Discover Applied Sciences

Review

DDoS in SDN: a review of open datasets, attack vectors and mitigation strategies

Winston Hill[1] · **Yaa Takyiwaa Acquaah**[1] · **Janelle Mason**[1] · **Daniel Limbrick**[1] · **Stephanie Teixeira-Poit**[1] · **Carla Coates**[1] · **Kaushik Roy**[1]

Received: 25 June 2024 / Accepted: 26 August 2024
Published online: 31 August 2024

Abstract

Distributed denial of service (DDoS) attacks pose a significant threat to Software Defined Networking (SDN) and are frequently employed by malicious actors. SDN has emerged as a prominent networking paradigm, providing users with a decoupled control and data plane, which grants greater control and programmability over the network. In comparison to traditional networks, SDN offers dynamic, agile, cost-effective, and manageable solutions. However, a notable drawback of SDN is that the central controller becomes a vulnerable attack surface, rendering it susceptible to complete network takeover through DDoS attacks. The novelty of this paper is to gather resources that will be used to mitigate DDoS attacks in SDN environments. This paper focuses on the exploration of open datasets featuring DDoS attacks, as well as examining attack detection and mitigation techniques and frameworks. By analyzing various detection and mitigation strategies, network administrators and security professionals can make informed decisions to enhance the robustness and resilience of SDN environments in the face of evolving DDoS threats.

Keywords Software-defined networking · Distributed denial of service attacks · Open datasets

1 Introduction

Traditional distributed networks refer to interconnected systems where computing resources, data, and tasks are distributed across multiple nodes rather than centralized in a single location. These networks have been foundational in facilitating communication, resource sharing, and collaboration in various computing environments for decades. In traditional distributed networks, each node operates independently and communicates with other nodes through established protocols, allowing for decentralized decision-making and fault-tolerant operation. Examples of traditional distributed networks include client–server architectures, peer-to-peer (P2P) networks, and distributed file systems, which have played crucial roles in applications such as web hosting, content delivery, and distributed computing. In traditional distributed networks, the data and control planes are implemented using hardware devices like routers and switches. Network operators configure traffic policies for each device, which include quality of service routing and switching. However, Software Defined Networking (SDN) aims to address the limitations of traditional networking by introducing a more flexible and manageable approach. SDN decouples the control plane and data plane, allowing for greater adjustability and ease of management. The separation of the different planes are shown in Fig. 1. SDN brings advantages such

✉ Winston Hill, Wahill@aggies.ncat.edu; Yaa Takyiwaa Acquaah, ytacquaah@ncat.edu; Janelle Mason, jcmason@ncat.edu; Daniel Limbrick, dblimbri@ncat.edu; Stephanie Teixeira-Poit, steixeirapoit@ncat.edu; Carla Coates, cdcoates@ncat.edu; Kaushik Roy, kroy@ncat.edu | [1]Department of Computer Science, North Carolina Agricultural and Technical State University, Greensboro, NC 27411, USA.

Discover Applied Sciences (2024) 6:472 | https://doi.org/10.1007/s42452-024-06172-x

as centralized control through a remote device called the controller. This separation of control and data plane promotes efficient network system management and simplifies updates and changes, reducing the likelihood of human errors. SDN also offers vendor-agnostic capabilities, enabling IT administrators to employ various network devices, and upgrade the infrastructure without restrictions. The SDN controller provides a comprehensive view of the entire network, enhancing visibility and control.

While SDN has revolutionized networking technology by enabling programmability, new vulnerabilities have been introduced. In Fig. 1 the attack surface is shown and shows the entry point for different attacks. A particularly challenging attack to handle in SDN networks is Distributed Denial of Service (DDoS). DDoS targets the memory of the controller and switches, causing network bandwidth and server resources to be disabled, and disrupting normal user operations. DDoS attacks are highly damaging and coordinated, using multiple compromised machines to launch simultaneous denial of service attacks on a target. This results in resource depletion and system crashes, rendering the targeted services unavailable to legitimate users. Attackers employ DDoS attacks for various purposes, including political or financial gains, minor disruptions, or even causing widespread disruption. Such attacks can lead to loss of business profits and user trust in the affected services. A DdoS attack is a coordinated assault on the availability of services provided by a target system or network, executed through a network of compromised computing systems. The target of the attack is known as the "primary victim", while the compromised systems used to launch the attack are referred to as "secondary victims" [1]. By leveraging secondary victims, attackers can conduct much larger and more disruptive attacks while remaining anonymous, as the attack is carried out by the compromised systems. This makes it significantly harder for network forensics to trace the attack back to the actual perpetrator.

DDoS attacks have a profound impact on Software-Defined Networks (SDNs) due to their unique architecture, which separates the control plane from the data plane [1]. In SDNs, the centralized controller is a critical component responsible for managing the entire network. This centralization makes the controller a prime target for DDoS attacks. If an attacker successfully overwhelms the controller with a high volume of requests, it can exhaust the controller's computational resources, leading to significant network disruptions, dropped packets, and delayed legitimate traffic. Additionally, DDoS attacks can exploit the flow setup mechanism in SDNs by generating a large number of unique flow requests, causing flow table overloads in switches. These switches, with limited flow table capacities, may start dropping packets or suffering performance degradation. Another vulnerability lies in the northbound and southbound APIs, which, if not adequately secured, can be targeted to disrupt communication between the controller and network devices.

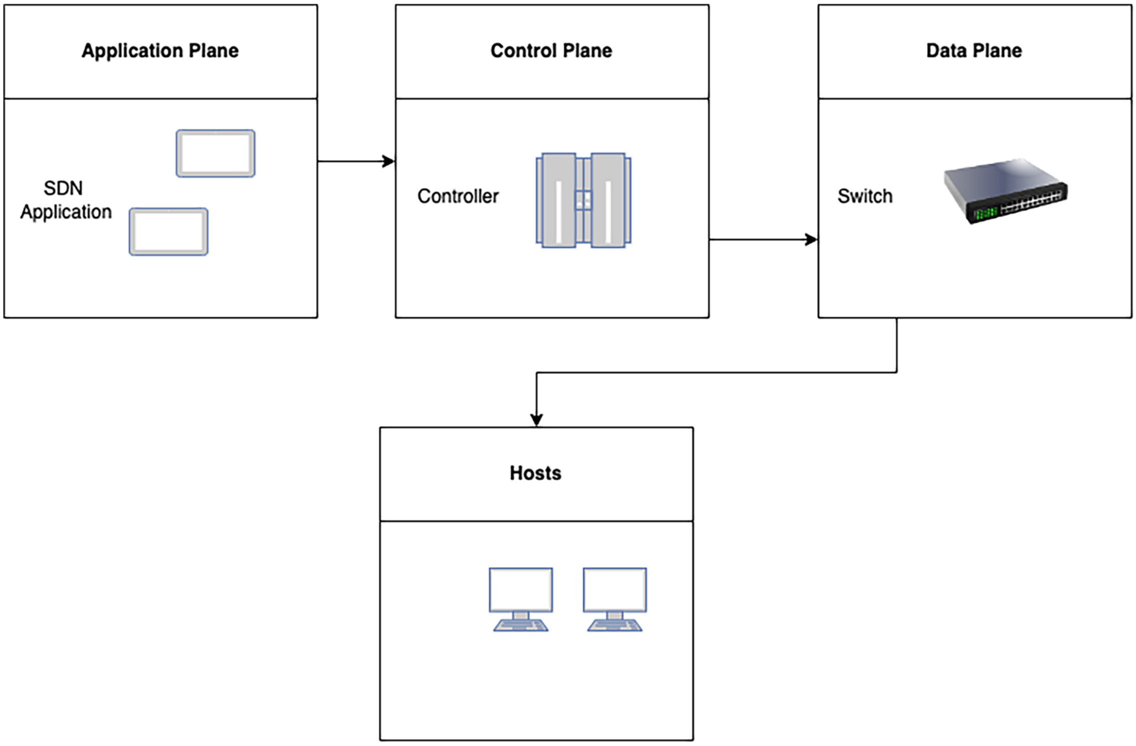

Fig. 1 SDN architecture [1]

Figure 2 (in [2]) illustrates the taxonomy of DDoS attacks, which can be broadly classified into two main categories: bandwidth depletion and resource depletion attacks. Bandwidth depletion attacks aim to flood the victim network with unwanted traffic, preventing legitimate traffic from reaching the intended recipient. On the other hand, resource depletion attacks are designed to exhaust the resources of the victim system, rendering it unable to process legitimate service requests.

Bandwidth depletion attacks can be categorized into flood attacks and amplification attacks. In flood attacks, a large volume of traffic is sent to a victim system by compromised devices (zombies), causing network bandwidth congestion. This congestion can lead to the victim system slowing down, crashing, or experiencing saturated network bandwidth, thus preventing legitimate users from accessing it. Flood attacks can be executed using both UDP (User Datagram Protocol) and ICMP (Internet Control Message Protocol) packets.

In a UDP flood attack, a large number of UDP packets are sent to random or specified ports on the victim system. The victim system attempts to process the incoming data to determine which applications have requested it. If no applications are running on the targeted port, the victim system sends an ICMP packet back to the sending system with a "destination port unreachable" message.

In an ICMP flood attack, compromised devices send large volumes of ICMP_ECHO_REPLY packets ("ping") to the victim system. These packets prompt the victim system to respond, and the combined traffic overwhelms and saturates the bandwidth of the victim's network connection.

Amplification attacks involve the attacker or compromised devices sending messages to a broadcast IP address, causing all systems within the subnet to respond to the victim system. Most routers support the broadcast IP address feature; when a broadcast IP address is specified as the destination, routers replicate the packet and send it to all IP addresses within the broadcast range. This mechanism amplifies and reflects the attack traffic, overwhelming and reducing the victim system's bandwidth.

A DDoS Smurf attack is a type of amplification attack in which the attacker sends packets to a network amplifier (a system that supports broadcast addressing), with the return address spoofed to the victim's IP address.

Similarly, in a DDoS Fraggle attack, the attacker sends UDP ECHO packets to a network amplifier. A variation of the Fraggle attack involves sending UDP ECHO packets to the port that supports character generation (chargen, port 19 on Unix systems), with the return address spoofed to the victim's echo service (echo, port 7 on Unix systems), creating an infinite loop. The UDP Fraggle packet targets the character generator in the systems reached by the broadcast address. These systems generate a character to send to the victim system's echo service, which then sends an echo packet back to the character generator, repeating the process endlessly.

DDoS resource depletion attacks involve the attacker sending packets that misuse network protocol communications or are malformed, tying up network resources and leaving none available for legitimate users.

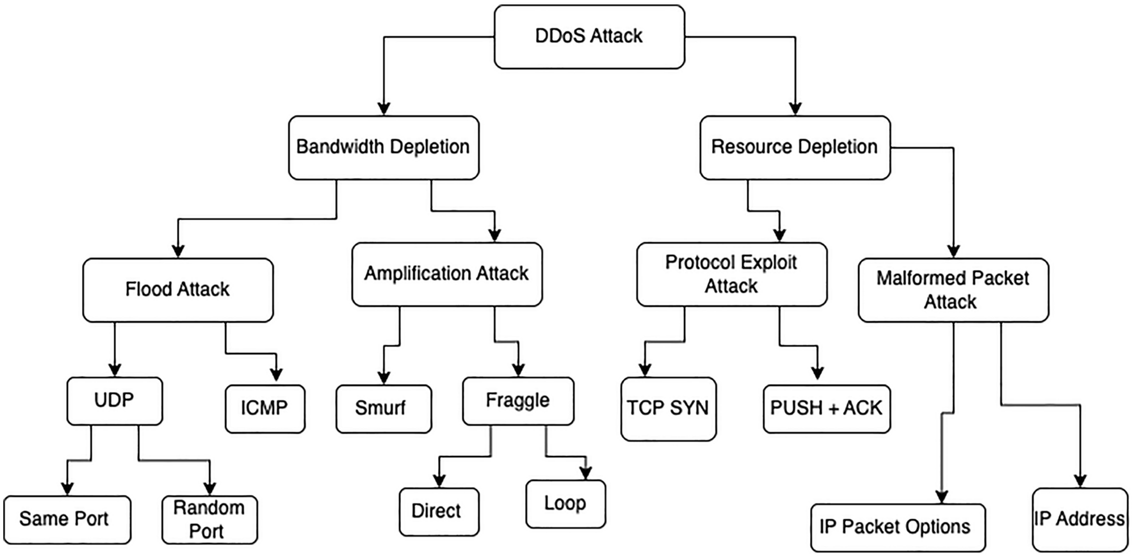

Fig. 2 Taxonomy of DDoS attack [2]

Protocol Exploit Attacks:Two examples are the misuse of the TCP SYN (Transfer Control Protocol Synchronize) protocol and the PUSH + ACK protocol. In a DDoS TCP SYN attack, the attacker instructs compromised devices to send bogus TCP SYN requests to a victim server, tying up the server's processor resources and preventing it from responding to legitimate requests. In a PUSH + ACK attack, the attacking agents send TCP packets with the PUSH and ACK bits set to one. These triggers in the TCP packet header instruct the victim system to unload all data in the TCP buffer (regardless of whether the buffer is full) and send an acknowledgment when complete.

A malformed packet attack occurs when the attacker instructs compromised devices to send incorrectly formed IP packets to the victim system to crash it. There are at least two types of malformed packet attacks. In an IP address attack, the packet contains the same source and destination IP addresses, which can confuse the victim system's operating system and cause it to crash. In an IP packet options attack, a malformed packet may randomize the optional fields within an IP packet and set all quality of service bits to one, forcing the victim system to use additional processing time to analyze the traffic.

Researchers have developed detection, mitigation frameworks, and techniques to address DDoS attacks. The rapid advancement of machine learning models has played a pivotal role in enhancing the precision of attack detection and mitigation. Open datasets are publicly available collections of data that are freely accessible for anyone to use, redistribute, and modify without any restrictions. These datasets cover a wide range of topics and are often provided by government agencies, research institutions, non-profit organizations, and individual contributors. Open datasets play a vital role in fostering transparency, innovation, and collaboration by providing valuable resources for research, analysis, application development, and education. They serve as foundational building blocks for addressing societal challenges, advancing scientific knowledge, and driving technological advancements across various domains. By using open datasets, researchers assess and fine-tune models, utilizing performance evaluation metrics such as precision, recall, and F-measure rates to gauge the efficacy of the models. The aims of this research which are open datasets featuring DDoS attacks, conducts a survey comparing DDoS in SDN, and discusses attack detection, mitigation techniques, and frameworks for mitigating distributed attacks across various services. This examination of prevailing frameworks and methodologies offers researchers and practitioners valuable perspectives on both the advantages and constraints of present approaches, thereby enabling the creation of more resilient and forward-thinking defense tactics.

Furthermore, the analysis of accessible datasets equips researchers and practitioners with a heightened awareness of the data at their disposal, enhancing its potential application in their research endeavors. The study delves into open datasets that include DDoS attack instances that are used in various research papers. For example, Wang and Wang [3] devised a traceback mitigation technique for (SDN) and relied on the InSDN [4] and CICIDS2017 [5] open datasets to develop and validate their method.

This survey explores diverse methods and frameworks designed for the detection and prevention of attacks, particularly DDoS attacks. Among the prominent frameworks examined in this research is Pro-Defense, which was developed by Bawany et al. [6] which specifically focuses on its application in smart cities. Pro-Defense adopts a modular approach, integrating various components to enhance its effectiveness.

The research contributions of this paper is an up-to-date collection of mitigation techniques and frameworks of DDoS attacks in SDN environments as well as a comprehensive overview of open datasets that feature DDoS attacks, various techniques, and frameworks for attack detection and mitigation.

Additionally, it highlights readily accessible datasets that contain instances of DDoS attacks. Secondly, by offering a comprehensive review of existing techniques and frameworks for attack detection and mitigation, this paper serves as an excellent starting point for researchers. Moreover, it presents an updated and more extensive collection of available datasets compared to those mentioned in Gebremariam et al. [7]. As a result, this work equips researchers with up-to-date datasets to enhance the accuracy and relevance of their experiments.

The compilation and organization of essential information in this paper facilitates researchers' progress, as it provides a consolidated and user-friendly resource. With the necessary information gathered and well-formatted, researchers can seamlessly access and leverage the data, ultimately advancing their investigations in the domain of attack detection and mitigation.

The remainder of this paper is organized as follows. Section II delves into the related works of DDoS in SDN. Section III gives a detailed overview of open datasets presenting an available resource that provide valuable insights into DDoS attack in SDN. Section IV focuses on the mitigation techniques and frameworks designed to thwart DDoS attacks in SDN. Section V presents discussion on critical analysis of findings, implications and potential future directions. Finally, Section VI summarizes our research contributions and offers recommendations for future work.

2 Literature review

Researchers have explored various techniques and datasets to address different security challenges. Some focused on using the same dataset but with different machine learning techniques, while others investigated different datasets from various years. This section explores these approaches in detail, covering topics like security in Internet of Things networks managed by SDN, techniques to detect and mitigate Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks using SDN, applying deep learning and machine learning for DDoS detection and mitigation, intrusion detection systems (IDS) using XGBoost, and leveraging artificial intelligence and machine learning for SDN security in general. Additionally, the section examines security vulnerabilities in vehicular networks secured by SDN.

2.1 Internet of Things networks managed by SDN

Sarica et al. [1] proposed an automated and intelligent approach for intrusion detection and mitigation in SDN, aimed at providing explainable security in the Internet of Things network of the 5G era. Their approach utilized automated flow feature extraction and a highly accurate random forest classifier for network flow classification at the SDN application layer. The classifier could detect various types of attacks and take corrective actions by installing new flow rules with high priority at the data plane. Additionally, the authors presented a SDN-specific dataset modeling a realistic IoT environment that contains flow data for common network attacks and normal traffic. Results on the precision of intrusion detection, as well as performance results with and without their proposed security mechanism, were reported [1]. While the researchers proposed a promising security mechanism, it’s worth noting that their detection system was primarily evaluated in the context of Denial of Service (DoS) and DDoS attacks. Consequently, the dataset used in the experiments may not adequately represent other types of attacks, such as fuzzing and port scanning. Additionally, the experiments involved a single malicious host generating attacks, which differs from real-world scenarios where threats can originate from various sources. Despite these minor limitations, the paper is commendably written and presents data in a clear and comprehensible manner.

Sarica et al. [8] generated untitled dataset for intrusion detection in IoT networks managed by SDN. The datasets include both static and dynamic IoT networks and consist of 27.9 million and 30.2 million data records, respectively. The datasets contain various types of cyber attacks, including DoS, DDoS, port scanning, OS fingerprinting, and fuzzing, as well as benign traffic. The Open Network Operating System (ONOS) controller was utilized as the controller in this study, while hping3 was the tool used to craft TCP/IP packets for DoS and DDoS attacks, Nmap was used for port scanning and OS fingerprinting, and Boofuzz was used for fuzzing. The datasets consist of 33 features. The primary differences between the features are the number of IoT devices in the network, with one dataset containing five IoT devices and the other containing 10 IoT devices. The legitimate traffic was self-generated through the examination of the IoT bot dataset, but a noteworthy challenge was the unavailability of open-source software for this purpose.

Koroniotis et al. introduced a new dataset called Bot-IoT, which included regular IoT-related traffic and various types of attack traffic common to botnets [9]. The data were labeled with features indicating the attack flow, category, and subcategory. Statistical measures such as correlation coefficient and entropy were used to determine the top 10 features. To evaluate the dataset’s quality, researchers trained SVM, Recurrent Neural Network (RNN), and Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) models. Four metrics were used to measure the validity of the dataset: Accuracy, Precision, Recall, and Fall-out. Results showed that the full-featured dataset achieved the highest accuracy and recall when trained with an SVM model, while the 10-best features dataset achieved the highest precision and lowest fall-out when trained with an SVM model. Although this study laid a valuable groundwork for comprehending attack traffic directed at IoT devices, it was not devoid of limitations. One key constraint was the use of only 5% of the generated dataset for testing, mainly due to challenges in managing the extensive 72,000,000 records. Instead, the researchers opted for a more manageable 3,000,000 records for both training and testing sets, potentially introducing statistical errors like false positives due to the smaller sample size. Furthermore, although the research detailed the data collection and processing methods for machine learning, replicating the process proved challenging, partly due to the learning curve associated with certain tools, such as the argus tool. Additionally, some tools that were initially open source had transitioned to a commercial model, as exemplified by ostinato, which might impose additional barriers for replication.

Researchers created a Counter-based DDoS Attack Detection (C-DAD) for detecting DDoS attacks framework [10]. The framework is integrated with the SDNWISE controller and uses the Cooja simulator to establish a Software Defined Internet of Things (SD IoT) network environment. SDNWISE is a two-part solution for wireless sensor networks. It offers both a software framework based on the concept of Software-Defined Wireless Sensor Networks (SD-WSN) and a physical hardware prototype to test and implement the software. By analyzing IoT traffic, the C-DAD framework can identify DDoS attacks with high accuracy. The researchers conducted three experiments to demonstrate the effectiveness of the framework in detecting these attacks. While the framework functions effectively, one limitation is the IoT controller operates in a single network in a heterogeneous environment and does not demonstrate its performance across multiple IoT networks in such diverse environments.

Yin et al. [11] created an algorithm and frameworkto detect and mitigate DDoS attacks in a SD-IoT environment. The framework comprises a pool of SDN controllers, SD-IoT switches with IoT gateways and IoT devices. The algorithm uses cosine similarity to compare vectors of the packet-in rate of the boundary switch. The researchers tested the framework and algorithm in a simulated SDN environment using Mininet, with Floodlight serving as the controller [11]. The researchers faced challenges in developing a proactive defense system that could promptly identify and counteract DDoS attacks in real-time. Furthermore, they did not implement dynamic load balancing in the controller pool, a mechanism designed to evenly distribute incoming network traffic or workloads across multiple controllers or resources. This load balancing strategy enhances resource utilization and promotes efficient network performance. In conclusion, simulations demonstrate that the proposed algorithm effectively identifies the source of a DDoS attack launched from an IoT device. This rapid detection allows for quicker mitigation, ultimately improving the security of vulnerable IoT networks. These networks are often constrained by the limited computational power and memory of their terminal devices, making the efficiency of the algorithm even more crucial.

Table 1 provides an overview of the dataset information advantages and disadvantages of Internet of Things Networks Managed by SDN.

2.2 Detection and mitigation of DoS, DDoS using SDN

Galeano-Brajones et al. [13] developed a stateful SDN approach to detect and mitigate DoS and DDoS attacks in an IoT network. They utilized the open state extension to create stateful features over an open virtual switch. The experiments were conducted in three scenarios. The first scenario analyzed the bandwidth consumption during a DoS attack and mitigation process and entropy values. The experiment used a SDN Testbed consisting of a Ryu controller and Mininet for the testbed. The first scenario served as a baseline, measuring bandwidth and entropy during the attack in a general test environment. The following two scenarios specifically focused on how the metric performs within an Internet of Things (IoT) setting. The next two scenarios utilized the Bot-IoT [12] dataset to test the detection and mitigation process of a DoS attack in an IoT environment. The last scenario focused on the detection and mitigation of DDoS attacks in an IoT environment. Several limitations are evident in this study. Firstly, the research exclusively employed UDP DDoS attacks in their experimentation, offering a limited perspective on the efficacy of mitigating DDoS attacks across different attack vectors. Secondly, issues arose when switches became unresponsive to the control plane when the window size was excessively large, potentially hampering the practicality of the proposed approach. Lastly, enhancing the detection process might be facilitated by incorporating alternative statistical-based metrics and machine learning techniques.

A two-phase detection approach was developed in [14] to enable early detection of DDoS attacks. This approach involved the use of two algorithms that assign a trust value to each user in the system. The proposed approach tackles DDoS attacks through a two-phase algorithm that assigns a dynamic trust value to each user. This approach contrasts with methods that rely on a single, ever-changing threshold value. The algorithm functions in two stages: the first phase involves extracting header fields from incoming data packets. The second phase then utilizes this extracted information to calculate a trust value for each user. This trust value plays a key role in identifying potential DDoS attacks. The experiment was conducted using mininet, Wireshark, Scapy, and pox controller. This research presents a robust method for identifying attacks in an SDN network. However, a notable drawback lies in the quest to strike a balance for the threshold system. If the threshold is set too low, it can result in false positives, while if it is set too high, it may lead to false negatives. Furthermore, the sensitivity of the detection system relies on the trust levels assigned in the system, which could introduce challenges in the detection process.

Assis et al. [15] proposed a defense system for SDN networks that can detect and mitigate DDoS attacks on the controller and external servers by inspecting traffic in one-second intervals. Their detection model uses a CNN and was compared to three other anomaly detection approaches: Logistic Regression (LR), MLP, and Dense MLP (D-MLP). The methods were evaluated on two scenarios: one using simulated SDN data generated by Mininet and Floodlight, and the other using a public dataset called CICDDoS 2019 [16]. CICDDoS 2019 provides more than 12 types of DDoS attacks. The CNN method achieved the highest accuracy, precision, and F-measure outcomes with low false-positive rates, while

Table 1 Summary of the dataset information advantages and disadvantages of Internet of Things networks managed by SDN

Literature	Dataset information	Advantages	Disadvantages
Sarica et al. [1]	SDN dataset [8]	The classifier could detect various types of attacks and take corrective actions by installing new flow rules with high priority at the data plane	The dataset used in the experiments may not adequately represent other types of attacks, the experiments involved a single malicious host generating attacks, which differs from real-world scenarios
Sarica et al. [8]	SDN dataset [8]	Datasets include both static and dynamic IoT networks	A noteworthy challenge was the unavailability of open-source software for this purpose
Koroniotis et al. [9]	Bot-Iot [12]	The research detailed the data collection and processing methods for machine learning	Replicating the process proved challenging
Bhayo et al. [10]	No dataset	The framework functions effectively	Does not demonstrate its performance across multiple IoT networks
Yin et al. [11]	No dataset	This rapid detection allows for quicker mitigation, ultimately improving the security of vulnerable IoT networks	Networks are often constrained by the limited computational power and memory of their terminal devices

the MLP and D-MLP methods were better for the recall metric. The researchers also described a mitigation module and presented a game-theoretical approach as an example of its operation. The game theory-based technique enhances the packet discard rate in a policy used inside the central controller of the SDN, and it was found to be successful in restoring the SDN's regular functions. While the detection module proved successful, the research has some limitations. One primary limitation pertains to the use of a limited number of hosts during the testing phase of the module. Deploying more hosts could have increased the challenge of detecting DDoS attacks, potentially making them more covert. Another limitation lies in the relatively restricted use of machine learning techniques in the detection module, which could have been broadened to enhance its effectiveness.

Kiani et al. [17] proposed an algorithm for identifying anomalies in SDN. One algorithm focused on detecting centralized inconsistencies in the flow table, while the other is designed to identify distributed inconsistencies across multiple network switches. The performance of these algorithms was evaluated using Mininet with a Ryu controller, and they were found to be effective in detecting anomalies. However, the only limitation observed was that the detection process took longer if the path to be traversed was longer. One limitation addressed by the researchers in their proposed algorithm is that an increase in the number of rules in the flow table results in longer path lengths being sent to the algorithm. Consequently, extending the time required for anomaly detection. However, it's noteworthy that the algorithm demonstrated its effectiveness in detecting all anomalies within a testbed environment comprising 30 switches and 100 hosts.

A method for mitigating traceback in SDN networks was developed by Wang et al. [3]. Traceback refers to the process of identifying the source of network traffic or malicious activities within a network. It involves tracing the path taken by packets or data packets from their destination back to their origin. The authors utilized 10 attributes to trace the source of a DDoS attack implemented with hping3. Additionally, a hybrid deep learning model called CNN-ELM, which combined Convolutional Neural Network (CNN) and Extreme Learning Machine (ELM), for accurate anomaly detection was proposed by Wang et al. [3]. The model's effectiveness was tested using the CICIDS-2017 [5] and InSDN [4] datasets. The CICIDS-2017 dataset achieved a 98.92% accuracy rate, while the InSDN dataset demonstrated even better accuracy of 99.91%. A notable drawback of this approach is its reliance on supervised learning, which necessitates a substantial investment in labeled data. The researchers are optimistic about addressing this challenge in their future work by implementing a graph neural network.

A security mechanism was proposed by authors of SDIoT-DDoS-DA to detect and mitigate DDoS attacks on SDN networks [18]. The system was implemented using the SDN-Wise framework and ONOS open network system as the controller. Trinoo was utilized to generate DDoS attacks, and the system's performance was tested on a dataset comprising 1054 requests. Trinoo is a well-known tool used in many past attacks against major websites, to simulate DDoS attack traffic. Trinoo launches attacks by flooding the target with UDP packets, while maintaining communication between the attacker and a central control program using TCP. This distinction between the packet type used in the attack itself (UDP) and the communication protocol between attacker and controller (TCP) is a key feature of Trinoo. One significant limitation of the study was the relatively small test size, consisting of just 1054 requests, which can impact the accuracy of the results. The simulation outcomes demonstrated that the detection system categorized 876 requests as "illegitimate access", suggesting their potential association with a DDoS attack. In this dataset, 11 requests were inaccurately identified as "False Positives", indicating that the system erroneously flagged some valid requests as malicious. The detection module successfully classified 178 commands as legitimate access requests. However, it also generated 32 false negatives. These represent instances where the system mistakenly labeled malicious requests as benign, highlighting a need for further refinement within the detection module itself.

Bawany et al. [6]. developed a modular framework called ProDefence to detect and mitigate DDoS attacks in a large-scale network, including a smart city based on an SDN infrastructure. The framework includes multiple components, such as a traffic flow collector, policy engine, attack detector, and mitigation engine. ProDefence has the advantage of allowing customized criteria for detecting DDoS attacks, while also supporting load balancing and reducing the risk of controller failure by using a distributed controller platform. ProDefense faces challenges in environments with inconsistent network controllers. The system's reliance on Node.js for consistency might be problematic during rapid network rule deployment. Additionally, while ProDefense claims adaptability to any programming language, this flexibility requires further investigation to ensure consistent operation across different scripting environments.

Researchers devised OpCloudSec, an attack and reaction system that is based on an anomaly [19]. The system utilizes a Deep Belief Network (DBN) to construct the attack prevention model. The UNB ISCX [20] dataset was employed for experimentation. The attack model demonstrated greater accuracy than SVM, Self-organizing Map (SOM), and NB classifiers. This study describes their system's ability to enhance security while minimizing disruptions to the overall network. Notably, the research relies on the UNB ISCX dataset, which, as it was not generated in an SDN testbed. It may not offer the same level of accuracy that an SDN-specific dataset would provide for validating the system. Additional testing is needed with a SDN specific dataset.

Singh et al. [21]Researchers reviewed approximately 70 prominent research articles on the topic. They found that around 47% of the studies used information theory-based methods, about 42% employed machine learning-based methods, and approximately 20% utilized artificial neural network-based methods to detect DDoS attacks in SDN. The controller remains the primary target for attackers, as it is the most critical component of the SDN-enabled network. Several research challenges persist, including the acceptance of SDN architecture, the non-availability of production-level controllers, scalability issues, security concerns regarding SDN switches and communication links, and dependency on a central controller. These challenges continue to hinder the implementation of a secure SDN, making it an open issue.

Singh et al. [22]Researchers propose a classification of security vulnerabilities exposed by SDN architecture that can be exploited by new-flow-based DDoS attacks. They provided an analysis of the latest developments in DDoS detection and mitigation research aimed at addressing these security vulnerabilities. Finally, they discussed SDN security-related research challenges that can be valuable for the research community and academics for further investigation. This review presents a classification of security issues in SDN architecture related to new-flow-based DDoS attacks. By focusing on intrinsic security issues, they devised a taxonomy of major design vulnerabilities in SDN architecture that allow DDoS attackers to initiate new-flow-based attacks. The proposed taxonomy offers a clear hierarchical view of potential vulnerabilities that can be exploited in SDN architecture. This framework will help the research community identify effective DDoS defense solutions by understanding the dimensions, interdependencies, and impacts of DDoS attack problems. Additionally, they provide a state-of-the-art review of DDoS defense solutions, including recent developments and techniques used by researchers in the SDN environment. Researchers present rationales for DDoS detection to gain insights into novel strategies employed over recent years, highlighting how DDoS attack patterns have evolved. Finally, they examine some challenging research questions related to SDN security that remain unresolved.

Table 2 provides an overview of the dataset information advantages and disadvantages of Detection and Mitigation of DoS, DDoS using SDN.

2.3 Deep learning and machine learning detection and mitigation techniques with DDoS

A framework proposed in [23], uses Long Short-Term Memory (LSTM) to detect DDoS attacks on the source side of the network. The framework aims to address performance degradation caused by unpredictable network traffic patterns by applying LSTM-based adaptive thresholds to each network on the source side. Additionally, the framework constructs a collaborative network consisting of multiple detection sites to aggregate feedback, including local traffic patterns, detection rates, and timestamps from each site. To evaluate the effectiveness of the framework, real-world DNS request traffic collected from DNS-STAT: Hedgehog, operated by ICANN (Internet Corporation for Assigned Names and Numbers), was used in the experiment [23]. A limitation of the study was the potential for improvement in the source-side attack detection method through the application of dynamic seasonality embedding. This approach involves analyzing and adapting to evolving patterns or trends in the data, which could enhance detection accuracy. Additionally, the detection process could be enhanced by adopting a more selective approach when choosing which devices collaborate in the system.

Ravi et al. [24] proposed the Learning-Driven Detection Mitigation (LEDEM) approach to detect DDoS attacks using a semi-supervised machine learning algorithm. The LEDEM approach was evaluated using the UNB-ISCX [20] dataset and testbed, as well as the emulated topology environment in Mininet, and was compared to existing solutions. The results showed LEDEM has significantly increased the accuracy rate of detecting DDoS attacks by 96.28%. A notable limitation in the research is its reliance on a single machine learning technique, the choice of semi-supervised ELM. This decision was challenging due to the potential for false positives associated with both supervised and unsupervised methods. Identifying an alternative approach that not only maintains the current model but also offers clear improvements poses a considerable challenge.

Santos et al. [25]: In this study, machine learning algorithms—namely MLP, SVM, Decision Tree, and Random Forest—were proposed to detect DDoS attacks in three categories: flow-table attacks, bandwidth attacks, and controller attacks. These attacks were generated using the Scapy tool, leveraging a list of over 20,000 IP addresses as attackers, with the hosts connected to the SDN network as targets, simulated using the Mininet Virtual Network (Mininet VN) with a POX controller. The experimental results revealed that the Decision Tree algorithm was generally the best performer due to its low processing time, despite the Random Forest algorithm achieving the highest accuracy in absolute terms.

Table 2 Summary of the dataset information advantages and disadvantages of detection and mitigation of DoS, DDoS using SDN

Literature	Dataset information	Advantages	Disadvantages
Galeano-Brajones et al. [13]	Bot-Iot [12]	They utilized the open state extension to create stateful features over an open virtual switch	Employed UDP DDoS attacks in their experimentation, offering a limited perspective on the efficacy of mitigating DDoS
Salem et al. [14]	No dataset	This approach contrasts with methods that rely on a single, ever-changing threshold value	The balance for the threshold system
Assis et al. [15]	CICDDoS 2019 [16]	CNN method achieved the highest accuracy and precision	A limited number of hosts during the testing phase of the module
Kiani et al. [17]	No dataset	The algorithm demonstrated its effectiveness in detecting all anomalies within a testbed environment comprising 30 switches and 100 hosts	Increase in the number of rules in the flow table results in longer path lengths
Wang et al. [3]	InSDN [4], CICIDS2017 [5]	The CICIDS-2017 dataset achieved a 98.92% accuracy rate, while the InSDN dataset demonstrated even better accuracy of 99.91%	Reliance on supervised learning
Wani et al. [18]	No dataset	The detection module successfully classified 178 commands as legitimate access requests	The system mistakenly labeled malicious requests as benign
Bawany et al. [6]	No dataset	ProDefence has the advantage of allowing customized criteria for detecting DDoS attacks	Claims adaptability to any programming language, this flexibility requires further investigation
Sharma et al. [19]	UNB ISCX [20]	The attack model demonstrated greater accuracy than SVM, Self-organizing Map (SOM), and NB classifiers	Research relies on the UNB ISCX dataset, which, as it was not generated in an SDN testbed
Singh et al. [21]	No dataset	Reviewed approximately 70 prominent research articles	The non-availability of production-level controller
Singh et al. [22]	No dataset	The proposed taxonomy offers a clear hierarchical view of potential vulnerabilities that can be exploited in SDN architecture	DDoS attack patterns have evolved

Nadeem, Muhammad Waqas, et al. [26] This paper evaluates several important feature selection methods for machine learning in the context of DDoS detection. The selection of optimal features significantly impacts the classification accuracy of machine learning techniques and the performance of the SDN controller. Researchers provide a comparative analysis of feature selection methods and machine learning classifiers to detect SDN attacks. The experimental results show that the Random Forest (RF) classifier, when trained on features selected by the Recursive Feature Elimination (RFE) method, achieves a high accuracy of 99.97%. The NSL-KDD dataset [27]is used for training and testing the machine learning classifiers for DDoS attack detection. Three different filter methods—Information Gain (IG), Correlation Coefficient (CC), and Chi-Square—are used for selecting optimal features. Additionally, three different wrapper methods—Forward Feature Selection (FFS), Backward Feature Elimination (BFE), and Recursive Feature Elimination (RFE)—are employed to rank the most optimal features. The Lasso embedded feature selection algorithm is also utilized to eliminate the weights of the least important features, providing a reduced set of features. Classifiers such as SVM, KNN, NB, RF, and DT are evaluated. The extraction and selection of optimal features are crucial for accurately detecting attacks in machine learning-based models. Experimental results demonstrate that the RF classifier trained on RFE-ranked feature subsets achieves excellent results in detecting attacks on the SDN controller. However, resource consumption of the SDN controller increases, and detection accuracy decreases under larger-scale network traffic. Additionally, using irrelevant and excessive features increases the SDN controller's workload, potentially affecting its efficiency.

Dehkordi et al. [28] This paper presents a new method for detecting DDoS attacks in SDN, comprising three main sections: collector, entropy-based, and classification. Experimental results using the UNB-ISCX [20] and other datasets demonstrate that this method outperforms existing approaches in terms of accuracy. The method was evaluated, revealing that entropy-based sections with static thresholds do not produce satisfactory results across different datasets. Better results were achieved with dynamic thresholds, though at the cost of a high false positive rate (FPR). To address this drawback, various classification algorithms were applied, resulting in more accurate outcomes. The significance of this method lies in its superior accuracy compared to other similar methods. The experimental results indicate that this approach achieves higher accuracy. However, since the proposed model focuses on finding solutions post-attack, the approach for preventing DDoS attacks in SDN networks needs further assessment. In this study, the Floodlight controller and Mininet 2.2.1 were used for network simulation. The RandomTree, Logistic Regression, J48, BayesNet, and REPTree classification algorithms, along with the K-fold method at $K = 10$, were involved in detecting low-volume attacks.

Perez-Diaz, Jesus Arturo, et al. [29] In this paper, we present a flexible modular architecture designed to identify and mitigate low-rate DDoS (LR-DDoS) attacks in SDN settings. They trained the intrusion detection system (IDS) using six machine learning (ML) models—J48, Random Tree, REP Tree, Random Forest, Multi-Layer Perceptron (MLP), and Support Vector Machines (SVM)—and evaluated their performance using the Canadian Institute of Cybersecurity (CIC) DoS dataset [5]. The findings demonstrate a detection rate of 95%, despite the inherent difficulty in detecting LR-DDoS attacks. They deployed the open network operating system (ONOS) controller on a Mininet virtual machine to closely mimic real-world production networks.The SlowHTTPTest tool was used to launch LR-DDoS attacks from the attackers to the virtualized web server in the evaluations, ensuring the results can be easily transferred to a real-world production environment. The intrusion prevention system successfully mitigated all attacks detected by the IDS. The modular and flexible security architecture designed allows for easy replacement of any module without affecting the others. The IDS module detects flows using different pre-trained ML models, which can be developed with various programming languages and frameworks. The evaluations of the six ML algorithms using the CIC DoS Dataset (2017)[5] reported a 95% accuracy rate. Using two different topologies, they demonstrated that all attacks identified by the IDS were successfully mitigated.

Table 3 provides an overview of the dataset information advantages and disadvantages Deep Learning and Machine Learning Detection and Mitigation Techniques with DDoS.

2.4 IDS using XGBoost algorithm

The authors in [30] developed an Intrusion Detection System (IDS) for Vehicular Ad-hoc Networks (VANETs), leveraging the ToN-IoT dataset [31]. To improve the model's efficiency and speed, the researchers employed the Chi^2 technique for feature selection, which reduced the number of features from 108 to 20. The Chi-square (Chi^2) technique is a valuable tool in machine learning for selecting the most relevant features from a dataset. It's a statistical approach that calculates a score based on the dependency between a feature and the target variable (class labels). This allows us to identify and exclude features that have little to no bearing on predicting the outcome. By focusing on the most informative features, the Chi^2 technique helps improve the overall efficiency and accuracy of machine learning models. The Synthetic Minority Over-sampling Technique (SMOTE) was applied to mitigate overfitting and bias towards the majority class and address

Table 3 Summary of the dataset information advantages and disadvantages of deep learning and machine learning detection and mitigation techniques with DDoS

Literature	Dataset information	Advantages	Disadvantages
Yeom et al. [23]	No dataset	Real-world DNS request traffic collected from DNS-STAT: Hedgehog, operated by ICANN (Internet Corporation for Assigned Names and Numbers)	The potential for improvement in the source-side attack detection method
Ravi et al. [24]	UNB ISCX [20]	The results showed LEDEM has significantly increased the accuracy rate of detecting DDoS attacks by 96.28%	The potential for false positives associated with both supervised and unsupervised methods
Santos et al. [25]	No dataset	The Random Forest algorithm achieving the highest accuracy	Decision Tree algorithm was generally the best performer due to its low processing time
Nadeem et al. [26]	NSL-KDD [27]	RF classifier trained on RFE-ranked feature subsets achieves excellent results in detecting attacks on the SDN controller	Resource consumption of the SDN controller increases, and detection accuracy decreases
Dehkordi et al. [28]	UNB ISCX [20]	Various classification algorithms were applied, resulting in more accurate outcomes	Better results were achieved with dynamic thresholds, though at the cost of a high false positive rate (FPR)
Perez-Diaz et al. [29]	CICIDS2017 [5]	All attacks identified by the IDS were successfully mitigated	The inherent difficulty in detecting LR-DDoS attacks

the imbalance dataset issues. Various evaluation metrics such as accuracy, precision, recall, F1-score, False Positive Rate (FPR), and confusion matrix were employed to assess the performance of the machine learning model. The eXtreme Gradient Boosting (XGBoost) algorithm exhibited the highest performance among the tested algorithms. This paper provides a comprehensive explanation of the data processing process for implementing machine learning techniques. The utilization of the Ton-IoT dataset required substantial preprocessing efforts. An alternative dataset could be used for testing their system in the future, which could prove beneficial.

In [32], the authors developed an IDS that quickly detects attacks and make decisions. The system used artificial neural network (ANN) and was tested with the UNSW-15 dataset [33] to assess its efficacy. The evaluation results showed the proposed method has an accuracy rate of 84% in detecting threats with a low false positive rate of 8%. The IDS will be integrated into an IoT controller to distinguish between benign and malicious data, with malicious data being discarded promptly. In the comparative analysis of this system two different datasets were employed. An ideal experiment would use the same dataset for all systems to ensure a fair comparison. It’s important to note that the NSL-KDD [27] differ from the UNSW-15 dataset in several aspects, such as their smaller size, fewer instances of attacks, a reduced number of networks, and a higher level of repetition.

Cruz et al. [34] proposed a method for creating a lightweight and low-overhead model that achieves an accuracy rate of 93.6%. The study utilized XGBoost as the machine learning model and tested it on the IoT-23 dataset [35] with a focus on replication attacks. A replication attack is a deceptive maneuver where an attacker tries to gain unauthorized access to a system by mimicking legitimate traffic. The model used in this method could be integrated into an IoT middleware solution to block access by suspicious devices. An enhancement to this research could be to assess the versatility of the model and system performance based on different datasets. Additionally, it would be beneficial to compare the XGBoost algorithm with other machine learning techniques, such as Random Forest or models that adapt themselves to the dataset during training.

Table 4 provides an overview of the dataset information advantages and disadvantages IDS using XGBoost algorithm.

2.5 Artificial intelligence and machine learning approaches in SDN

Gebremariam provides valuable insights into the utilization of AI/ML in SDN and Network Functions Virtualization (NFV) networks, by identifying significant progress in this field as well as future challenges in [7]. They believe that the integration of AI/ML could lead to networks that are self-configured, self-adaptive, and self-managed. Nevertheless, limited network resources and the absence of easily available datasets remain significant obstacles to research in this domain. This research introduces several noteworthy concepts. Cloud operations require efficient algorithms and edge computing to handle complex tasks while minimizing data center power consumption. Additionally, strategically storing data across the network safeguards against failures and attacks, ensuring data integrity and resilience.

Zhoe created a framework that automates the generation, deployment and adjustment of proactive defense measures for IoT [36].The team utilized the SDN architecture’s flexibility to efficiently deploy defense mechanisms. The framework is founded on Moving Target Defence (MTD) techniques and cyber deception. This approach intends to mislead attackers and limit disruptions to the system. The team employed the Ryu controller and Mininet for the SDN infrastructure during the development of the test environment. This framework exhibits two key weaknesses. Firstly, it’s vulnerable to resource-exhaustion attacks. Attackers with significant resources can overwhelm the system with a flood of requests, rendering it inoperable. Secondly, the framework struggles to differentiate between legitimate users and insiders. This limits its effectiveness to external threats, leaving it susceptible to attacks launched from within the system itself.

Sahoo et al. [37] developed a technique for detecting and mitigating DDoS attacks. It utilizes a machine learning model that combines SVM, kernel principal component analysis (KPCA), and genetic algorithm (GA). The SVM acts as the classifier, while KPCA is used to extract features from the dataset, and GA optimizes the SVM. Furthermore, a refined kernel function (N-RBF) is utilized to reduce feature differences, also called noise. To evaluate the model, the authors used two datasets, including NSL-KDD [27], and an SDN testbed with a POX controller. The SDN testbed was set up using Mininet. Several limitations are associated with this solution. Firstly, it struggles with the detection of ICMP packets due to their classification in the “smuff” class, making it challenging to distinguish between potential attacks and benign traffic. Secondly, scalability concerns arise as the testbed employed a single controller, performing adequately in that context. However, uncertainties emerge when considering environments with multiple controllers. This prompts questions about its ability to maintain effectiveness.

Tan et al. developed a framework for identifying and preventing DDoS attacks [38]. The framework employs a trigger mechanism on the data plane to detect such attacks and uses a machine learning algorithm based on KNN and K-Means

Table 4 Summary of the dataset information advantages and disadvantages of IDS using XGBoost algorithm

Literature	Dataset information	Advantages	Disadvantages
Gad et al. [30]	ToN-IoT [31]	The eXtreme Gradient Boosting (XGBoost) algorithm exhibited the highest performance among the tested algorithms	The utilization of the Ton-IoT dataset required substantial preprocessing efforts
Hanif et al. [32]	NSL-KDD [27]	The proposed method has an accuracy rate of 84%	An ideal experiment would use the same dataset for all systems to ensure a fair comparison
Cruz et al. [34]	IoT-23 [35]	The model used in this method could be integrated into an IoT middleware solution to block access by suspicious devices	Did not compare the XGBoost algorithm with other machine learning techniques

on the SDN controller to identify suspicious flows. The effectiveness of the framework has been evaluated using both the NSL-KDD [27] dataset and a SDN environment created with Mininet, with the ONOS controller being utilized. Researchers encountered challenges related to the controller's capability as the network's workload expanded to accommodate larger-scale traffic. The framework's DDoS detection suffers under heavy network loads. As data volume increases, especially during peak activity, the controller becomes overloaded. This strain on the controller hinders its ability to effectively detect DDoS attacks. Consequently, the network's vulnerability to DDoS attacks grows, potentially leading to a decline in service quality during high-traffic periods.

A framework uses a discrete scalable memory-based support vector machine (DSM-SVM), developed by Revathi et al. [39], to forecast and mitigate DDoS attacks on an SDN. To evaluate the algorithm, the researchers employed the KDD99 [40] and set up a testbed for the SDN using Mininet and the RYU Controller. Based on the research presented, some concerns exist based on the split ratio of the training and testing data.A well-balanced split ratio is crucial for enabling the algorithm to generalize effectively. This enhances the framework's capacity to detect and mitigate attacks. The study also explores the system's application in online services, demonstrating its effectiveness in preventing and mitigating attacks. This translates to a safer and more dependable online service environment.

Fajar investigated the different approaches to mitigate DDoS attacks that originate from various sources [41]. Security issues related to SDN were examined also. Despite ongoing efforts by the security community to develop mechanisms against DDoS attacks, existing techniques have limitations. To address these shortcomings and bolster overall SDN security, the authors propose harnessing the unique strengths of Software-Defined Networking (SDN) for a more robust defense strategy.

Elsayed et al. [4] introduced a new publicly available SDN dataset called InSDN [4], contains 343,939 instances of normal and attack traffic. The normal data is 68,424 instances and the attacks data is 275,515 instances. The dataset also contains 83 features. The dataset includes various attack categories, such as DoS, DDoS, Web Attacks, Botnet, Password-Guessing Attacks, Probes, and Exploitation. The authors evaluated the effectiveness of the dataset using precision, recall, F-score, and training time as performance indicators. When the four publicly available datasets were compared, Ada Boost The results from the To further evaluate the quality of the dataset, eight supervised learning algorithms, including a single Decision Tree (DT), Random Forest (RF), Adaptive Boosting learner (AdaBoost), k-nearest neighbor (KNN), Naive Bayes (NB), linear kernel Support Vector Machine (SVM), radial basis function kernel Support Vector Machine (RBF-SVM), and multi-layer perception model (MLP), were used. The AdaBoost classifier demonstrated the highest performance scores, followed by DT and RF classifiers [4]. The research paper exhibited a thorough examination of its limitations. Several of the limitations addressed significant class imbalance issues in the dataset, potentially leading to elevated false alarm rates and diminished the evaluation accuracy. Additionally, the experimental setup relied on an ONOS controller, thus not comprehensively representing the diversity of SDN controllers and their associated security measures. Lastly, due to hardware constraints, the dataset employed only a single controller. This issue fell short of replicating the full scale of an enterprise architecture with multiple controllers, switches, and nodes.

Tan et al. [42] Researchers analyze the detection and defense mechanisms for DDoS attacks in SDN by leveraging the inherent advantages of SDN combined with machine learning algorithms. Their approach employs a targeted method to detect and defend against DDoS attacks specifically aimed at the SDN controller. Experiments demonstrate that the proposed detection methods yield positive results. Additionally, the detection trigger mechanism effectively identifies abnormal flows while conserving controller resources, and the defense strategy efficiently mitigates DDoS attacks. However, under larger-scale network traffic, the controller's burden increases and the efficiency of DDoS detection decreases. The setup, the SDN controller used was ONOS and Mininet to simulate the underlying network environment. Scapy generates random destination IP addresses to simulate DDoS attacks on the SDN control plane. To detect DDoS attacks, a combined machine learning algorithm based on K-Means and KNN was used. To evaluate the performance of the method the NSL-KDD [27] dataset used. Finally,reserchers test the K-Means algorithm, the KNN algorithm, and the combined K-Means and KNN algorithm, comparing the results with those of the DPTCM-KNN algorithm and KD Tree.

Ali et al. [43]The objective of this systematic review is to identify, evaluate, and discuss recent efforts on machine learning (ML) and deep learning (DL)-based DDoS attack detection strategies in SDN networks. To achieve this, we conducted a systematic review of publications that used ML/DL approaches to identify DDoS attacks in SDN networks between 2018 and early November 2022. The comprehensive search covered several digital libraries (including IEEE, ACM, Springer, and others) and Google Scholar. They analyzed the relevant studies and categorized the results into five key areas: Types of DDoS attack detection in ML/DL approaches. Methodologies, strengths, and weaknesses of existing ML/DL approaches for DDoS attack detection. Benchmarked datasets and classes of attacks used in the existing literature. Preprocessing strategies, hyperparameter values, experimental setups, and performance metrics used in the existing literature. Many studies report accuracy rates exceeding 99%. However, because most of these studies evaluated their models using offline data analysis, performance metrics may differ in real-world or production settings. Notably, existing papers often use different datasets and assessment techniques, complicating direct comparisons of their results.

Gadze, James Dzisi, et al. [44] This paper investigates the potential and efficiency of deep learning-based models, specifically Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN), in detecting and mitigating DDoS attacks. It focuses on TCP, UDP, and ICMP flood attacks targeting the SDN controller. The performance of these models was evaluated based on accuracy, recall, and true negative rate, and compared with classical machine learning models. Additionally, the time taken to detect and mitigate the attacks was detailed. The results show that the RNN LSTM algorithm is a viable deep learning model for detecting and mitigating DDoS attacks in the SDN controller. The proposed model achieved an accuracy of 89.63%, outperforming linear-based models such as SVM (86.85%) and Naive Bayes (82.61%). Although the KNN model, a linear-based algorithm, achieved a higher accuracy of 99.4%, the proposed model offers a good trade-off between precision and recall, making it suitable for DDoS classification. Researchers also observed that the split ratio of the training and testing datasets impacts the performance of deep learning algorithms. A 70/30 split ratio provided the best performance compared to 80/20 and 60/40 splits. Data traffic was generated using the hping3 tool, simulating normal TCP, UDP, and ICMP traffic between two network endpoints. The dataset consisted of 10,031 instances, with 4270 (approximately 43%) being malicious traffic and 5761 (approximately 57%) being normal traffic. This data was used to build binary classification models using various ML algorithms, including K-Nearest Neighbors (KNN), Logistic Regression, Linear SVC, SVC, Decision Tree, Random Forest, Gradient Boosting, and Naive Bayes classifiers (Gaussian, Bernoulli, and Multinomial), as well as the primary algorithms under investigation, RNN LSTM and CNN.

Table 5 provides an overview of the dataset information advantages and disadvantages of artificial intelligence and machine learning approaches in SDN.

2.6 Attacks in vehicular networks using SDN and more

Yu et al. [45] conducted a platform for identifying and mitigating DDoS attacks in vehicular networks that operate on an SDN-based network was developed. The platform utilizes a flow table detection technique. To evaluate the effectiveness of the platform, it was tested on the DARPA 1999 [46], DARPA 2000 [47], and CAIDA DDoS 2007 [48] datasets. Furthermore, a Mininet-based SDN testbed was created to replicate the environment, which used the floodlight controller. This research addresses concerns about SDN controller overload caused by an influx of PACKET_IN messages. The approach successfully mitigated this issue by utilizing the PACKET_IN trigger to expedite response times. Additionally, the study employed a variety of transport protocols, including TCP, UDP, and ICMP, in DDoS attack scenarios. It implemented a feature selection method grounded in correlation coefficients, enhancing the effectiveness of their analysis.

Maity et al. developed a DDoS detection system using probability modeling that utilizes TCP flag distributions in the SDN controller rather than traffic speed to reduce false negatives. The system was assessed using the DARPA 2009 dataset [49] and tested in an SDN environment with the aid of Mininet and a Floodlight controller. The researchers encountered limitations in their experimentation due to the absence of diverse network traffic types required for comprehensive system testing. Furthermore, the system's robustness against DDoS attacks fell short of expectations. The researchers express their aspirations to transition towards real-time implementation in a distributed environment in future work.

Table 6 provides an overview of the dataset information advantages and disadvantages of Attacks in Vehicular Networks using SDN and more.

Table 7 provides an overview of prior studies, showcasing their methodologies, utilized datasets, and key highlights.

3 Datasets

Open datasets or publicly available datasets are of utmost importance for the advancement of research on SDN detection and mitigation. They serve as a shared foundation, enabling researchers to build upon existing knowledge and enhance the reproducibility of experiments. The availability of open datasets to the public facilitates the creation of a standardized dataset utilized by the community for evaluating various use cases and applications. Among the featured datasets, those specifically focused on DDoS attacks are invaluable resources for research in this domain.

DARPA98 [46]: The DARPA 98 dataset simulates typical traffic patterns found on a government website hosting hundreds of users across thousands of hosts. During its creation, over 300 instances of 38 distinct automated attacks

Table 5 Summary of the dataset information advantages and disadvantages of artificial intelligence and machine learning approaches in SDN

Literature	Dataset information	Advantages	Disadvantages
Gebremariam et al. [7]	No dataset	Strategically storing data across the network safeguards against failures and attacks, ensuring data integrity and resilience	Limited network resources and the absence of easily available datasets
Zhou et al. [36]	No dataset	This approach intends to mislead attackers and limit disruptions to the system	Vulnerable to resource-exhaustion attacks
Sahoo et al. [37]	NSL-KDD [27]	The SVM acts as the classifier, while KPCA is used to extract features from the dataset, and GA optimizes the SVM	Uncertainties emerge when considering environments with multiple controllers
Tan et al. [38]	NSL-KDD [27]	The effectiveness of the framework has been evaluated using both the NSL-KDD [27] dataset and a SDN environment created with Mininet	The controller hinders its ability to effectively detect DDoS attacks
Revathi et al. [39]	KDD99 [40]	The study also explores the system's application in online services, demonstrating its effectiveness in preventing and mitigating attack	Some concerns exist based on the split ratio of the training and testing data
Fajar et al. [41]	No dataset	The authors propose harnessing the unique strengths of Software-Defined Networking (SDN) for a more robust defense strategy	None
Elsayed et al. [4]	InSDN [4]	The AdaBoost classifier demonstrated the highest performance scores, followed by DT and RF classifiers	The dataset employed only a single controller
Tan et al. [42]	NSL-KDD [27]	Experiments demonstrate that the proposed detection methods yield positive results	Under larger-scale network traffic, the controller's burden increases and the efficiency of DDoS detection decreases
Ali et al. [43]	No dataset	Studies report accuracy rates exceeding 99%	These studies evaluated their models using offline data analysis, performance metrics may differ in real-world or production settings
Gadze et al. [44]	No dataset	70/30 split ratio provided the best performance compared to 80/20 and 60/40 splits	No dataset hard to replicate

Table 6 Summary of the dataset information advantages and disadvantages of attacks in vehicular networks using SDN and more

Literature	Dataset information	Advantages	Disadvantages
Yu et al. [45]	CAIDA DDoS 2007 [48], DARPA 99 [47], DARPA 2000 [50]	Approach successfully mitigated this issue by utilizing the PACKET_IN trigger to expedite response times	None
Maity et al. [49]	DARPA 2009 [51]	None	The system's robustness against DDoS attacks fell short of expectations

were executed against UNIX hosts acting as victims. MITS Lincoln Lab generated this dataset over a span of 7 weeks, comprising 4 GB of binary data. It replicates a scenario resembling a small Air Force network linked to the Internet, serving as a benchmark for evaluating intrusion detection systems [9].

DARPA99 [47]: This comprehensive evaluation, which resulted in the creation of the DARPA 99 dataset, engaged eight research sites in analyzing the detection capabilities and false alarm rates of intrusion detection systems. The assessment unfolded within a controlled network environment, featuring host computers exposed to attacks and sophisticated traffic generators mimicking live traffic patterns reminiscent of those observed in a small Air Force base. The traffic emulation accurately replicated the behaviors of hundreds of users and thousands of hosts. Over a period of 3 weeks for training data and 2 weeks for test data, the evaluation executed over 200 instances of 58 distinct attack types against both UNIX and Windows NT hosts. This endeavor culminated in the generation of the DARPA 99 dataset, a valuable resource for cybersecurity research and development.

DARPA2000 [50]: In 1999, Lincoln Lab introduced the DARPA 2000 intrusion detection dataset, which spans 5 weeks and encompasses simulated network traffic. This dataset comprises two distinct attack scenarios: LLDoS 1.0 and LLDoS 2.0.2. Both scenarios feature identical attack and victim hosts. Each scenario encompasses Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L) attacks, Probes, and Data. These attacks unfold across five stages: host scanning, transmission of vulnerability query packets, exploitation of vulnerabilities for privilege escalation, installation of DDoS software on victim hosts, and execution of DDoS attacks on target hosts.

DARPA2009 [51]: The DARPA 2009 dataset was crafted using synthesized traffic to replicate interactions between a /16 subnet (172.28.0.0/16) and the Internet. Captured over a span of 10 days from November 3rd to November 12th, 2009, this dataset encompasses synthetic HTTP, SMTP, and DNS background traffic. It features a diverse array of security events and attack types, illustrating contemporary attack methodologies. These include various distributed denial of service (DDoS) attacks and worms designed to demonstrate different propagation characteristics. Comprising 7000 pcap files, the dataset totals around 6.5 terabytes in size.

KDD99 [40]: To obtain a subset of the DARPA 1998 dataset, simulation of a US Air Force LAN was conducted, inducing various types of attacks. This simulation utilized 9 weeks of TCP dump data collected at MIT Lincoln Laboratory. The resulting subset, known as the KDDCup dataset, comprises approximately 4,900,000 individual instances described by 41 features. These instances are classified as either normal or representing an intrusion. The training data includes 24 labels, categorized as normal behavior and 23 distinct types of attacks. Additionally, the test data incorporates 14 additional attack types. These attacks are classified into four categories: User to Root (U2R), Denial of Service (DoS), Remote to Local (R2L), and probing attacks.

NSL-KDD [27]: After removing redundant and duplicate records from the training and test data of the KDDCup dataset, the NSL-KDD dataset was developed, comprising only selected and essential records. The entire official dataset was then divided into a training dataset containing 125,973 records and a test dataset containing 22,544 records. Among the total of 37 attacks, 27 attacks were allocated to the testing dataset, while 23 attacks were assigned to the training dataset for experimentation. The NSL-KDD dataset maintains the same number of features as the KDDCup dataset, with 41 features and 5 attack classes. These attacks are categorized into four groups: Probe attack, Denial of Service attack (DoS), User to Root (U2R), and Remote to Local (R2L). The dataset includes one normal class and four different types of attacks. Additionally, it features a binary class attribute and a reasonable number of training and test instances.

DEFCON-8 [52]: The DEFCON-8 version encompasses port scanning and buffer overflow-based attacks, while another version includes attacks on the FTP protocol, bad packets, port scans, and sweep attacks. However, this dataset has limitations due to the disparity between real-time and normal traffic observed during Capture the Flag (CTF) competitions, which impacts the evaluation of Intrusion Detection Systems (IDS).

Table 7 Summary of previous works based on methods, datasets used, and highlights

Authors	Methods	Datasets used	Highlights
Sarica et al. [1]	Random forest classifier	SDN Dataset [8]	Their approach relied on automated flow feature extraction and highly accurate classification of network flows by a random forest classifier in the SDN application layer, for detecting diverse classes of attacks and taking corrective actions through the installation of new flow rules with high priority at the data plane
Galeano-Brajones et al. [13]	An entropy-based algorithm	Bot-Iot [12]	Three scenarios were presented. The first one displayed the bandwidth consumption and entropy values during a DoS attack and the mitigation process using a Ryu controller and mininet as the testbed. The next two scenarios utilized the Bot-IoT dataset to evaluate the detection and mitigation process of DoS attacks in an IoT environment. Lastly, the final scenario focused on DDoS attack detection and mitigation in an IoT environment
Wang et al. [3]	CNN-ELM and extreme learning machine	InSDN [4], CICIDS2017 [5]	Researchers have created a traceback mitigation method for SDN networks. By using ten attributes they were able to trace the path of the attack source. Researchers also proposed a deep learning hybrid model CNN-ELM—convolutional neural network and extreme learning machine
Sharma et al. [19]	Deep belief network	UNB ISCX [20]	Researchers created an attack and reaction system based on an anomaly called OpCloudSec. The system utilized the Deep Belief Network to create the attack prevention model
Salem et al. [14]	Two-phases algorithm	No dataset	Researchers created a two-phase detection approach using two algorithms to help in the early detection of a DDoS attack. This approach assigns a trust value to each user in the system. Next, as the users interact with the controller sending requests, the requests are recorded and documented by amount
de Assis et al. [15]	CNN	CICDDoS 2019 [16]	Researchers proposed a defense system for SDN networks that inspects traffic in one-second time intervals. The system detects and mitigates the occurrence of DDoS attacks on the controller and on an external server. The Mitigation module was described, and a game-theoretical approach was presented as an example of operation
Ravi et al. [24]	LEDEM	UNB ISCX [20]	Researchers have introduced an innovative approach, coined as Learning-Driven Detection Mitigation (LEDEM), for identifying Distributed Denial-of-Service (DDoS) attacks by utilizing a semi-supervised machine learning algorithm
Gad et al. [30]	XGBoost	ToN-IoT [31]	An IDS for VANET was created by researchers utilizing the ToN-IoT dataset. To enhance the model's efficiency and speed, Chi^2 was applied to the dataset for feature selection, resulting in a reduction of features to 20
Hanif et al. [32]	ANN	NSL-KDD [27]	An intrusion detection system (IDS) capable of detecting attacks and making prompt decisions has been developed by researchers. The system will be integrated into an IoT controller to determine whether incoming data is benign or malicious, with malicious data being discarded immediately
Wani et al. [18]	SDIoT-DDoS-DA	No dataset	A security mechanism for the detection and alleviation of DDoS attacks on SDN networks, called SDIoT-DDoS-DA, has been proposed by researchers
Sahoo et al. [37]	SVM, KPCA, and GA	NSL-KDD [27]	A technique has been developed by researchers to detect and mitigate DDoS attacks, which employs a machine learning model combining Support Vector Machine (SVM), kernel principal component analysis (KPCA), and genetic algorithm (GA)

Table 7 (continued)

Authors	Methods	Datasets used	Highlights
Yu et al. [45]	A flow table detection technique	CAIDA DDoS 2007 [48], DARPA 99 [47],DARPA 2000 [50]	A platform was created by researchers to identify and counter DDoS attacks in vehicular networks that operate over an SDN-based network. The platform employed a flow table detection technique
Maity et al. [49]	A probabilistic model	DARPA 2009 [51]	A team of researchers devised a DDoS detection system based on probability modeling. Instead of relying on traffic speed, this model uses TCP flag distributions in the controller to reduce the false negative rate
Yeom et al. [23]	LSTM	No dataset	Researchers proposed an LSTM (Long Short-Term Memory) DDoS attack detection framework that works on the source side. To mitigate performance degradation caused by irregular network traffic behavior the framework applies LSTM-based adaptive thresholds to each source-side network

UNIBS [53]: The UNIBS packet traces were collected over three consecutive working days from the edge router of the University of Brescia campus network in Italy. This dataset includes traffic captured using 20 workstations, each running the GT (Ground Truth) client daemon. The traffic was collected via tcpdump on the faculty router, a dual Xeon Linux box connecting the local network to the Internet through a dedicated 100 Mb/s uplink. The dataset comprises 27 GB of data, primarily TCP (99%) and UDP traffic, amounting to approximately 79,000 flows. The captured traffic includes web (HTTP and HTTPS), mail (POP3, IMAP4, SMTP, and their SSL variants), Skype, peer-to-peer applications like BitTorrent and eDonkey, and other protocols such as FTP, SSH, and MSN.

CAIDA DDOS 2007 [48]: The dataset comprises network traffic traces captured during Distributed Denial-of-Service (DDoS) attacks, collected in 2007. These attacks aim to disrupt the normal flow of traffic to a targeted computer or network by inundating it with a barrage of network packets, thereby obstructing legitimate traffic from reaching its intended destination. However, one drawback of the CAIDA dataset is its lack of attack diversity. Furthermore, the collected data lacks comprehensive features from the entire network, posing challenges in distinguishing between abnormal and normal traffic patterns.

LBNL [54]: The LBNL dataset features anonymized traffic, containing only header data. Generated at the Lawrence Berkeley National Laboratory, the dataset captures real outbound, inbound, and routing traffic from two edge routers. It does not include labeled data, nor were any additional features created. The primary applications in the internal and external traffic include web, email, and name services, while internal hosts also used applications such as Windows services, network file services, and backup. Malicious traffic mainly consists of failed incoming TCP SYN requests, indicating TCP port scans targeting LBNL hosts. However, the dataset also includes some outgoing TCP scans. Most of the UDP traffic (both incoming and outgoing) involves successful connections, with hosts replying to received UDP flows.

TUIDS [55]: This dataset was generated by professors from Tezpur University, India, featuring DoS, Probing, Scan, U2R, and DDoS attack scenarios conducted in a testbed environment. The testbed network comprises 250 hosts, 15 L2 switches, 8 L3 switches, 3 wireless controllers, and 4 routers across 5 different networks. The flow-level data includes only the features produced by the flow-capturing process, with no additional features created.

InSDN [4]: The InSDN dataset is designed to be realistic for intrusion detection system (IDS) evaluation. It includes a variety of recent attacks, like denial-of-service (DoS), distributed denial-of-service (DDoS), brute force, malware, probes, exploits, and web attacks. Additionally, it incorporates common application traffic for HTTPS, HTTP, DNS, email, FTP, and SSH. The dataset uses multiple attack scenarios, combining internal SDN attacks with external ones. To capture rich network details, it leverages the CICFlowMeter tool to extract over 80 statistical features. In total, there are 343,939 entries of both normal and attack traffic, making it highly comparable to real-world network attack data.

UNB ISCX [20]: The UNB ISCX dataset was created at the Canadian Institute for Cybersecurity. It uses profiles to define attack and distribution techniques in a network. Real traces were analyzed to create accurate profiles for evaluating intrusion detection systems. Data was collected using a real-time testbed with multi-stage attacks, employing two profiles: α (based on specific attacks) and β (based on filtered traffic traces). These profiles generated real-time traffic for protocols like HTTP, SMTP, SSH, IMAP, POP3, and FTP. Various multi-stage attack scenarios were included to produce malicious traffic.

UNSW-NB15 [33]: A dataset developed at UNSW Canberra utilized IXIA PerfectStorm to generate a mixture of benign and attack traffic. This resulted in a 100 GB dataset in the form of PCAP files, featuring 49 novel attributes and totaling 2,540,044 records stored in four CSV files. The dataset was created for the generation and validation of intrusion detection systems, designed within a synthetic environment to simulate attack activities.

Bot-Iot [12]: The Bot-IoT dataset combines normal IoT-related and other network traffic with various types of attack traffic commonly used by botnets. Developed on a realistic testbed, it includes labeled features indicating attack flows, attack categories, and subcategories for multiclass classification purposes. Additional features were generated to enhance the predictive capabilities of classifiers trained on this model. Through statistical analysis, a subset of the original dataset comprising the 10 best features was produced. Created in 2018, the Bot-IoT dataset includes both IoT and normal network user traffic, with a total of 46 features and six types of attacks: service scanning, OS fingerprinting, DoS, DDoS, keylogging, and data theft. However, the dataset is imbalanced, with significantly fewer records for some attack types—118 records for data theft and 1469 for keylogging—insufficient for most machine learning algorithms. The dataset's notable feature is the inclusion of IoT traffic, unlike most existing intrusion detection datasets. Despite its richness in features, the dataset suffers from a low number of samples for certain attack types and normal traffic.

CICIDS2017 [5]: The CICIDS2017 dataset encompasses both benign behaviors and details of emerging malware attacks, such as Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS (Sharafaldin et al. 2018). It is meticulously labeled based on timestamps, source and destination IP addresses, source and destination ports, protocols, and specific attack types. To compile this dataset, a comprehensive network topology was established, comprising Modems, Firewalls, Switches, Routers, and nodes running various operating systems, including Microsoft Windows (such as Windows 10, Windows 8, Windows 7, and Windows XP), Apple's macOS iOS, and Linux open-source operating systems. In total, the dataset contains 80 network flow features extracted from the captured network traffic.

CICDDoS 2019 [16]: The CICDDoS2019 dataset is a publicly available for researchers and security professionals on the frontlines of combatting DDoS attacks. Designed specifically to evaluate Intrusion Detection Systems (IDS) against these ever-evolving threats, CICDDoS2019 offers a realistic training ground. The dataset incorporates the latest and most common DDoS attack types, along with benign traffic, mimicking real-world network conditions. To provide a rich tapestry for analysis, CICDDoS2019 leverages CICFlowMeter to extract over 80 statistical features from the network traffic. With a massive dataset exceeding 343,939 entries encompassing both normal and attack traffic, CICDDoS2019 empowers comprehensive training and testing of IDS models, making it a valuable resource for fortifying defenses against DDoS attacks.

IoT-23 [35]: For researchers on the cybersecurity frontlines of securing the Internet of Things (IoT), the IoT-23 dataset is a game-changer. IoT-23 dives deep into the world of malware specific to IoT devices. The dataset provides a wealth of real-world network traffic captured from both compromised and healthy devices. This realistic foundation allows researchers to develop machine learning algorithms that can effectively distinguish malicious activity. With captures involving 20 different malware samples, IoT-23 offers a broad spectrum of threats for analysis. Furthermore, the data is meticulously labeled, making it easy for machine learning models to learn and identify these threats. The freely available nature of IoT-23 makes it an accessible resource for researchers to leverage in their fight to secure the ever-expanding world of IoT devices.

ToN-IoT [31]: Developed by UNSW Canberra IoT Labs and the Cyber Range, it offers a comprehensive perspective on IIoT security by capturing a variety of data streams within an IIoT system. This includes telemetry data from connected devices, logs from both Linux and Windows operating systems, and network traffic data. This richness, known as heterogeneous data, mirrors real-world scenarios and empowers in-depth analysis. The dataset is conveniently available in CSV format at the ToN-IoT repository. Furthermore, ToN-IoT goes beyond raw data by incorporating labels for normal and attack behavior. It even provides a dedicated "attack-type" sub-category, pinpointing specific attacks like ransomware, password brute-forcing, scans, denial-of-service variations, data injection, backdoors, Cross-site Scripting (XSS), and man-in-the-middle (MITM) attacks. These attacks were meticulously simulated within the IIoT network targeting diverse IoT and IIoT sensors. This wealth of data empowers researchers and security professionals to develop and test cutting-edge, AI-powered security solutions specifically designed to safeguard IIoT environments.

SDN Dataset [8]: This dataset offers two parts, each modeling a different type of IoT network: static and dynamic. The static network dataset contains 27.9 million data records, while the dynamic network dataset has 30.2 million records. Both datasets include a variety of cyberattacks alongside benign traffic. They encompass 33 features and represent five distinct attack types: Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), port scanning, OS fingerprinting, and fuzzing attacks. The network environment was virtualized using Mininet, controlled by an ONOS controller, and connected through an Open vSwitch. The static network included five simulated IoT devices, while the dynamic network began with ten devices, with two being powered down at various points during recording. This variation allows for the simulation of different network scenarios.

The utilization of open datasets allows for the replication of models and ensures consistent results across different studies. This promotes transparency and facilitates the advancement of the field by enabling researchers to build upon existing methodologies. Table 8 presents a review of the open datasets that can be accessed online, serving as fundamental pillars for research on detection and mitigation strategies.

4 Distributed denial of service mitigations and detection strategies

DDoS attacks continue to pose significant challenges for SDN-implemented networks. Researchers have made significant progress in developing various methods to mitigate and detect these attacks. The information in this section presents an overview of recent works that focus on effective mitigation and detection strategies. It is worth noting that these techniques are continuously evolving as DDoS attacks become increasingly sophisticated.

Table 8 Overview of open datasets

Dataset	Research paper using dataset
DARPA 98 [46]	As mentioned in [9]
DARPA 99 [47]	[45]
DARPA 2000 [50]	[45]
DARPA 2009 [51]	[49]
KDD99 [40]	[4] and [39]
NSL-KDD [27]	[4, 32, 37] and [38]
DEFCON-8 [52]	As mentioned in [9]
UNIBS [53]	As mentioned in [9]
CAIDA DDoS 2007 [48]	[45]
LBNL [54]	As mentioned in [9]
TUIDS [55]	As mentioned in [9]
InSDN [4]	[4] and [3]
UNB ISCX [20]	[19] and [24]
UNSW-NB15 [33]	[32]
Bot-Iot [12]	[9] and [13]
CICIDS2017 [5]	[4] and [3]
CICDDoS 2019 [16]	[15]
IoT-23 [35]	[34]
ToN-IoT [31]	[30]
SDN Dataset [8]	[1]

4.1 Internet of Things networks managed by SDN

Sarica et al. [1]: Their strategy is based on the automated extraction of flow features and precise classification of network flows using a random forest classifier in the SDN application layer. This allowed them to identify a wide range of attacks and initiate corrective measures by establishing new high-priority flow rules in the data plane.

4.2 Detection and mitigation of DoS, DDoS using SDN

Galeano-Brajones et al. [13]: Three scenarios are outlined: The first one gauges bandwidth consumption during a DoS attack and its mitigation, incorporating entropy values. This scenario employs a Ryu controller and Mininet. The following two scenarios utilize the Bot-IoT dataset to assess DoS attack detection and mitigation within an IoT framework. The final scenario places its focus on the detection and mitigation of DDoS attacks in an IoT environment.

Wang et al. [3]: Researchers developed a traceback method for SDN networks using ten attributes to trace attack sources and introduced a hybrid deep learning model, CNN-ELM (Convolutional Neural Network and Extreme Learning Machine).

Sharma et al. [19]: An attack and response system named OpCloudSec was devised by researchers, utilizing a Deep Belief Network to construct the attack prevention model.

Salem et al. [14]: Researchers developed a two-phase approach using algorithms to early detect DDoS attacks. It assigns trust values to system users and records their interaction requests with the controller.

De Assis et al. [15]: Researchers introduced a one-second interval traffic inspection defense system for SDN networks, capable of detecting and mitigating DDoS attacks on both the controller and an external server. They described the Mitigation module and illustrated its operation using a game-theoretical approach.

Wani [18]: Researchers proposed SDIoT-DDoS-DA, a security mechanism to detect and mitigate DDoS attacks on SDN networks.

4.3 Deep learning and machine learning detection and mitigation techniques with DDoS

Ravi et al. [24]: Researchers introduced LEDEM, a novel approach for detecting DDoS attacks using a semi-supervised machine learning algorithm.

4.4 IDS using XGBoost algorithm

Gad et al. [30]: Researchers developed a VANET IDS using the ToN-IoT dataset and improved its efficiency by applying Chi^2 feature selection, reducing features from 108 to 20.

Hanif et al. [32]: Researchers created an IDS for rapid attack detection and decision-making. It was integrated into an IoT controller to assess incoming data, discarding malicious data instantly.

4.5 Artificial intelligence and machine learning approaches in SDN

Sahoo et al. [37]: Researchers devised a technique to detect and mitigate DDoS attacks using a machine learning model that combines SVM, KPCA, and GA.

4.6 Attacks in vehicular networks using SDN and more

Yu et al. [45]: Researchers built a platform to detect and counter DDoS attacks in SDN-based vehicular networks using a flow table detection technique.

Maity et al. [49]: Researchers built a platform to detect and counter DDoS attacks in SDN-based vehicular networks using a flow table detection technique.

Researchers have developed numerous techniques for detecting and mitigating attacks, employing a wide range of machine learning models or combining multiple models to enhance accuracy. The section below also showcases frameworks for attack detection and mitigation. The primary emphasis of these frameworks lies in the field of DDoS mitigation and detection.

4.7 Deep learning and machine learning detection and mitigation techniques with DDoS

Yeom et al. [23]: Researchers introduced an LSTM-based DDoS detection framework that operates on the source side and mitigates performance issues due to irregular network traffic by employing adaptive LSTM-based thresholds for each source-side network.

4.8 Detection and mitigation of DoS, DDoS using SDN

Bawany et al. [6]: Researchers created ProDefence, a modular framework to detect and counter DDoS attacks in a large-scale SDN-based smart city network, featuring components like a traffic flow collector, policy engine, attack detector, and mitigation engine.

4.9 Artificial intelligence and machine learning approaches in SDN

Zhou et al. [36]: Researchers crafted a framework for automatic generation, deployment, and adjustment of proactive defense measures in IoT. It relies on MTD techniques and cyber deception to confound attackers and minimize disruptions.

Tan et al. [38]: For DDoS detection, a data plane trigger mechanism was used alongside a controller-based machine learning algorithm employing KNN and K-Means to spot suspicious flows.

Revathi et al. [39]: Researchers created a framework using DSM-SVM to predict and counter SDN-based DDoS attacks.

4.10 Internet of Things networks managed by SDN

Bhayo et al. [10]: Researchers created a DDoS attack detection framework using C-DAD (Counter-based DDoS Attack Detection).

Yin et al. [11]: Researchers devised a framework and algorithm to detect and mitigate DDoS attacks in an SD-IoT environment. The algorithm utilized cosine similarity to compare packet-in rate vectors at the boundary switch.

5 Discussion

This review delves into cutting-edge applications of machine learning techniques and frameworks in DDoS in SDN. It recognizes the rapidly evolving nature of this field, highlighting the importance of referencing the latest research to establish new benchmarks and propel advancements.Research has prioritized applying machine learning to SDN security, but a key challenge is the lack of publicly available datasets. These datasets are crucial for advancing research in this field [7]. In comparison, our survey identifies a limited number of open datasets specifically related to SDN. This could be attributed to the fact that, at the time of the survey, there were fewer SDN datasets available. However, these datasets often lacked the desired testbed environments, despite containing the relevant attacks.

Furthermore, researchers conducted a survey on SDN mitigation techniques, and the security challenges associated with SDN [41]. While the papers surveyed may be slightly outdated, they provided a comprehensive understanding of various types of DDoS attacks on SDN. Our survey further augments the existing reservoir of knowledge by introducing additional attack detection and mitigation techniques.

The literature review indicates that certain open datasets were not analyzed in [41]. There could be several reasons for this discrepancy. Firstly, it is possible that these datasets are outdated and do not include the attacks or testbed setups relevant to the current research. Additionally, some of the older datasets may be insufficient in size for the application of modern machine-learning techniques. Despite their limited utilization in present-day research, it is still valuable to acknowledge and comprehend the previous standards set by these datasets.

Moving forward, as machine learning and SDN technologies continue to advance, it is anticipated that a broader array of up-to-date and relevant datasets will become accessible, facilitating more robust investigations and evaluations of attack detection and mitigation methods. As the field progresses, it will be crucial to strike a balance between honoring the insights garnered from earlier studies and harnessing the potential of cutting-edge datasets and methodologies to address the evolving landscape of DDoS attacks in SDN environments.

6 Conclusions and future work

In conclusion, this paper provides a comprehensive overview of open datasets featuring DDoS attacks, as well as various techniques and frameworks for attack detection and mitigation. The availability of these open datasets is crucial for establishing a standardized platform that enables researchers to contribute to the field. The effectiveness and accuracy of the research in attack detection and mitigation are demonstrated through the utilization of these open datasets. The paper reinforces its findings with practical case studies, showcasing real examples of how researchers have leveraged open datasets to develop and validate mitigation techniques. This bridges the gap between theoretical understanding and practical application. Furthermore, the paper acknowledges the use of SDN datasets to incorporate more current techniques and frameworks.

Additionally, the paper compares its findings with two other survey papers that delve into the state of open datasets and attack detection and mitigation techniques. This comparison provides valuable insights and further strengthens the understanding of the research landscape.

Moreover, efforts should be directed towards the creation of comprehensive benchmark datasets that encapsulate a wide array of attack scenarios, network topologies, and SDN configurations. The development of standardized evaluation metrics would further enhance the comparability and reproducibility of results across different studies. Collaborative initiatives involving academia, industry, and cybersecurity organizations could play a pivotal role in addressing the scarcity of open datasets.

For future research, there are plans to develop an SDN dataset specifically focusing on DDoS attacks. This dataset will be created and evaluated using an SDN testbed, and its results will be compared to other SDN datasets that already contain DDoS attack data. This initiative aims to contribute to the advancement of research in the field of SDN and enhance the effectiveness of attack detection and mitigation techniques.

Acknowledgements This research is based upon work supported by the ONR (Award No. N00014-22-1-2724). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.

Author contributions WH wrote the main manuscript text, all authors reviewed the manuscript. JM, SP, CC, YA, KR made substantial contributions to the conception or design of the work; or the acquisition, analysis, or interpretation of data; or the creation of new software used in the work; WH, YA, JM, KR, DL drafted the work or revised it critically for important intellectual content; KR, WH approved the version to be published; and agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Funding No funding was obtained for this study.

Availability of data and materials The data used for this experiment are available in public repositories. The detailed information is given in the result analysis section.

Declarations

Competing interests The authors declare that they have no competing interests.

References

1. Sarica AK, Angin P. Explainable security in SDN-based IoT networks. Sensors. 2020;20(24):7326. https://doi.org/10.3390/s20247326.
2. Stephen MS. Distributed denial of service: taxonomies of attacks, tools and countermeasures. Electrical Engineering Princeton University; 2004.
3. Wang J, Wang L. SDN-defend: a lightweight online attack detection and mitigation system for DDoS attacks in SDN. Sensors. 2022;22(21):8287. https://doi.org/10.3390/s22218287.
4. Elsayed MS, Le-Khac N-A, Jurcut AD. InSDN: a novel SDN intrusion dataset. IEEE Access. 2020;8:165263–84. https://doi.org/10.1109/ACCESS.2020.3022633.
5. CIC-IDS2017. University of New Brunswick est.1785. (n.d.-a). https://www.unb.ca/cic/datasets/ids-2017.html.
6. Bawany NZ, Shamsi JA, Salah K. DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arab J Sci Eng. 2017;42:425–41.
7. Gebremariam AA, Usman M, Qaraqe M. Applications of artificial intelligence and machine learning in the area of SDN and NFV: a survey. In: 2019 16th International multi-conference on systems, signals & devices (SSD), Istanbul, Turkey; 2019, pp. 545–549. https://doi.org/10.1109/SSD.2019.8893244.
8. Sarica AK, Angin P. A novel SDN dataset for intrusion detection in IoT networks. In: 2020 16th International conference on network and service management (CNSM); 2020, pp. 1–5. https://doi.org/10.23919/CNSM50824.2020.9269042.
9. Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT Dataset; 2018.
10. Bhayo J, Hameed S, Shah SA. An efficient counter-based DDoS attack detection framework leveraging software defined IoT (SD-IoT). IEEE Access. 2020;8:221612–31.
11. Yin D, Zhang L, Yang K. A DDoS attack detection and mitigation with software-defined Internet of Things framework. IEEE Access. 2018;6:24694–705.
12. The Bot-IOT dataset. The Bot-IoT Dataset | UNSW Research. (n.d.). https://research.unsw.edu.au/projects/bot-iot-dataset.
13. Galeano-Brajones J, Carmona-Murillo J, Valenzuela-Valdés JF, Luna-Valero F. Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: an experimental approach. Sensors. 2020;20(3):816.
14. Salem FM, Youssef H, Ali I, Haggag A. A variable-trust threshold-based approach for DDoS attack mitigation in software-defined networks. PLoS ONE. 2022;17(8):e0273681.
15. de Assis MV, Carvalho LF, Rodrigues JJ, Lloret J, Proença ML Jr. Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. Comput Electr Eng. 2020;86:106738.
16. CIC-DDoS2019. University of New Brunswick est.1785. (n.d.-a). https://www.unb.ca/cic/datasets/ddos-2019.html.
17. Kiani R, Bohlooli A. Distributed rule anomaly detection in SDN-based IoT. In: 2021 5th International conference on Internet of Things and applications (IoT). IEEE; 2021, pp. 1–6.
18. Wani A, Revathi S. DDoS detection and alleviation in IoT using SDN (SDIoT-DDoS-DA). J Inst Eng (India) Ser B. 2020;101(2):117–28. https://doi.org/10.1007/s40031-020-00442-z.

19. Sharma PK, Singh S, Park JH. OpCloudSec: open cloud software-defined wireless network security for the Internet of Things. Comput Commun. 2018;122:1–8.
20. UNB ISCX. University of New Brunswick est.1785. (n.d.-b). https://www.unb.ca/cic/datasets/ids.html.
21. Singh J, Behal S. Detection and mitigation of DDoS attacks in SDN: a comprehensive review, research challenges and future directions. Comput Sci Rev. 2020;37:100279.
22. Singh MP, Bhandari A. New-flow based DDoS attacks in SDN: taxonomy, rationales, and research challenges. Comput Commun. 2020;154:509–27.
23. Yeom S, Choi C, Kim K. LSTM-based collaborative source-side DDoS attack detection. IEEE Access. 2022;10:44033–45.
24. Ravi N, Shalinie SM. Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. IEEE Internet Things J. 2020;7(4):3559–70.
25. Santos R, et al. Machine learning algorithms to detect DDoS attacks in SDN. Concurr Comput Pract Exp. 2020;32(16):e5402. https://doi.org/10.1002/cpe.5402.
26. Nadeem MW et al. DDoS detection in SDN using machine learning techniques. Comput Mater Continua. 71(1) (2022). https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC-71-1/TSP_CMC_21669/TSP_CMC_21669.pdf.
27. NSL-KDD Dataset. University of New Brunswick est.1785. (n.d.). https://www.unb.ca/cic/datasets/nsl.html.
28. Banitalebi Dehkordi A, Soltanaghaei MR, Boroujeni FZ. The DDoS attacks detection through machine learning and statistical methods in SDN. J Supercomput. 2021;77(3):2383–415.
29. Perez-Diaz JA, et al. A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access. 2020;8:155859–72.
30. Gad AR, Nashat AA, Barkat TM. Intrusion detection system using machine learning for vehicular ad hoc networks based on ToN-IoT dataset. IEEE Access. 2021;9:142206–17.
31. The Ton_IoT datasets. The TON_IoT Datasets | UNSW Research. (n.d.). https://research.unsw.edu.au/projects/toniot-datasets.
32. Hanif S, Ilyas T, Zeeshan M. Intrusion detection in IoT using artificial neural networks on UNSW-15 dataset. In: 2019 IEEE 16th international conference on smart cities: improving quality of life using ICT & IoT and AI (HONET-ICT). IEEE; 2019, pp. 152–156.
33. The UNSW-NB15 Dataset | UNSW Research—UNSW sites. (n.d.). https://research.unsw.edu.au/projects/unsw-nb15-dataset.
34. da Cruz MA, Abbade LR, Lorenz P, Mafra SB, Rodrigues JJ. Detecting compromised IoT devices through XGBoost. IEEE Trans Intell Transp Syst. 2022;24:15392–9.
35. IOT-23 dataset: A labeled dataset of malware and benign IOT traffic. Stratosphere IPS. (n.d.). https://www.stratosphereips.org/datasets-iot23.
36. Zhou Y, Cheng G, Yu S. An SDN-enabled proactive defense framework for DDoS mitigation in IoT networks. IEEE Trans Inf Forensics Secur. 2021;16:5366–80.
37. Sahoo KS, Tripathy BK, Naik K, Ramasubbareddy S, Balusamy B, Khari M, Burgos D. An evolutionary SVM model for DDoS attack detection in software-defined networks. IEEE Access. 2020;8:132502–13.
38. Tan L, Pan Y, Wu J, Zhou J, Jiang H, Deng Y. A new framework for DDoS attack detection and defense in SDN environment. IEEE Access. 2020;8:161908–19.
39. Revathi M, Ramalingam VV, Amutha B. A machine learning based detection and mitigation of the DDOS attack by using SDN controller framework. Wirel Pers Commun. 2021;127:2417–41.
40. KDD Cup 1999 Dataset. KDD Cup 1999 Data. (n.d.). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
41. Fajar AP, Purboyo TW. A survey paper of distributed denial-of-service attack in software defined networking (SDN). Int J Appl Eng Res. 2018;13(1):476–82.
42. Tan L, et al. A new framework for DDoS attack detection and defense in SDN environment. IEEE Access. 2020;8:161908–19.
43. Ali TE, Chong Y-W, Manickam S. Machine learning techniques to detect a DDoS attack in SDN: a systematic review. Appl Sci. 2023;13(5):3183.
44. Gadze JD, et al. An investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers. Technologies. 2021;9(1):14.
45. Yu Y, Guo L, Liu Y, Zheng J, Zong YUE. An efficient SDN-based DDoS attack detection and rapid response platform in vehicular networks. IEEE Access. 2018;6:44570–9.
46. 1998 DARPA Intrusion Detection Evaluation Dataset. MIT Lincoln Laboratory. (n.d.-a). https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset.
47. 1999 DARPA Intrusion Detection Evaluation Dataset. MIT Lincoln Laboratory. (n.d.-b). https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.
48. Center of applied internet data analysis. https://www.caida.org/data/
49. Maity P, Saxena S, Srivastava S, Sahoo KS, Pradhan AK, Kumar N. An effective probabilistic technique for DDoS detection in OpenFlow controller. IEEE Syst J. 2021;16(1):1345–54.
50. 2000 DARPA intrusion detection scenario specific datasets. MIT Lincoln Laboratory. (n.d.-c). https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets.
51. DARPA_2009. The ant lab: Analysis of network traffic. (n.d.). https://ant.isi.edu/datasets/readmes/DARPA_2009_DDoS_attack-2009.1105.README.txt.
52. Defcon, "The Shmoo Group," http://cctf.shmoo.com/, 2011.
53. UNIBS, University of Brescia Dataset (2009). http://www.ing.unibs.it/ntw/tools/traces/.
54. Lawrence Berkley National Laboratory (LBNL), ICSI, LBNL/ICSI enterprise tracing project (2005). http://www.icir.org/enterprise-tracing/.
55. Bhuyan MH, Bhattacharyya DK, Kalita JK. Towards generating real-life datasets for network intrusion detection. Int J Netw Secur. 2015;17:683–701.