/9hz6f62gtk/1 (accessed on 20 August 2025).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.