Mesopotamian journal of Cybersecurity
Vol.4,No.3, **pp**. 86–135
DOI: https://doi.org/10.58496/MJCS/2024/018; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Systematic Review

A Systematic Literature Review on Cyber Attack Detection in Software-Define Networking (SDN)

Dalia Shihab Ahmed[1,*], Abbas Abdulazeez Abdulhameed [1], Methaq T. Gaata [1]

[1] *Computer Science Department, College of Science, Mustansiriyah University, Baghdad-Iraq.*

ARTICLEINFO

Article history

Received 13 Aug 2024
Accepted 11 Oct 2024
Published 11 Nov 2024

Keywords

Systematic literature review (SLR)
Software-defined networking (SDN)
Machine learning (ML)
Deep learning (DL)
Federate Learning (FL)
Entropy
Cyberattacks
Distributed denial service (DDoS)
Intrusion detection system (IDS)

ABSTRACT

The increasing complexity and sophistication of cyberattacks pose significant challenges to traditional network security tools. Software-defined networking (SDN) has emerged as a promising solution because of its centralized management and adaptability. However, cyber-attack detection in SDN settings remains a vital issue. The current literature lacks comprehensive assessment of SDN cyber-attack detection methods including preparation techniques, benefits and types of attacks analysed in datasets. This gap hinders the understanding of the strengths and weaknesses of various detection approaches. This systematic literature review aims to examine SDN cyberattack detection, identify strengths, weaknesses, and gaps in existing techniques, and suggest future research directions in this critical area. A systematic approach was used to review and analyse various SDN cyberattack detection techniques from 2017--2024. A comprehensive assessment was conducted to address these research gaps and provide a comprehensive understanding of different detection methods. The study classified attacks on SDN planes, analysed detection datasets, discussed feature selection methods, evaluated approaches such as entropy, machine learning (ML), deep learning (DL), and federated learning (FL), and assessed metrics for evaluating defense mechanisms against cyberattacks. The review emphasized the importance of developing SDN-specific datasets and using advanced feature selection algorithms. It also provides valuable insights into the state-of-the-art techniques for detecting cyber-attacks in SDN and outlines a roadmap for future research in this critical area. This study identified research gaps and emphasized the importance of further exploration in specific areas to increase cybersecurity in SDN environments.

1. INTRODUCTION

Traditional network security appliances such as firewalls, intrusion prevention systems (IPSs), intrusion detection systems (IDSs), and deep packet inspection (DPIs) offer protection against cyber threats [1][2]. However, their reliance on specialized hardware, proprietary NOS, and vendor-specific protocols makes network administration complex and requires skilled specialists for each vendor's equipment. This lack of a common framework hinders efficient control and management.

In contrast, software-defined networking (SDN) as defined in IRTF RFC 7426, provides a framework that decouples the control plane (handling network intelligence) from the data plane (forwarding traffic on the basis of control plane logic). This approach is summarized by the Open Networking Foundation (ONF) as follows: (1) Decoupling the traffic logic and control from forwarding, (2) centralized logical control, and (3) network services programmability. This increases flexibility and interoperability, leading to an open system that overcomes the limits of traditional networks. The open nature of SDN enables software development to control network resources, traffic flow, and potential inspection/modification. This empowers network administrators with greater control and flexibility, streamlining network management and enhancing security [3].

SDN has emerged as an influential tool for addressing various cyber-attacks including scanning, spoofing, sniffing, web application attacks, and malware attacks. Its versatility has led to applications in various sectors such as smart grids,

*Corresponding author. Email: dalia_shihab@uomustansiriyah.edu.iq

blockchain (BC), IoT, health, and malware remediation. The use of SDN in cybersecurity is particularly intriguing as traditional networks are complex and difficult to manage owing to the vertical integration of data and control planes. SDNs are appealing in cybersecurity because they overcome the complexities and management challenges inherent in traditional networks because of their vertically integrated data and control planes [4][5].

SDN presents a paradigm shift by decoupling data and control planes, enabling centralized network management and event correlation on the basis of open standards such as OpenFlow (OF), Protocol Oblivious Forwarding (POF), Negotiable Datapath Models (NDM), Programming Protocol Independent Packet Processor (P4), and Path Computation Element Protocol (PCEP). This open and standardized approach has facilitated the development of more robust defense mechanisms against cyberattacks. The standardization landscape in SDN is continuously evolving, with various organizations and consortiums actively involved in defining standards. This collaborative effort has led to the emergence of open-source implementations, making SDN adoption more accessible and cost-effective [6].

SDNs have revolutionized the field of network management and security, offering incomparable flexibility, scalability, and centralized control. While traditional network security tools are effective against cyberattacks, they are constrained by difficulty and management issues. The rise of SDN as an appealing alternative is attributed to its ability to isolate the control plane from the data plane, allowing centralized administration and improved adaptability. Despite the numerous benefits that SDN offers, it also presents new vulnerabilities and attack vectors that require robust detection mechanisms. The increasing sophistication of cyber-attacks requires the development of progressive techniques for threat detection and mitigation in SDN environments. Table 1 lists the acronyms used throughout the article.

The following subsections discuss the motivation behind this SLR in Section 1.1, related works to this SLR in Section 1.2, and the contributions and organizations of this SLR in Section 1.3.

TABLE I. LIST OF ABBERVIATIONS

Acronym	Full Name	Acronym	Full Name	Acronym	Full Name
ACC	Accuracy	GCN	Graph Convolutional Neural Network	ONF	Open Networking Foundation
AE	Autoencoder	GE	Generalized Entropy	PCA	Principal Component Analysis
APTs	Advanced Persistent Threats	GNB	Gaussian Naïve Bayesian	PCEP	Path Computation Element Protocol
ASVM	Advanced Support Vector Machine	GRU	Gated Recurrent Unit	POF	Protocol Oblivious Forwarding
AUC	Area Under Curve	ID	Information Distance	POP3	Post Office Protocol Version 3
BGRU	Bidirectional Gated Recurrent Unit	IDPS	Intrusion Detection and Prevention System	PREC	Precision
CART	Classification and Regression Trees	IDS	Intrusion Detection Systems	QDA	Quadratic Discriminant Analysis
CNN	Convolutional Neural Network	IG	Information Gain	QoS	Quality of Service
DBN	Deep Belief Networks	IGR	Information Gain Ratio	R2L	Remote to Local
DDoS	Distributed Denial-of-Service	IPS	Intrusion Prevention Systems	RF	Random Forests
DDQN	Double Deep Q-Network	K-NN	K-Nearest Neighbor	RNN	Recurrent Neural Network
DL	Deep Learning	KPCA	Kernel Principal Component Analysis	ROC	Receiver Operating Curve
DNN	Deep Neural Network	LDS	Link Discovery Service	ROS	Random Oversampling
DNS	Domain Name System	LLDP	Link Layer Discovery Protocol	SAE	Stack Autoencoder
DoS	Denial-of-Service	LOA	Lion Optimization Algorithm	SBI	Southbound Interface
DPI	Deep Packet Inspection	LR	Logistic Regression	SDN	Software-Defined Networking
DR	Detection Rates	LSTM	Long Short-Term Memory	SEN	True Negative Rate
DT	Decision Trees	MCC	Matthews Correlation Coefficient	SIMFO	Self-Improved Moth Flame Optimization
FAR	False Alarm Rate	MITM	Man in the Middle	SMOTE	Synthetic Minority Oversampling
FL	Federated Learning	ML	Machine Learning	SOM	Self-Organizing Maps
FM	Factorization Machine	MLP	MultiLayer Perceptron	SPEC	Specificity
FN	False Negative	NB	Naïve Bayes	SVC	Support Vector Classifier
FNR	False Negative Rate	NBI	Northbound Interface	SVM	Support Vector Machine
FP	False Positive	NDM	Negotiable Datapath Models	TN	True Negative
FPR	False Positive Rate	NIDS	Network Intrusion Detection System	TNR	True Negative Rate
FTP	File Transfer Protocol	NN	Neural Network	TP	True Positive
GA	Genetic Algorithm	Non-IID	Non-Independently and Identically Distributed	TPR	True Positive Rate
GAN	Generative Adversarial Networks	NOS	Network Operating Systems	U2R	User to Root

1.1 Motivation Behind This SLR

The motivation behind conducting an SLR on entropy, ML, DL, and FL approaches for detecting cyber-attacks in SDN networks is to offer a comprehensive overview of current research in this area and assess the strengths and weaknesses of these methods. Cyberattacks pose significant threats to SDN networks. However, traditional security measures may struggle to detect these attacks, as attackers employ new approaches to overwhelm SDN systems with diverse traffic patterns, leading to degradation of the SDN controller and denial of access to legitimate users. ML and DL technologies are proposed as potential solutions for categorizing these attacks. These techniques analyse network traffic flow patterns and identify abnormal traffic behaviors indicative of cyberattacks. However, there is a lack of consensus on the most effective entropy, ML, DL, and FL approaches for detecting cyberattacks. The SLR addresses these knowledge gaps by systematically reviewing and consolidating approaches. Through the SLR, the authors aim to present a comprehensive and transparent overview of relevant research, identify shortcomings in current methodologies, and assist the academic community in identifying effective techniques for detecting cyberattacks in SDN networks and developing more robust detection mechanisms against such threats

1.2 Previous Related Survey

Previous reviews have addressed DDoS attacks on SDN networks and related security measures, with specific surveys focusing on SDN DDoS attacks. However, to the best of our knowledge, the majority of studies focus mainly on DDoS attacks while neglecting other potential forms of cyberattacks. Furthermore, a comprehensive evaluation through systematic analysis and synthesis of existing studies on entropy, ML, DL, and FL for detecting and mitigating cyber-attacks in SDN environments is lacking. We performed a qualitative comparison with existing surveys to showcase the distinctiveness of our work for detecting cyberattacks.

TABLE II. COMPARISON OF SYATEMATIC LITERATURE REVIEW WITH EXISTING SURVEYS ON CYBER ATTACK DETECTION IN SDN.

References	Approaches				Public Datasets	Feature Selection	Type of Attacks	Strength	Weakness	Research Gaps
	Entropy	ML	DL	FL						
[7]	×	✓	✓	×	✓	×	×	×	×	×
[8]	×	×	✓	×	×	×	✓	×	×	×
[9]	×	×	✓	×	×	×	×	✓	✓	×
[10]	×	✓	✓	×	✓	×	×	×	✓	×
[11]	×	✓	×	×	×	×	×	×	×	×
[12]	×	✓	×	×	✓	×	×	×	×	×
Our work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

(✓): Addressed, (×): Unaddressed.

To highlight related current work in this context [7], we delve into the classification of research works focusing on ML and DL techniques accurately applied to intrusion detection systems and discuss the methodologies employed and the datasets used in these studies. Similarly, [8] focused on articles on DL, intrusion, and attacks from major databases. This highlights the use of DL for intrusion detection and classification. This research aims to present evaluation criteria, taxonomic literature, and challenges in this field. Furthermore, the research outlined in [9] presents a taxonomy of DL intrusion detection models, reviews relevant research, and evaluates four key DL models on legacy datasets. Additionally, it discusses research challenges and suggests future directions for applying ML methods in intrusion detection. Moreover, the study referenced in [10] provides a comprehensive overview of AI-based IDS design and emphasizes the importance of using ML and DL techniques for efficient network security. Additionally, the review discussed in [11] compares the various ML techniques that are used in the detection of DDoS attacks in the SDN environment. Moreover, the article mentioned in [12] surveys the DoS/DDoS detection techniques in SDN based on ML and surveys the tools and datasets considered by the reviewed studies.

Overall, as shown in Table 2, our SLR study is qualitatively distinctive from other studies in SDN cyber-attack detection approaches. It provides a holistic overview of state-of-the-art approaches by reviewing and analysing the entropy, ML, DL, and FL approaches. Additionally, this SLR identifies publicly or privately available datasets; highlights the feature selection methods applied, strengths, and weaknesses; and highlights the research gaps identified.

1.3 Contributions and Organization of SLR

Research on SDN technology is still in its primary stages of development, making SDN networks susceptible to cyberattacks. Various security methodologies utilizing entropy, ML, DL, and FL have been developed to counter these threats. However, for academicians and the security community to understand the research landscape in this domain and track its evolution, further efforts are needed to systematically review, synthesize, and conduct thorough investigations into existing approaches.

Therefore, the primary contribution of this review is the systematic analysis of current techniques for detecting cyber-attacks in SDN, aiming to identify their strengths, weaknesses, and possible gaps. Addressing the existing gap in the literature regarding comprehensive evaluations of preparation techniques, advantages, and attack types in analysed datasets, this review provides valuable perspectives on the current state-of-the-art cyberattack detection techniques. The objectives of this review include the following:

1. Comprehensive Background of SDN Architecture: We provide a strong foundation of SDN architecture, highlighting its forwarding process and key components.
2. Classification of Attacks against SDN Planes: Our research categorizes attacks against SDN control planes on the basis of attack surfaces, including the Northbound Interface (NBI), Southbound Interface (SBI), SDN Controller, and multicontroller environments. This structured approach facilitates the understanding of diverse attack vectors in SDN environments.
3. Detailed Analysis of Datasets: We analyse various datasets used for cyber-attack detection and provide a comprehensive explanation of their characteristics. We specify the lack of detailed public dataset surveys and accentuate the need for generating SDN-specific datasets.
4. Feature Selection Techniques: Our survey offers an overview of feature selection techniques utilized by researchers to select optimal feature sets from generated or available datasets. We highlight the importance of creating SDN-specific datasets and applying feature selection algorithms to improve real-world network detection capabilities.
5. Analysis of Detection Approaches: We conduct a comprehensive analysis of literature on entropy, ML, DL, and FL approaches to cyberattack detection and mitigation, specifically in SDN networks.
6. Evaluation Metrics Analysis: We analyse evaluation metrics employed to evaluate the effectiveness of defense mechanisms against cyberattacks.
7. Roadmap for Future Research: We provide a roadmap for researchers who are working on SDN cyber-attack detection approaches, highlighting strengths, weaknesses, gaps, and future research directions.

The importance and implications of this review lie in its ability to provide readers with a comprehensive overview of cyber-attack detection in SDN from 2017--2024. By addressing research gaps, assessing the strengths and weaknesses of current approaches, and suggesting future research directions, this review serves as a valuable resource for researchers, practitioners, and policymakers in the field of network security.

This paper is organized as follows: Section 1: Introduction. Section 2: Research methodology for the SLR presented in this study. Section 3: Background of the architecture of the SDN model and OpenFlow forwarding process. Section 4: Attack taxonomy which discussing various attacks targeting different entry points in the SDN architecture. Section 5: Comprehensive description of publicly available datasets that are used for cyber-attack detection. Section 6: Feature selection methods for producing optimal feature datasets to enhance cyber-attack detection. Section 7: Literature survey on the use of entropy, ML, DL, and FL algorithms for cyber-attack detection in SDN environments. Section 8: Analysis of dataset sources. Section 9: Analysis of metrics for performance evaluation. Section 10: Discussion of research strengths and weaknesses of these articles. Section 11: Discussion of research gaps in SDN security, highlighting areas for further research and development of new security methods. Last Section 12: Conclusions.

2. RESEARCH METHOD

This section outlines the methodology for conducting a systematic literature review (SLR) in this study. The main aim of SLR is to provide a comprehensive and structured overview of existing research within a specific field while concurrently identifying research gaps and potential directions for future research. In this section, we present details of the methodology applied, including formulation of research questions (RQs), construction of search strings, selection of data sources, and establishment of eligibility criteria, encompassing both inclusion and exclusion criteria. A summary of the research protocol is shown in Figure 1.

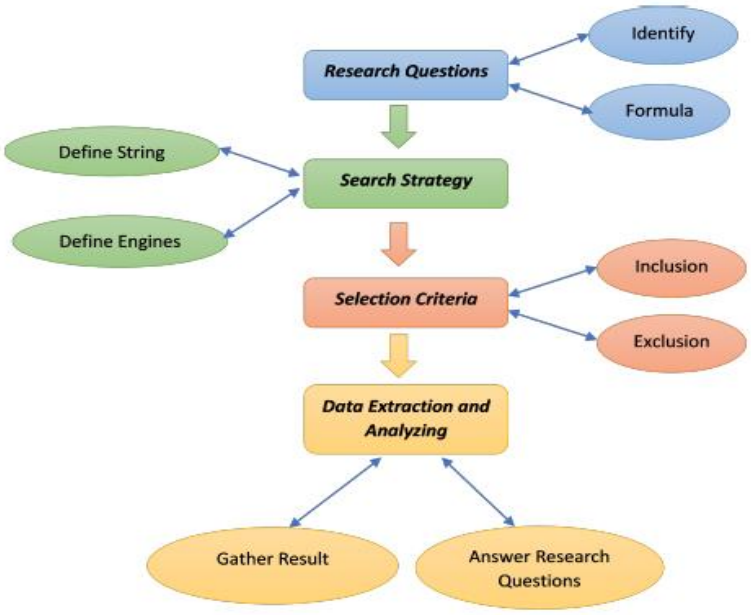

Fig. 1. Survey Protocol Overview.

2.1 Research Questions (RQ)

The primary objective of conducting a systematic review is to address specific research questions by analysing data extracted from previous studies. In this study, several research questions were identified and addressed, including the following:

RQ1: What are the existing techniques for detecting and mitigating cyber-attacks in SDN?

RQ2: What are the main types of attacks recognized by the researchers and tools used to detect cyber-attacks on SDN planes?

RQ3: What are the strengths and weaknesses of current methodologies for detecting cyberattacks?

RQ4: What metrics are commonly used to evaluate the performance of cyber-attack detection techniques?

RQ5: What datasets are used for evaluating and validating present approaches, and are there publicly available realistic datasets for cyber-attacks on SDN networks?

RQ6: What are the research gaps and future directions in the field of cyber-attack detection techniques in SDN?

2.2 Search String

This study used a comprehensive search methodology to gather related literature via Boolean OR/AND operators. This approach simplified the connection of keywords, terms, synonyms, and abbreviations, ensuring the systematic exploration of research topics. The search plan was executed in two distinct phases: automatic search and subsequent manual search. During the first phase, the automatic search was performed using predefined keywords derived from the research questions and the structure of the SLR. For example, keywords such as (“Software Defined Networking” AND “Cyber-attacks”) OR (“SDN” AND “DDoS”) AND (“Intrusion Detection System” OR “IDS” AND “Network Security”). These keywords were carefully selected to include the most relevant and related studies. The search string consisting of predefined keywords, was applied to digital database sources. Additionally, the search string was saved on all database sources to obtain notifications about newly published articles matching its search criteria. Research papers obtained from several databases were then screened on the basis of predefined research questions and inclusion and exclusion criteria for SLR were established.

In the second phase, manual screening process was undertaken to review the references cited in the primary studies. This involved employing backwards and forward search techniques to track the citations of primary studies. By examining references and applying inclusion and exclusion criteria, studies obtained in this stage were added to Mendeley. Mendeley

was utilized as a tool to manage collected studies efficiently, facilitate elimination of duplicate studies, and establish the last set of selected studies for analysis.

2.3 Data Sources

For this SLR, an accurate search strategy was employed to identify pertinent studies published from 2017--2024. The search procedure encompassed querying multiple academic research databases as outlined in Table 3.

TABLE 3. SEARCH DATABASE SOURCE.

Sn	Sources	URL
1.	IEEE Xplore	https://ieeexplore.ieee.org/Xplore/home.jsp
2.	SpringerLink	http://www.springer.com.
3.	ScienceDirect	http://www.sciencedirect.com
4.	ACM Digital Library	http://dl.acm.org.
5.	Wiley Online Library	http://onlinelibrary.com
6.	Hindawi	http://hindawi.com
7.	MDPI	http://mdpi.com
8.	Scopus-Elsevier	https://www.scopus.com
9.	Taylor & Francis Online	https://www.tandfonline.com

2.4 Inclusion/exclusion criteria

The inclusion and exclusion criteria terms are defined in the SLR to ensure that studies relevant to the research questions are selected. Table 4 outlines these criteria which focus on entropy, ML, DL, and FL approaches for detecting and mitigating cyberattacks in SDN networks. Studies that did not meet inclusion criteria in the SLR were excluded.

TABLE IV. SUMMARY OF THE INCLUSION AND EXCLUSION CRITERIA.

Sn	Inclusion	Exclusion
1.	Research that centers on cyber-attacks, as well as detection and mitigation mechanisms in SDN.	The study did not specifically address cyber-attacks, detection, and mitigation mechanisms in SDN.
2.	Content from conferences, journals, and book chapters is written in English.	Written in other languages.
3.	Research published from 2017 to 2024.	Research not published from 2017 to 2024.
4.	Included in database source.	The full version is not accessible.
5.	Related to the approach for detecting and mitigating cyber-attacks on SDN networks.	Related to cyber-attacks on SDN within IoT, cloud computing, 5G, mobile, wireless, and ad hoc networks, as well as cyber-attacks on traditional networks.
6.	Related to research questions.	Not related to research questions.

2.5 Research study selection procedure

The selection procedure for research studies is a vital step in identifying relevant literature that aligns with the research questions of this SLR. By using automatic search techniques, a total of 300 studies were initially retrieved. In Section 2.2, predefined keywords employed to gather research studies from various database sources are described in detail. A series of stages were implemented to ensure that only studies relevant to the topic of this SLR were included, as shown in Figure 2.

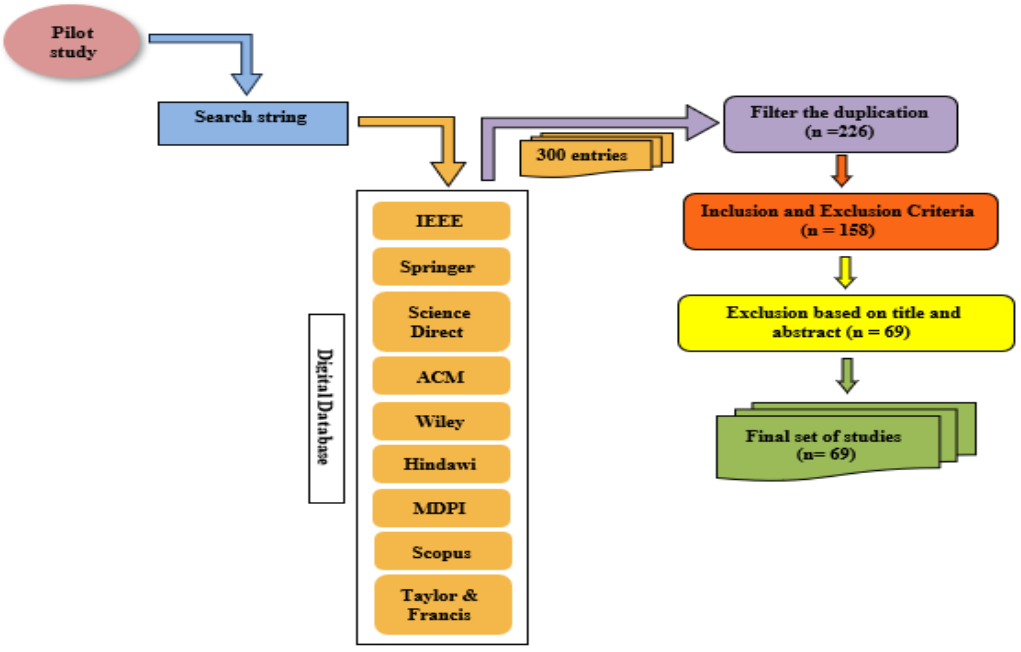

Fig. 2. Overall Research Methodology Protocol.

In the first stage, the Mendeley reference manager was used to exclude duplicate research studies, resulting in a total of 226 studies. The second stage involved the application of inclusion and exclusion criteria, as presented in Table 4, to extract related studies and exclude those that were deemed irrelevant. This stage yielded a total of 158 studies, and a significant number of papers were eliminated because of their lack of relevance to our research topic. Examples of such papers included those addressing SDN security issues in the context of cloud computing, the IoT, and blockchain. However, our research focuses specifically on reviewing security issues and challenges related to SDN planes, namely, the application plane, control plane, and data plane. Finally, 69 studies were selected on the basis of the title and abstract. Additionally, from these studies, a detailed taxonomy of cyber-attack solutions was developed, as illustrated in Figure 3.

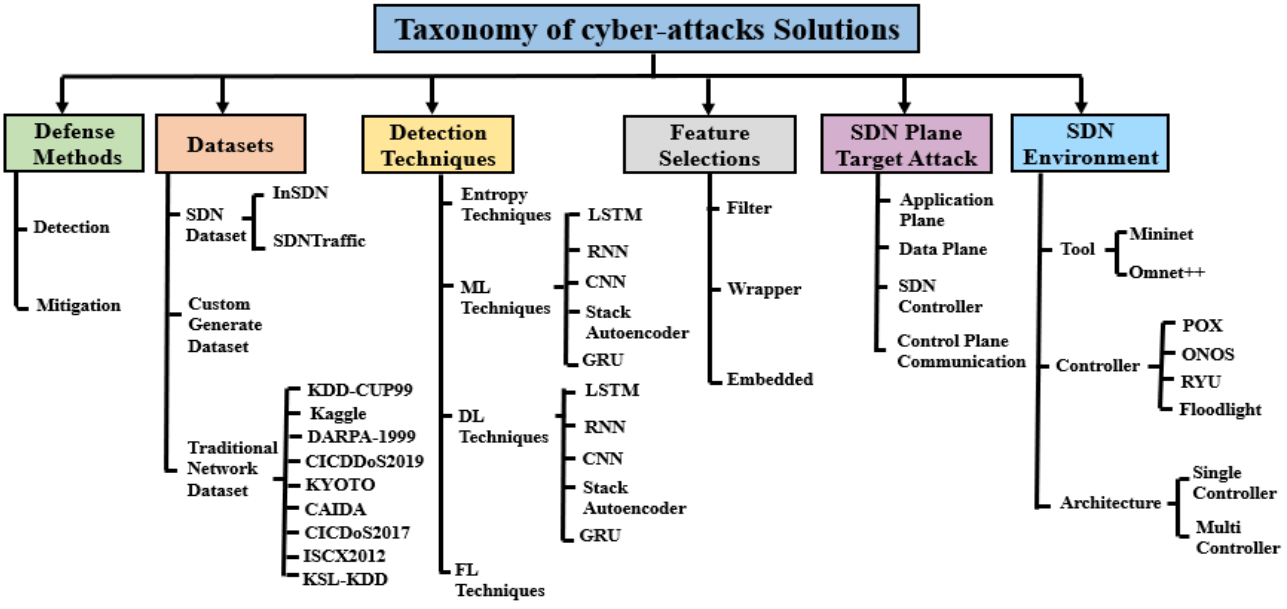

Fig. 3. Taxonomy of Cyber-attacks Defense Solutions.

2.6 Limitations of the Research Methodology

While this systematic literature review provides a comprehensive analysis of cyber-attack detection in SDN, it is important to acknowledge several limitations. The methodology relies on specific databases, which may exclude relevant studies not indexed in these sources. Additionally, the review focused primarily on cyber-attack detection techniques in SDN from 2017--2024, and there is an opportunity to expand the scope to include earlier studies or more recent developments in the field. The review's focus on entropy, ML, DL, and FL approaches restricts its consideration of other potential methodologies, such as those based on blockchain technology. While the review focused on security issues and challenges on SDN planes, there is an opportunity to broaden the scope to include studies related to cyberattacks on SDN within IoT, cloud computing, 5G, mobile, wireless, and ad hoc networks, as well as traditional networks. Not all research methods and questions have been fully addressed in this review, but efforts have been made to address some open research issues, offer insights into future directions, and identify gaps in the literature. Despite these limitations, this systematic literature review provides valuable insights into the current advancements in cyber-attack detection in SDN and offers a roadmap for future research in this vital field.

2.7 Data Extraction and Data Synthesis

The data extraction and synthesis stage involved thorough examination of the 69 selected studies with relevant data being abstracted and recorded via Microsoft Excel spreadsheets and the Mendeley reference manager. This process entailed creating a data extraction form to systematically capture and report all pertinent information derived from primary studies. Consequently this SLR considers various columns in the data extraction form (as depicted in Table 5).

TABLE V. EXTRACTION OF DATA ITEMS OF PRIMARY STUDIES AND DESCRIPTIONS.

No.	Data Extracted	Description
1.	Study ID.	Specific identity numbers for each study to facilitate tracking and analysis.
2.	Publication Year.	The year in which the study was published.
3.	Method used.	Lists the various related methods that were employed in the article (i.e., entropy, ML, DL, and FL).
4.	Types of Attacks.	The types of cyber-attacks considered in the study.
5.	Dataset type.	Lists the various datasets used by the study (i.e., benchmark datasets and realistic or unrealistic datasets).
6.	Target Plane.	The SDN plane (application, control, or data plane) targeted by the study.
7.	Number of Controllers.	The number of SDN controllers used in the study.
8.	Controller Type.	The type of SDN controller used in the study.
9.	Number of Features.	The number of features extracted from network traffic for analysis.
10.	Feature Selection Used.	The feature selection methods are used to select relevant features for analysis.
11.	Features Utilized.	The specific features used for analysis.
12.	Study Strengths	Describes the model’s positive attributes.
13.	Study Weaknesses	Lists the model’s shortcomings.
14.	Evaluation Metrics.	The metrics used to evaluate the performance of the proposed solutions.

3. SOFTWARE-DEFINED NETWORKING (SDN)

In this section, the SDN architecture model is comprehensively examined, providing detailed explanations of the application, control, and data planes. Additionally, OpenFlow forwarding process is described, shedding light on its functionality in the SDN framework.

3.1 SDN Architecture Model

The SDN (software-defined networking) architecture has been seen as solution to address the difficulty of traditional networks. It represents an important technological advancement in the field of networking. The essential concept of SDN architecture is to separate the logical control of network devices from the data forwarding plane. This architecture consists of three interconnected layers, "application plane," "control plane," "data plane," and two interfaces, "southbound API," and "northbound API". Segmentation of the SDN architecture into these layers and interfaces simplifies network management and enhances its scalability [4][6]. Figure 4 provides visual representation of these layers and interfaces. Each layer serves a distinct purpose within the architecture. Certain features, such as the NOS, application networks, southbound API, and northbound API, are essential components of any SDN implementation. However, other features, such as language-based virtualization or hypervisors, are optional. The following subsections provide an overview of each layer and interface API, starting at the top and moving downwards. This will help to further understand the functionality and significance of each component within the SDN architecture [1].

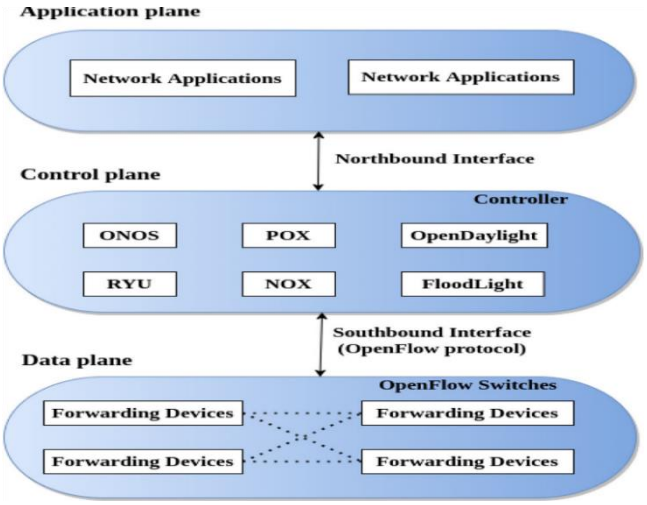

Fig. 4. SDN architecture model in planes.

3.1.1 Application Plane

The upper layer of the SDN architecture model is known as the "application plane". This layer is responsible for managing and configuring network devices in the data plane. It interacts with the data plane via the northbound API interface and obtains network information from the control layer. This layer consists of various software programs that can be developed by developers. Within the application layer there are six distinct types of applications: (1) traffic engineering apps (2) network analysis and monitoring apps (3) failover apps (4) network maintenance apps (5) network security apps and (6) diverse apps, such as firewall, prevention, and detection systems. These applications define the forwarding device's functionality. Unlike traditional networks with static devices limited to single functions, SDN devices have the flexibility to perform multiple functions. This dynamic nature of SDN devices allows for greater functionality and adaptability within the network [6].

3.1.2 Northbound API

The SDN architecture incorporates two essential API interfaces, SBIs and NBIs, which serve as connectors between diverse layers. The Northbound API acts as a communication interface between network applications operating in the application that operates in the application and control layers of the SDN architecture. It facilitates the movement of information and commands between these layers. In contrast, SBI primarily utilizes the OpenFlow protocol, which is an open standard protocol for communications between control and data layers. Additionally, advancements in technology have led to the introduction of various Northbound interfaces (APIs) by different organizations and enterprises. For example, the RESTful API has gained widespread support and is widely implemented in various SDN controller platforms, such as NOX. However, diverse manufacturers of SDN controllers may present and create their proprietary northbound APIs. Examples of such controllers include floodlight and OpenDaylight [13].

3.1.3 Control Plane

The control plane, an integral component in SDN architecture, works between application and data planes and is managed by the SDN controller or NOS. The controller manages network operations, making decisions on flow forwarding and packet dropping through programming. It serves two key functions: controlling the network by enforcing rules from the application level to the infrastructure and monitoring the global and local network status, as shown in Figure 5. The controller maintains policies, synchronizes network status, and provides a full network view through downwards and upwards flows.

Many interfaces, such as SBIs and NBIs, simplify communication for controllers and different network planes. Additional east–west interfaces are utilized in scenarios involving multiple controllers, enabling data exchange, connectivity checks, and coordination between controllers and forwarding devices. Scalability in the control layer is vital for SDN performance, with distributed controllers recommended to increase processing capacity and reduce workload on individual controllers [6].

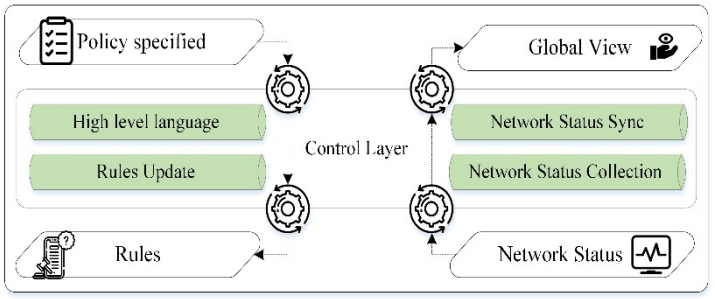

Fig. 5. Logical Design of the SDN Controller [51].

SDN controller platforms can adopt centralized or distributed architectures. In a centralized setup, a single controller manages all network devices and is suitable for networks of varying scales but is susceptible to single-point failures and cyberattacks. Distributed architectures employ multiple controllers to oversee specific network segments, mitigating the risks associated with single controller failures. Notable SDN platforms such as POX, NOX, Floodlight, OpenDaylight, and ONOS distributed SDN.

3.1.4 Southbound API

The SDN architecture presents the physical separation of the forwarding plane and the control plane, achieved through the use of southbound API interfaces. These interfaces create connections between the control plane, which is located on an independent controller, and the forwarding plane, which is located on network devices. The SBIs play a vital role in establishing a connection between the control plane and the data plane. It is essential to ensure that this link remains accessible and secure to ensure the correct functioning of the forwarding functions. Its main function is to enable the control plane to send management and monitoring messages to the data plane. In response, the data plane collects these messages and transmits them to the controller, providing updates on the network's current state [13].

Among various SBIs, the OpenFlow protocol is widely emphasized and considers the standard interface defined by the ONF. The OpenFlow protocol incorporates a secure communication mechanism to ensure protected communication over SBI. However, it is important to note that the use of the OpenFlow protocol is not mandatory, as suggested by ONF. Other protocols, such as the OvSDB (open vSwitch database), OpenState, and OpFlex, are also available as alternatives for implementing SBI.

3.1.5 Data Plane

In SDN architecture, the data plane, or infrastructure layer, includes network devices such as switches and routers that form the foundation of the network. These devices are responsible for basic forwarding functions and the absence of decision-making capabilities, with control intelligence centralized in the SDN controller. Built on open and standard interfaces, especially the OpenFlow protocol, these network devices ensure standardized configurations, consistent communication, and compatibility between devices in both the data and control planes. This contrasts with traditional networks that fight device heterogeneity, proprietary standards, and decentralized control logic [6].

In the SDN and OpenFlow architectures, Figure 6 depicts controllers and forwarding devices as key components. The controller functions as the network's intelligence center, running on commodity hardware, while forwarding devices manage the packet forwarding process, either by hardware or software. OpenFlow devices contain flow tables comprising matching rules, actions, and counters. Matching rules specify header fields such as TCP, UDP, IP, and Ethernet; actions define traffic operations, such as forwarding or dropping packets; and counters track packet statistics for each flow [4].

The OpenFlow switch comprises a flow table and establishes a secure channel for communication with the controller. Its pipeline contains sequential flow tables that dictate packet processing. The pipeline, which uses network status, generates new forwarding rules sent back to switch, which autonomously processes and forwards subsequent packets. The straightforward nature and conceptual clarity of OpenFlow contribute to its extensive adoption in SDN data plane devices [13].

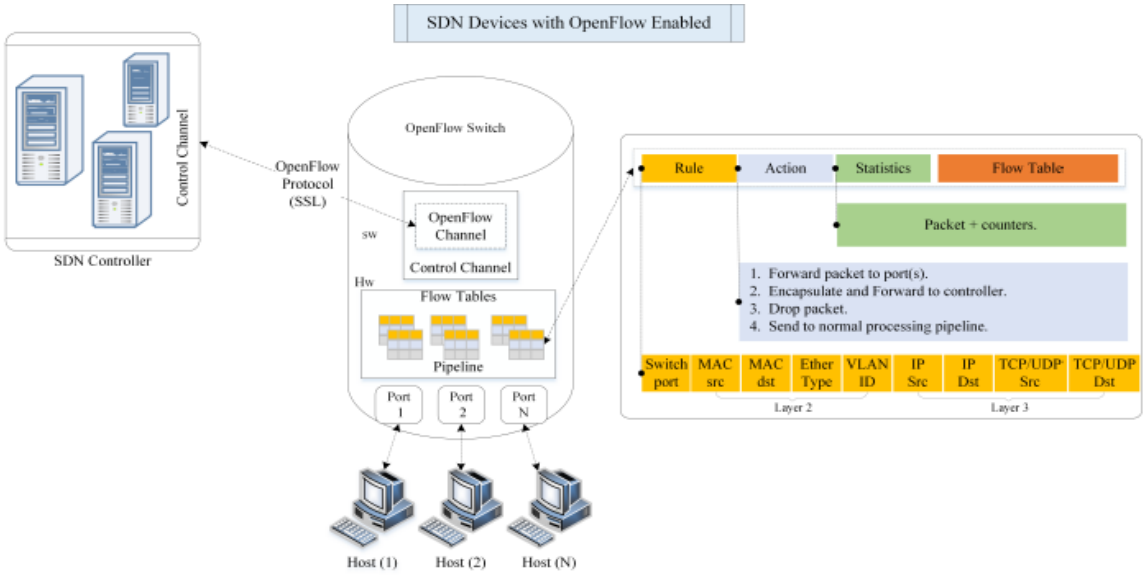

Fig. 6. Core Elements of SDN and OpenFlow Architecture.

3.2 OpenFlow Forwarding Process

The SDN controller plays a crucial role in SDN architecture by handling forwarding processing rules for network flows. It is achieved by installing flow entries onto OpenFlow-enabled switches. The OpenFlow specification allows switches with flexibility to operate in either proactive or reactive mode. In proactive mode, backup flow rules are preinstalled into switches' flow tables before any network flow entries are received. This approach offers advantages such as minimal setup time and a reduced frequency of communication with the SDN controller. However, it is not cost efficient to preinstall all backup rules in advance because the memory of SDN switches is limited and expensive. As these backup rules are only used in the case of failures, preinstalling all of them would unnecessarily consume flow table resources. In contrast, SDN's reactive mode involves a more event-driven and interactive approach. When packets arrive at the switch, flow entry rules are dynamically installed in real time. However, this process can be time-consuming and introduce delays in packet forwarding. This is because it needs to request, search for, or compute the required rules in real time on the basis of the incoming packets. Overall, the choice between proactive and reactive modes involves trade-offs between setup time, communication frequency, resource utilization, and packet delay [6][14].

4. CYBERATTACK AGAINST THE SDN NETWORK ENVIRONMENT

As shown in Figure 7, the taxonomy categorizes attacks on the basis of their target within the SDN architecture. These attacks are organized on the basis of their entry points within the SDN control plane, namely, the North Bound Interface, Controller, South Bound Interface, and Link connecting two controllers.

4.1 Controller-based Attacks

Controller-based attacks are described below [19][20][17]:

4.1.1 Packet in Flooding (DoS/DDoS)

SDNs face security challenges from attacks targeting SBIs and controllers, exploiting cooperative controllers and vulnerable switches. Centralized control presents a single point of failure, mitigated by the use of multiple controllers and careful rule implementation to avoid DoS attacks. In OpenFlow, poor rule design can overload controllers with inquiries, affecting network switches. Reactive networks are more vulnerable to DoS threats than proactive networks are, requiring caution in flow adjustments to prevent flooding controllers with Flow Mod notifications. While OpenFlow 1.3 suggests monitoring packets to the controller, specific guidelines on rate-limiting signals and rule entries are lacking.

4.1.2 Saturation of the Controller (DoS/DDoS)

The saturation of controller attacks targets SDN controllers in SDN architecture, aiming to overwhelm their resources and disrupt network traffic management. By flooding the controller via a Northbound API with high volumes of traffic,

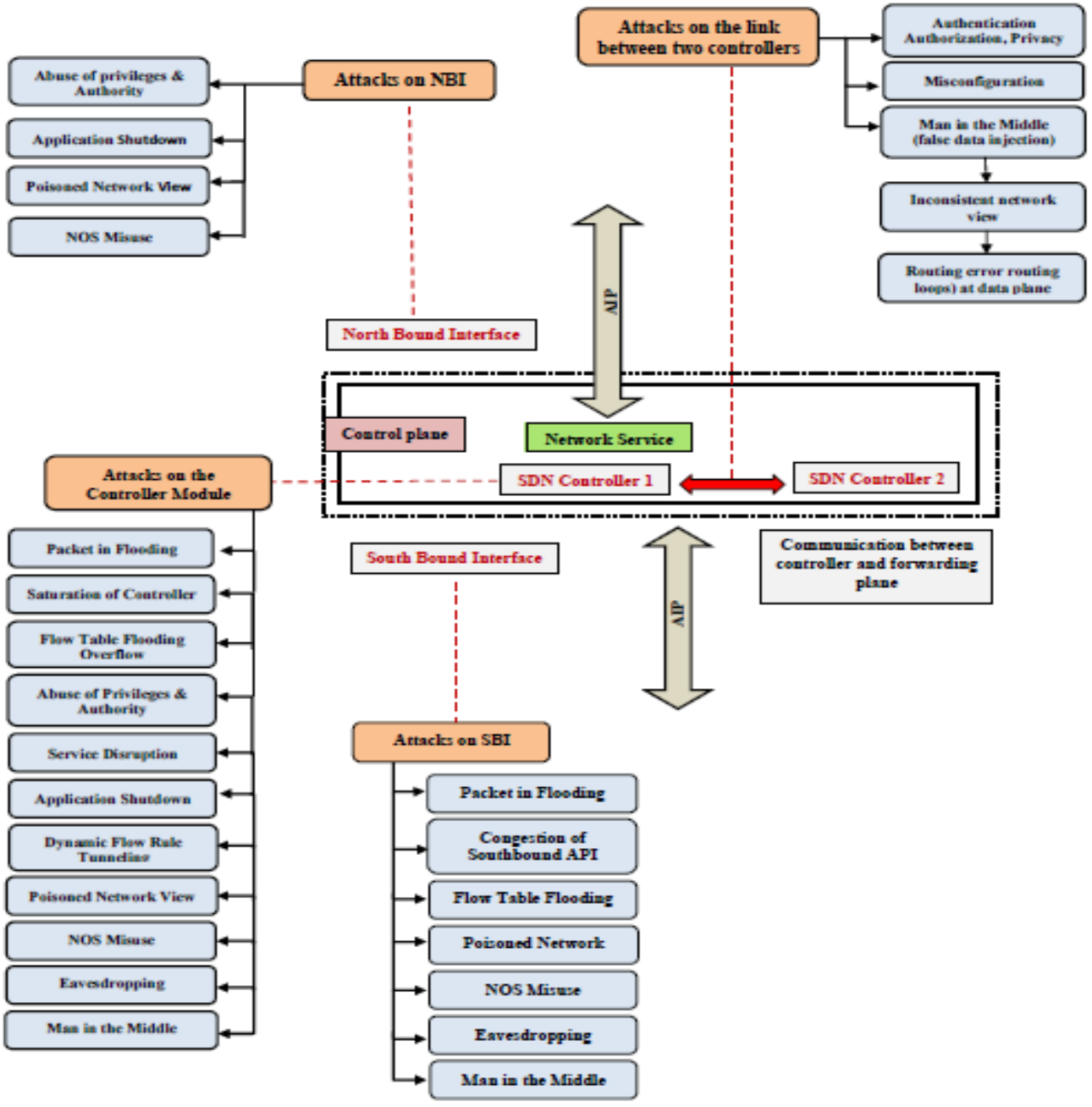

Fig. 7. Attack Taxonomy.

attackers can hinder its performance, potentially causing slowdowns or shutdowns. Mitigation strategies include implementing rate limiting, monitoring traffic patterns for anomalies, utilizing load balancing, updating controller software regularly, and imposing access control policies. A well-defined reply plan involving isolating the controller's affected controller and redirecting traffic to backup or alternative components is vital to minimize the attack's impact and maintain network availability.

4.1.3 Flow Table Flooding/Overflow (DoS/DDoS)

The SDN faces security threats when it targets the SBI and controller by exploiting susceptible controllers and switches. Flow table overflow attacks can reduce space in flow tables, leading to potential DoS incidents and compromising network

functionality. These attacks, the scenario depicted in Figure 8, pose important security risks in SDN, potentially triggered by adversaries with access to hosts within the network or compromised hosts.

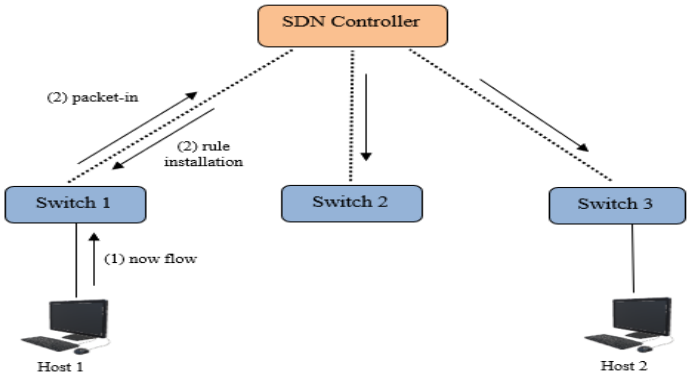

Fig. 8. Flow Table Overflow Attack Scenario.

4.1.4 Abuse of Privileges & Authority

Malicious SDN applications target the NBI and SDN controllers, which poses security threats. These applications, such as Sh14, RH15a, and RH, exploit serious system operations to crash controllers, operate data structures, establish remote channels to command and control (C&C) servers, and perform shell commands. Some rogue applications have downloaded and performed files with root access, increasing security risks in the network.

4.1.5 Service Disruption

These attacks target SDN controllers and exploit several domains, using tactics such as impersonating API communications to operate network elements and create new flows. If successful, attackers can run traffic routing within SDN, possibly bypassing security measures and regulations, or launch DoS attacks to disrupt the controller. Controller vulnerabilities may arise from the use of Linux-based operating systems with known flaws, default passwords, and a lack of security settings in production environments. Attackers can make their controllers deceive network components, gain control over flow tables, and manipulate network flows, posing important security risks.

4.1.6 Application Shutdown

Attacks targeting NBI often arise from compromised northbound protocols, which impact SDN controllers. SDN controllers use various northbound APIs, such as Python, Java, C, REST, XML, and JSON, with insecure APIs serving as possible entry points for attackers. Exploiting vulnerabilities in the northbound API can grant unauthorized access to the SDN network, allowing attackers to create and enforce their policies, potentially gaining control over the whole SDN ecosystem. For example, the default password in the REST API could be simply discovered, enabling attackers to access and alter the SDN ecosystem configuration if not altered by SDN implementation.

4.1.7 Dynamic Flow Rule Tunneling

Malicious SDN applications target SDN controllers, with researchers uncovering attack methods such as bypassing flow controls in OpenFlow switches. Studies such as Po12 and Po15 demonstrate how attackers can avoid drop rules and access network hosts through malicious flow rules. This attack uses dynamic flow rule tunneling, exploiting commands in the OpenFlow protocol to perform network access.

4.1.8 Poisoned Network View

Attacks targeting the SDN controller, northbound, and SBIs exploit LDS in the SDN control plane, which is vital for network application efficiency. Adversaries compromise the controller's topology view by making artificial links across infected devices. Depending on the LLDP, packets pose a security susceptibility within the LD. Researchers highlight that SDN controllers lack adequate protection for integrity and source verification of LLDP packets, enabling hackers to manipulate link data by injecting fake LLDP packets or replaying legitimate packets between switches.

4.1.9 NOS Misuse

Malicious SDN applications instigate attacks that impact SBI, NBI, and SDN controllers. Research has explored a varied range of unauthorized activities these applications can perform. Notably, the release of SDN rootkit-like RH15b for OpenDaylight introduces susceptibilities, enabling unauthorized control over network programming and analysis components. Attackers can manipulate flow rules, including adding malicious rules and removing valid rules discreetly, which is facilitated by OpenFlow for remote communication between the attacker and rootkit component within the NOS, thereby demonstrating the difficulty of communication in SDN data and control planes.

4.1.10 Eavesdropping

These attacks target unencrypted control channels in the SBIs and SDN controller. Eavesdropping in SDN can occur within a data plane or communication lines connecting controllers to forwarding devices. Common locations for eavesdropping attacks include switches and forwarding links, allowing malicious actors to monitor transmitted data for potential exploitation. These attacks are notably common in TCP networks.

4.1.11 Man in the Middle

These targeted attacks attempt to exploit vulnerabilities found in SBIs, SDN controllers, and connections between two controllers. These vulnerabilities stem from an absence of encryption in control channels, compromised SBIs, and insecure data links.

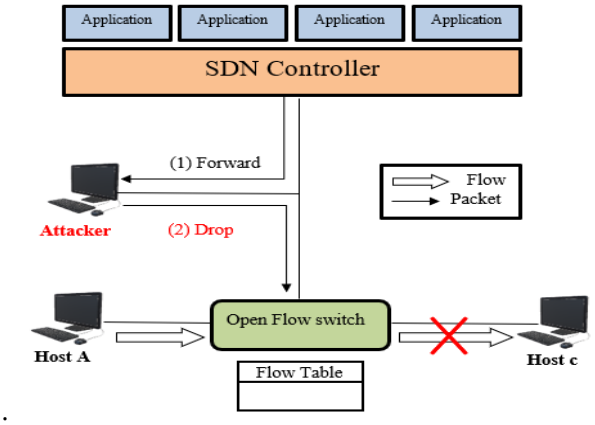

Fig. 9. MITM attack scenario.

Figure 9 illustrates the MITM attack model, where ongoing OpenFlow messages transmitted through the control channel are actively operated. This interference disrupts interactions between the control and data planes. In this attack scenario, the following actions occur when a flow rule instructs a switch to forward a group of flows from host A to host C: (1) the controller delivers the flow rule to the switch, and (2) the attacker actively modifies the rule's action variable of flow to "drop." As a result, (3) an alter flow rule is installed on the switch, causing the flow from host A to host C to drop.

4.2 North Bound Interface-based Attacks

The North Bound Interface-based attacks are described below[18][19][1]:

4.2.1 Abuse of Privileges & Authority

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.4 for detailed information on this specific attack).

4.2.2 Application Shutdown

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.6 for detailed information on this specific attack).

4.2.3 Poisoned Network View

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.8 for detailed information on this specific attack).

4.2.4 NOS Misuse

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.9 for detailed information on this specific attack).

4.3 Southbound Interface-based Attacks

Southbound interface-based attacks are described below[5][20]:

4.3.1 Packet during flooding

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.1 for detailed information on this specific attack).

4.3.2 Congestion of the Southbound API

This attack is a form of DoS attack that specifically targets the SBI of SDN systems to overwhelm the Southbound API with high traffic volume, causing network congestion and hindering legitimate traffic processing. The southbound API acts as a link between the controller and physical switches, controlling their actions. Numerous methods can be used to execute this type of attack, including network flooding or SYN flooding. In network flooding, attackers flood Southbound APIs with extreme requests, surpassing the controller's processing capacity and causing congestion, leading to delays and network performance degradation. Mitigation strategies include limiting requests, monitoring traffic for anomalies, and implementing security technologies such as firewalls and IDSs to improve SDN infrastructure protection against such attacks and improve overall security and performance.

4.3.3 Flow Table Flooding/Overflow

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.3 for detailed information on this specific attack).

4.3.4 Poisoned Network View

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.8 for detailed information on this specific attack).

4.3.5 NOS Misuse

This attack was previously addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.9 for detailed information on this specific attack).

4.3.6 Eavesdropping

This attack has previously been addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.10 for detailed information on this specific attack).

4.3.7 Man in the Middle

This attack has previously been addressed in the subsection titled "Controller-based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.11 for detailed information on this specific attack).

4.4 Attacks on the Link between Two Controllers

The attacks on the link between two controllers are described below[19][21][22]:

4.4.1 Authentication, Authorization, and Privacy

These attacks impact control planes, particularly in multiple controller system interconnections. Susceptibilities such as the absence of authorization, insufficient authentication, and unencrypted communication channels contribute to these security breaches. Initially, designed with a single controller architecture to avoid single points of failure, SDN later adopted a distributed control model called controller clusters, where each controller manages a set of switches collectively. However, moving to a multicontroller setup introduces challenges related to network privacy, authentication, authorization, and potential configuration conflicts, which pose hidden security risks to the architecture.

4.4.2 Misconfiguration

These attacks precisely target the control plane, mostly connections between controllers in multicontroller systems. These attacks arise from improper system design, which can result in security misconfiguration vulnerabilities. Security misconfiguration vulnerability occurs when the design setting or misconfiguration leaves the application module susceptible to attacks. These vulnerabilities can manifest as configuration flaws in application subsystems or components.

Attack vectors can exploit these misconfiguration flaws to execute various types of attacks. These attack vectors can take advantage of misconfiguration flaws in several ways:

1) Buffer Overflow: In the SDN environment, buffer overflow attacks can result from misconfigurations where improperly configured controllers or network devices allow attackers to flood buffers with extreme data, causing system instability or crashes. Attackers exploit this susceptibility by sending malicious packets with carefully crafted payloads to trigger buffer overflows. To counter this threat, SDN administrators should ensure suitable configurations of devices with adequate buffer sizes, keep software updated, deploy IDSs, and conduct regular security audits to recognize and address buffer overflow vulnerabilities. preemptively, enhancing the complete security and stability of SDN environments.
2) Code Injection: These attacks arise from insecure configurations of SDN controller software, allowing attackers to insert malicious code. Weak passwords unsecured interfaces, or outdated software can contribute to this susceptibility. Once injected, attackers can take control of the controller to perform harmful actions such as DOS attacks or data theft. They may manipulate the controller's programming to disrupt network traffic, access sensitive data, or cause irreversible damage.
   Mitigation strategies include keeping software up to date, restricting access, using strong authentication, isolating the controller, and conducting regular security checks to prevent code injection attacks and improve SDN security.
3) Credential stuffing/Brute Force: Brute force attacks include systematically trying numerous username and password combinations to gain unauthorized access. Weak or default passwords in SDN networks can make them susceptible to such attacks. Attackers often use automated tools to speed up the process. Implementing robust passwords, multifactor authentication, monitoring for unusual login activities and limiting login efforts can help defend against brute force attacks and improve SDN network security.
4) Command Injection: Command injection attacks in SDN environments arise when malicious commands are injected into network devices due to misconfigurations, exploiting susceptibilities in input validation. Attackers target weak authentication mechanisms, default passwords, or other misconfigurations to gain unauthorized control over the network. Mitigation strategies include using robust passwords, deactivating unnecessary services, updating firmware, implementing input validation, monitoring for suspicious activity, and employing traffic analysis tools to detect and block unauthorized traffic, enhancing overall network security and resilience.
5) Cross-site Scripting (XSS): This attack involves injecting malicious code into susceptible web applications, which are commonly seen in web-based interfaces such as SDN controller GUIs. Attackers insert JavaScript code into web pages or form fields to perform malicious actions in victims' browsers such as theft of cookies or redirecting to harmful sites. Misconfigurations in web or application servers, or SDN controller software, can enable XSS attacks. Mitigation strategies include input validation, output encoding, secure coding practices, secure frameworks, access control policies, and regular security audits to avoid and address XSS vulnerabilities efficiently in SDN environments.
6) Forceful browsing: This is known as directory traversal and includes attackers trying to access files beyond designated directories, posing an attack on SDN security by possibly granting unauthorized access to confidential data. Mitigation strategies include input sanitization, validation to avoid URL manipulation, applying access controls on directories and files, thorough access control reviews, deploying web application firewalls to filter out malicious traffic and conducting security audits and penetration testing to identify and rectify susceptibilities in SDN applications or web servers, improving the overall security posture against forceful browsing attacks.

4.4.3 Man in the Middle

This attack was previously addressed in the subsection titled "Controller-Based Attacks" of the "Attack Taxonomy" section. (Refer to subsection 4.1.11 for detailed information on this specific attack).

5. DESCRIPTION OF PUBLIC DATASETS

To identify cyberattacks, it is necessary to have an appropriate and standardized SDN dataset. We have examined multiple public datasets available for detecting cyberattacks**:**

1. KDD-CUP99 dataset, 1999: developed by MIT's Lincoln Laboratory in 1998 and 1999, is widely recognized for assessing IDSs. Originating from DARPA packet traces in a military-simulated environment, it includes 41 traffic characteristics categorized into content, fundamental, and traffic features, encompassing four attack types: DoS, U2R, R2L, and probe attacks. Data redundancy in datasets, which is particularly prevalent in training and testing sets, challenges detection accuracy, especially for low-attack instances such as R2L and U2R. Detection algorithms may exhibit bias toward high-frequency attacks such as DoS attacks because of this redundancy issue [23].

2. DARPA Dataset, 1999: The dataset was developed by Lincoln Laboratory in 1998 and 1999 to evaluate intrusion detection system performance, featuring real-time and offline assessment components. Offline testing involves analysing

network traffic and audit records to detect malicious activities in simulated networks. It comprises events such as IRC, email, and FTP and attacks such as Rootkit, DoS, and Nmap. However, limitations include not accurately reflecting real network traffic, a lack of false positives, and possible obsolescence for assessing current network attacks and infrastructure. There is also a lack of real attack datasets for analysis [24].

3. Kyoto Dataset 2006+: The dataset collected from honeypot servers between November 2006 and August 2009 features 24 statistical attributes, with 14 overlapping with the KDD dataset. It collects regular and malicious traffic by introducing a second server, aiming for additional realistic datasets. However, it suffers from an unbalanced class distribution with mainly malicious data and lacks documented attack types, leading to biased intrusion detection assessment. The limited regular traffic, including only email and DSN traces, represents a small percentage of overall activity, resulting in an incomplete depiction of internet traffic. The dataset's separation of regular and malicious traffic contexts makes it an artificial and disjointed dataset, complicating accurate valuations of SDN internet infrastructure susceptibilities [25].

4. CAIDA DDoS2007: The dataset was developed by the Cooperative Association of internet Data Analysis and comprises three discrete datasets. The CAIDA OC48 dataset features varied data types observed over the OC48 link in San Jose, containing approximately 100 GB of unprocessed data traffic. The CAIDA DDoS dataset contains one hour of DDoS attack traffic separated into five-minute pcap files. Finally, the 2016 CAIDA internet trace dataset includes inert traffic from CAIDA's Equinix-Chicago monitor, leveraging fast internet infrastructure. However, these datasets, which are tailored for specific attacks, lack protocol data, payload, and destination information, present challenges and are not considered effective benchmarking datasets [26].

5. NSL-KDD Dataset, 2009: Tavallaee et al. presented the NSL-KDD dataset as an improvement over the KDD'99 dataset, addressing issues such as data redundancy. Divided into training and testing sections, the test set emphasizes attacks, possibly missing 17 from training data. However, both datasets are outdated for current network trends and lack representation of new attack patterns. Attributes unrelated to SDN are present in both datasets, impacting their utility. Past studies that utilized NSL-KDD in SDN setups focused on limited characteristics and achieved poor classifier performance in identifying malicious traffic, leading to low detection rates and high false alarms. The detection of diverse attacks such as R2L and U2R, which require content features not available through OpenFlow, remains a challenge in SDN applications that rely on these datasets [27].

6. ISCX2012 Dataset: Shiravi et al. advanced a dataset using simulated network profiles: Alpha for making attack traffic and Beta for regular traffic. It covers network protocols such as IMAP, SMTP, HTTP, SSH, FTP, and POP3, including full packet data. With 20 packet attributes focused on DoS and brute force attacks, the dataset's DoS attack diversity is limited and lacks susceptibility coverage across OSI layers. Limited to HTTP traffic, it may not reflect real traffic trends dominated by HTTPS. Furthermore, like KDD'99 and NSL-KDD, the dataset's feature set from the OpenFlow protocol may not serve ML valuations [28].

7. Dataset by Alkasassbeh et al.: The dataset developed by Alkasassbeh et al. includes a modern DDoS attack dataset with 2,160,668 records and 27 features. It encompasses various types of attacks, such as HTTP, UDP, Smurf, and SiDDoS. Many researchers have utilized this dataset to detect DDoS attacks in SDN environments via ML algorithms [29].

8. UNSW-NB15 dataset: The dataset was produced by the Australian Center for Cyber Security. It contains nearly two million records and a total of 49 characteristics obtained through the use of Bro-IDS, Argus tools, and other newly created methods. The dataset comprises numerous attack types such as Worms, Shellcode, Reconnaissance, Port Scans, Generic, Backdoor, DoS, Exploits, and Fuzzers. Numerous researchers have employed this dataset in intrusion detection via ML algorithms [30].

9. CICIDS2017 Dataset: Sharafaldin et al. created a CICIDS2017 dataset featuring six attack profiles, offering new threat scenarios compared with earlier datasets. With an equivalent number of flow-based characteristics, it presents an HTTPS beta profile to address the increasing HTTPS usage online. However, the dataset derived from ISCX2012 faces challenges such as differences in feature numbers, with 80 flow-based features compared with ISCX2012's 20 packet features. The identified faults include empty class labels and incomplete data entries [31], whereas duplicate records may delay effective training of the IDS [13].

10. CSE-CIC-IDS2018 dataset: Developed in collaboration by CSE and CIC, the CSE-CIC-IDS2018 dataset expands on CICIDS2017, incorporating over 80 flow-based features resulting from traffic captured via CICFlowMeter-V3. By applying profiles to categorize traffic into regular (B-profiles) and irregular/attack (M-profiles) profiles, the dataset mirrors attack scenarios from CICIDS2017. However, it inherits the limitations and shortcomings present in the previous CICIDS2017 dataset [32].

11. CICDDoS2019 dataset: The dataset was developed by the Canadian Institute for Cybersecurity, offering a realistic portrayal of contemporary DDoS incidents akin to real-world PCAP data. CICFlowMeter-V3 for network monitoring extracts 80 features with labelled flows, covering aspects such as IPs, protocols, timestamps, and attack types. Simulating

abstract user behavior through protocols such as FTP, HTTP, and others adds depth and realism to the dataset, featuring a varied array of current DDoS attacks such as LDAP, NTP, and SYN floods [33].

12. SDNTrafficDS Dataset, 2019: Myint et al. introduced the SDNTrafficDS dataset in 2019, which is tailored for SDN traffic settings and features UDP and SYN flood attacks alongside regular traffic. Data were sourced from OpenFlow switches via an Open Daylight controller, with researchers extracting five crucial features to analyse network traffic. These selected features offer important insights into traffic patterns within the SDN domain, enhancing the understanding of dataset characteristics [6].

13. Kaggle DDoS 2019 dataset: Researchers can access the DDoS attack dataset on Kaggle created by Prasad et al., which combines flows extracted from public intrusion detection system datasets by CIC Canada. This dataset merges DDoS flows with normal flows, creating a large dataset with 84 features capturing different DoS and DDoS attacks via various tools. By using this dataset alongside others available on Kaggle, researchers can enrich their studies and deepen their knowledge of DDoS attacks [34].

14. InSDN dataset, 2020: This dataset was developed by Elsayed et al. in 2020 and serves as a specific dataset for validating IDSs in the context of SDN with various attacks such as DDoS and Web attacks generated via multiple tools; it features over 80 attributes, including 48 specifics to SDN. In addition to attack data, it covers typical network activities such as DNS and HTTP. While addressing the limits of existing datasets, this dataset is not universal and has been tested solely with ONOS controllers, lacking valuation with other controllers [35].

Table 6 provides a high-level overview of the strengths and weaknesses of each dataset. It is important to carefully consider the specific requirements of the research project when selecting a dataset for SDN cyberattack detection.

TABLE VI. COMPARISON OF PUBLIC DATASETS FOR SDN CYBERATTACK DETECTION.

Dataset	Relevance to SDN	Completeness	Limitations
KDD-CUP99	Limited	Incomplete	Data redundancy, outdated, not representative of modern attacks
DARPA	Limited	Incomplete	Outdated, not representative of modern attacks, lack of real attack data
Kyoto 2006+	Limited	Incomplete	Imbalanced class distribution, lack of documented attack types, limited regular traffic
CAIDA DDoS2007	Limited	Incomplete	Tailored for specific attacks, lack of protocol data, payload, and destination information
NSL-KDD	Limited	Incomplete	Outdated, not representative of modern attacks, attributes unrelated to SDN, limited feature set for ML
ISCX2012	Limited	Incomplete	Limited DoS attack diversity, limited to HTTP traffic, feature set may not suffice for ML
Alkasassbeh et al.	High	Complete	Modern DDoS attack dataset, various attack types
UNSW-NB15	High	Complete	Diverse attack types, numerous characteristics
CICIDS2017	High	Complete	New threat scenarios, HTTPS Beta profile, equal number of flow-based characteristics
CSE-CIC-IDS2018	High	Complete	Expands on CICIDS 2017, over 80 flow-based features, mirrors attack scenarios from CICIDS 2017
CICDDoS2019	High	Complete	Realistic portrayal of contemporary DDoS incidents, a diverse array of current DDoS attacks
SDNTrafficDS	High	Complete	Tailored for SDN traffic settings, important insights into traffic patterns within SDN domain
Kaggle DDoS 2019	High	Complete	A substantial dataset with 84 features capturing different DoS and DDoS attacks
InSDN	High	Complete	Specialized dataset for validating IDSs in the context of SDN, over 80 attributes, including 48 specifics to SDN

Notes: Relevance to SDN refers to the extent to which the dataset contains data relevant to SDN environments, such as OpenFlow traffic and SDN-specific attacks. Completeness refers to the extent to which the dataset covers a wide range of attack types and network scenarios. Limitations refer to any shortcomings of the dataset, such as outdated data, limited attack diversity, or lack of specific features.

The majority of publicly available datasets consist of traditional network data, with SDNTrafficDS and InSDN being exceptions focused on SDN traffic and diverse attack scenarios. While they offer valuable resources for studying intrusion

and anomaly detection in SDN, the lack of standardized SDN datasets poses a challenge. Converting traditional network datasets to flow-based SDN networks may not accurately capture SDN behavior during cyberattacks. Nevertheless, traditional datasets such as Alkasassbeh et al., UNSW-NB15, CICIDS 2017, CSE-CIC-IDS2018, CIC DDoS 2019, and Kaggle DDoS 2019 can still be useful for benchmarking and understanding SDN-specific attacks. Creating SDN-specific datasets is encouraged to improve cyber-attack detection capabilities.

6. ANALYSIS OF FEATURE SELECTION TECHNIQUES

The entropy, ML, DL, and FL methods have contributed to improvements in cyber-attack detection, but their effectiveness varies depending on the context. However, while ML algorithms are effective, they often operate on datasets with numerous features, some of which may not be crucial for detecting attacks. To increase efficiency and accuracy, it is essential to identify and select the most relevant features from the available dataset, as proposed by Polat et al. through embedded, wrapper, and filter-based feature selection techniques [36]. These techniques are illustrated in Figures 10, 11, and 12.

Filter-based methods: Filter-based methods focus on the inherent properties of features and use statistical techniques to select the most relevant ones. This technique serves as a preprocessing phase and is faster than other methods because it does not require model training. It calculates metrics such as the Fisher score, IG, variance threshold, correlation coefficient, and chi-square test. The efficacy of this approach has been demonstrated in various studies, such as [37], where the use of filter-based feature selection led to increased detection accuracy.

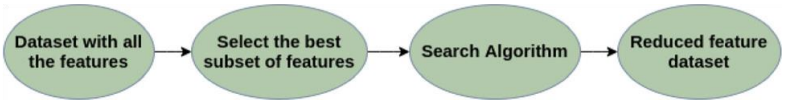

Fig. 10. Filter-based feature selection method.

Wrapper-based method: The wrapper-based approach evaluates how the inclusion of specific features benefits the performance of the classifier. ML classifiers are trained on subsets of features, and the selection process continues until an ideal subset is identified. This method is computationally expensive but has proven to be more effective than statistical approaches. Examples of wrapper-based feature selection algorithms include backwards and recursive feature removal, as well as forward feature selection.

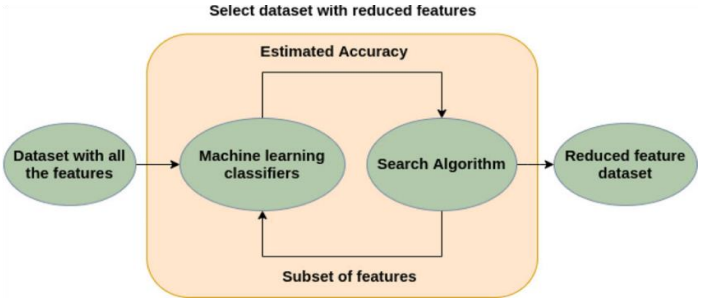

Fig. 11. Wrapper-based feature selection method.

Embedded-based method: The embedded method combines the qualities of filter-based and wrapper-based methods to enhance result prediction. Each feature selection method is coupled with a different algorithm, which helps achieve the desired goals. Algorithms with built-in feature selection strategies are utilized. This approach performs feature selection and classification operations concurrently. Examples of embedded-based approaches, such as L1 (LASSO) regularization and DT, have been utilized in studies such as [38], leading to enhancements in detection accuracy.

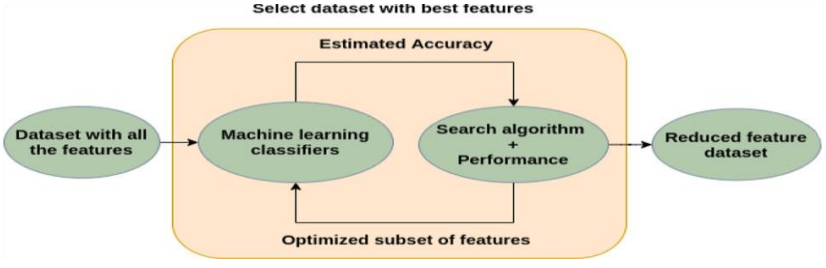

Fig. 12. Embedded-based feature selection method

Feature selection methods have unique advantages and limitations that affect their suitability for cyber-attack detection in SDN environments. The filter-based method is efficient but may overlook interactions, wrapper-based methods offer accuracy at higher computational cost and embedded methods integrate both approaches. Table 7 provides a comprehensive overview of the feature selection algorithms used in cyber-attack detection studies, including IG, PCA, and chi-square tests. Researchers utilize these algorithms on public or self-generated SDN-based datasets to select the best features for cyber-attack detection. As presented in Section 5, generating SDN-based datasets is crucial because of their limited availability. Obtaining the optimal feature set through feature selection algorithms on SDN-based datasets can increase the efficiency and accuracy of cyber-attack detection. This approach can benefit researchers by streamlining the detection process and improving overall performance.

TABLE VII. FEATURES AND FEATURE SELECTION UTILIZED FOR CYBER-ATTACK DETECTION.

Re.	No. of Feature	Feature Selection	Features
[39]	19	--	Src_ip, Dst_ip, Dst_mac, Packet Rate, PktCount, Flows, ByteCount, Protocol, Duration_sec, Switch-id, TX_Bytes, RX_Bytes, Port Number, Src_mac, Tx_kbps, Rx_kbps, Tot_kbps, Pktperflow, Byteperflow.
[16]	80	Chi-squared (x2)	Source IP address, Destination IP address, Source port, Destination port, Protocol, Packet size, Flow duration, Number of packets, Number of bytes.
[40]	1	--	source IP address
[41]	2	Based on packet header features	source IP, destination IP.
[42]	1		Renyi entropy of packet rate.
[43]	5	Manual selection based on typical network traffic characteristics	Average no. of packets, average no. of packet bits, Growth rate of port, Growth rate of flow, Growth rate of source IP.
[44]	3	Based on information entropy	Source IP address, Packet length, Protocol.
[45]	2	Based on the characteristics of DDoS attacks (increase in USIP and decrease in NUDIP during attacks)	Number of unique source IP addresses (USIP), Normalized number of unique destination IP addresses (NUDIP).
[46]	2	--	Destination address entropy and source address entropy.
[3]	6	Entropy is chosen as the feature based on its ability to measure the randomness of destination addresses.	Source IP, Destination, IP, Source port, Destination port, protocol type.
[47]	--	--	Bandwidth usage on network ports, Transmitted Packets, Received Packets,Transmitted bytes, Received bytes, Errored packets.
[4]	--	--	No. of packets per second, Source host, Destination host.
[38]	8	Dynamic, based on joint entropy calculation	Source IP address, Destination IP address, Source port, Destination port, Protocol type, Packet size, Time To Live, TCP flag.
[48]	--	--	Payload, Packet type, Topology type, Destination port, Source port, Total packet sent.
[49]	1	Entropy of destination IP addresses is chosen as the feature for attack detection.	destination IP addresses.
[50]	89	--	The features used include messages exchanged between the controller and network devices, nonmatching flows, flow modification messages from the controller, flow table entries, the CPU utilization of transmission equipment, and average latency for flow creation.
[5]	--	--	unknown IP destination address, packets interarrival time, TLP header, ToS header, switch's stored capacity, average rate of packets with unknown destination addresses, IP Options header, average number of flows.
[51]	23	SelectKBest (Chi2) method	dt, switch, src, Pktcount, bytecount, Flows, packetins, Protocol, tx_kbps, rx_kbps, tot_kbps.
[52]	2	--	Host_time (peak-time or off-peak-time), Number of hosts connected per second.
[53]	50	Maximum likelihood estimation and linear interpolation	Flow-based features, such as Flow Bytes/s, Flow Packets/s, and Flow Duration.
[17]	3	--	jitter, throughput, and response time.
[54]	41	--	Duration, Protocol-type, Src-bytes, Srv-count, Dst-host-same-src-port-rate
[18]	16	Selected based on their significance in attack detection.	Packet count at switches, Packet_in messages to controller, flow entries in switches, etc.
[55]	4		Duration time, Packets number, Relative dispersion of match bytes, Relative dispersion of packet interval.
[6]	5	*Volumetric features: ANPI, ANBI *Asymmetric features: VPI, VBI, ADTI.	Average number of flow packets in the sampling interval (ANPI), Average number of flow bytes in the sampling interval (ANBI), Variation of flow packets in the sampling interval (VPI),

Re.	No. of Feature	Feature Selection	Features
			Variation of flow bytes in the sampling interval (VBI), Average duration of traffics in the sampling interval (ADTI).
[56]	2	--	Time Duration, Packet flow.
[57]	7	Correlation-based feature selection	Duration, protocol type, service, source bytes, destination bytes, count, and server count.
[58]	6	Based on previous research and analysis of DDoS attack characteristics.	Speed of source IP, speed of source port, standard deviation of flow packets, standard deviation of flow bytes, speed of flow entries, ratio of pair-flow.
[19]	5	--	Entropy of source IP address, entropy of source port, entropy of destination port, entropy of packet protocol, and total number of packets.
[59]	3	Entropy-based feature extraction	Source IP address entropy, Destination IP address entropy, Source port entropy, Destination port entropy
[37]	5	IGR and Chi-square-based cross-feature selection	bytecount, pktcount, dt, tot_dur, dur.
[33]	115	--	Source Port, Destination Port, Protocol, Flow Duration, Packet Lengths, Down/Up Ratio, Active Mean, Active Std, Active Max, Active Min, Idle Std, Inbound, Label.
[34]	88	--	Bits/s, packets/s, source IP entropy, source port entropy, destination IP entropy, destination port entropy, and 82 other digital features.
[22]	83	Feature weighting and threshold tuning	Network flow features such as packet size, flow duration, and source/destination IP addresses.
[60]		--	Traffic flow characteristics, such as packet size, interarrival time, and source/destination IP addresses.
[61]	125	--	lists various flow-based and statistical features used in the model.
[1]	20	--	20 features (16 automatically extracted flow table features and 4 manually constructed statistical features) Flow table features: Src_ip, Dst_ip, Src_mac, Dst_mac, In_port, Src_port, Dst_port, protocol, duration, ByteCount, PacketCount. Manually constructed statistical features: Avg f, lowSpeed, Avgduration, AvgpacketSize, RatioasymmetricFlow.
[62]	83	--	Packet characteristics.
[63]	4	--	Protocol, source address, packet length, port number, and packet payload.
[26]	76	--	Packet characteristics.
[13]	86 reduced to 4	--	Backwards packet length (B.packet Len), Standard deviation(Std), Flow Duration, Average Packet Size, Flow Inter Arrival Time (IAT).
[64]	22	--	primarily statistical and count features.
[65]	4	Random Forest Regressor	Backwards Packet Length (B. packet Len) Std, Flow Duration, Avg Packet Size, and Flow inter arrival time (IAT) Std.
[66]	6	Entropy calculation for nominal attributes (IP and port)	Bits/s, packets/s, source IP entropy, destination IP entropy, source port entropy, destination port entropy.
[67]	41	--	Basic features, content-based features, traffic features, and host-based features from the KDD99 dataset.
[68]	6	--	Average number of packets per flow, Average packet size per flow, Packet change rate, Flow change rate, Average number of packets per second, and Average number of bytes per second.
[69]	100	LOA	timely features, connection-based features, and content-based features.
[65]	4	Selected the four best features for DDoS detection based on previous research.	B.packet length Std, Avg. packet size, flow duration, and IAT Std.
[25]	15	--	The system extracts 15 features from each traffic session, encompassing packet sizes, byte sizes, counts, rates, intervals, IP addresses, TTL, port numbers, flags, window sizes, and payload sizes.
[43]	20	--	packet header fields, payload features, and temporal features.
[70]	5	Based on importance scores.	The best five features for each DDoS type.
[29]	20	--	Flow-based features, Protocol-based features, Content-based features, Statistical features.
[24]	71	--	Basic features, Flow features, Content features, Link traffic features.
[71]	48 reduced to 18	PCA	Protocol, Bwd-IAT-Max, Flow-duration, Bwd-IAT-Tot, Tot-Fwd-Pkts, Pkt-Len-Var, TotLen-Fwd-Pkts, Idle-Std, Fwd-Pkt-Len-Mean, Idle-Max, Flow-Byts/s, Idle-Min, Flow-Pkts/s, Idle-Mean, Flow-IAT-Mean, Flow-IAT-Max, Flow-IAT-Std, Fwd-IAT-Tot.
[32]	41	--	Intrinsic, content, time-based traffic, and host-based traffic features.
[72]	12	Min–max normalization	P addresses, ports, protocol, flags, flow duration, L7 protocol.

7. A PROPOSED TAXONOMY OF CYBERATTACK DETECTION TECHNIQUES

The taxonomy of SDN cyber-attack detection and mitigation techniques categorizes various studies into five main categories: statistical model/entropy-based techniques, ML-based techniques, DL-based techniques, and federated learning detection techniques. Tables 8.1 to 8.4 provide a summary of the studies within each category.

7.1 Cyber-attack detection via entropy analysis techniques

Entropy-based techniques have been used to detect cyberattacks in SDN. The programmability of SDN enables the extraction and analysis of network flow statistics, which can aid in cyberattack detection.

In this study [39], a cooperative detection technique was proposed to identify DDoS attacks in SDN environments. This technique combines information entropy and advanced learning methodologies to distribute detection responsibilities between data and control planes. Initially, the information entropy detection module is deployed on the switch, while a sophisticated detection module utilizing the CNN-BiLSTM architecture with batch normalization and an attention mechanism is executed on the controller. The study focused on detecting SYN and ICMP flood attacks targeting both the data and control planes via a single controller. The experimental results conducted on a publicly available DDoS attack SDN dataset yielded an accuracy rate of 99.54%, a low FAR of 0.50%, and a high DR of 99.33%. By distributing the detection procedures, this approach reduces the controller's workload and improves the detection process's overall effectiveness. However, it was not tested in an SDN test bed.

The authors of [16] proposed a hybrid model for detecting and classifying DDoS attacks in multicontroller SDN environments utilizing the POX controller. This model uses an entropy-based method for initial detection and a chi-square test for feature selection and then employs RNN, MLP, GRU, and LSTM models for fine-grained packet-based classification. This model achieved a high accuracy rate of up to 99.42% when tested with the publicly available CICDDoS2019 dataset, demonstrating its effectiveness in detecting and classifying various DDoS attacks, including flood attacks (TCP-SYN, UDP, ICMP), DNS, and NTP amplification. Additionally, the model integrates a mitigation process to safeguard network devices from DDoS attacks. However, the solution is focused solely on detecting and classifying DDoS attacks, lacking the ability to identify other types of cyber threats, such as malware or phishing attacks.

In [40], the authors proposed a novel entropy-based approach for detecting DDoS attacks. They utilized a self-generated private dataset to detect both low-rate and high-rate attacks on the control plane, employing a single POX controller SDN environment. This approach proves its effectiveness in identifying both low-rate and high-rate DDoS attacks that target single or multiple victims, achieving high DR and low FPR. The experimental findings reveal that the approach significantly improves the DR for high-rate DDoS attack traffic detection compared with low-rate traffic, with enhancements ranging from 6.25% to 20.26%. Additionally, this approach effectively reduces the FPRs for high-rate DDoS attack traffic detection compared with low-rate traffic, resulting in reductions ranging from 64.81% to 77.54%. This approach uses a static threshold and employs the source IP address as a feature for entropy calculation. However, it relies on a fixed threshold to differentiate between regular and attack traffic, which may not be suitable for all situations and needs adjustment on the basis of network conditions.

The defense framework named CC-Guard was proposed in [15] and aims to safeguard SDN controllers against DDoS attacks. CC-Guard employs a comprehensive and coordinated approach that includes real-time attack detection, switch migration, anomaly detection, and mitigation. The anomaly detection module in CC-Guard adopts a two-stage process: it uses entropy-based coarse-grained judgment to swiftly identify suspicious traffic, followed by DL (CNN, GRU)-based classification of suspected traffic. The performance was evaluated via a simulation multicontroller using the RYU controller, and CC-Guard achieved an accuracy rate of 99.63. The evaluation results prove that CC-Guard effectively detects and mitigates DDoS attacks targeting SDN controllers while ensuring efficient utilization of network resources. This framework provides comprehensive protection against an extensive range of DDoS attacks. However, the CC-Guard defense framework may not be fully effective against highly sophisticated DDoS attacks.

In [41], the authors proposed an innovative detection approach that combines a dynamic threshold with a rule-based detection mechanism on a self-generated private dataset to detect DDoS attacks on single or multiple victims on the control plane via a single POX controller. The dynamic threshold is designed to adapt to varying attack traffic rates, thereby increasing the detection accuracy and reducing the FP. This approach results in high DR and low FPR across all attack scenarios, including both low- and high-rate DDoS attacks that target single or multiple victims.

In [42], a self-feedback dynamic thresholding system was proposed for two-stage detection of cyberattacks on a self-generated private dataset to detect DDoS attacks on the SDN control plane with the utilization of the floodlight controller. This system incorporates a trigger mechanism to minimize the occurrence of invoking resource-intensive detection algorithms. The trigger threshold is dynamically adjusted on the basis of previous outcomes of both trigger and detection processes. This system is evaluated via a floodlight controller, Mininet, and Hping3. The results demonstrate that the system effectively decreases the number of calls to the detection algorithm without compromising the quality of the detection

results. Furthermore, the system is robust to various initial thresholds. However, it requires careful selection of the initial threshold value and is not effective against all types of attacks other than DDoS.

In this study [43], a cooperative scheme is proposed for detecting DDoS attacks in SDN networks. This scheme leverages edge switches for preliminary detection and a controller for precise detection. This scheme uses information entropy in the edge switches to enable real-time traffic monitoring while employing an RF in the controller for accurate attack detection. By distributing detection tasks, this approach alleviates the workload on the controller, enhances the attack detection speed, and ensures high detection accuracy. The simulation results validate efficiency of the system in detecting ICMP and SYN flood attacks across both the control and data using multiple controllers with the RYU controller. This leads to reduced controller overhead and minimized detection delay. However, the generalizability of the findings to other types of DDoS attacks was limited.

In [44], the DDoS attack detection method for SDN was proposed which uses information entropy and DL, including the CNN, SVM, DNN, and DT methods. The method is implemented on the publicly available CICIDS2017 dataset to detect DDoS attacks on the data plane via the POX controller. The method employs a two-level detection approach to detect DDoS attacks. The initial detection phase, which is based on information entropy, is performed by the controller to determine the switch through which suspicious traffic enters the network. Fine-grained packet-based deep detection uses a CNN to distinguish DDoS attack traffic. The controller issues a flow table to implement a defense strategy for intercepting attack traffic. The proposed method attains high accuracy rate of 98.98% when compared with the DNN, SVM, and DT approaches. However, it was not effective against all types of DDoS attacks.

In [45], the scheme for detecting and defending against DDoS attacks in SDN environments was proposed. This scheme uses time series analysis and uses entropy, autoregressive integrated moving average (ARIMA), chaotic theorem, and exponential filter methods on the MAWI Working Group Traffic Archive public dataset to detect DDoS attacks targeting the control plane. The detection system operates with a single controller via the RYU controller. This scheme uses a model that combines upcoming traffic feature forecasting, chaos theory, exponential filtering, and the dynamic threshold method to detect sudden changes in the network. The experimental results demonstrate that the proposed algorithm achieves high detection rate of 98.82% while maintaining low FAR of 0.8%. However, this approach requires careful parameter tuning and considers a limited feature vector.

In [46], DDoS detection and defense mechanisms are based on cognitive-inspired computing with dual address entropy and SVM methods. These methodologies are applied to the publicly available DDoS attack 2007 dataset to detect DDoS flooding attacks across both the control and data planes. The system is designed to operate with multiple controllers via ONOS controller. This mechanism enables real-time detection and defense in the initial phases of a DDoS attack, ensuring timely restoration of normal communication. The experimental results demonstrate that the proposed mechanism not only achieves rapid attack detection but also has high DRs and low FPRs. Moreover, it effectively implements suitable defense and recovery measures when an attack has been identified. However, there are concerns about an increased risk of misjudgment and the perceived inefficiency of the recovery algorithm.

In [3], the Shannon entropy-based algorithm was introduced on a self-generated private dataset for the detection of DDoS and UDP flooding attacks on the data plane of the SDN environment operated by a single controller. The algorithm leverages the entropy of destination IP addresses in incoming packets to identify possible attacks. An alert is triggered by the algorithm whenever the entropy level falls below a predetermined threshold. To mitigate attacks once detected, the algorithm benefits from the flexibility presented by the OpenFlow protocol and POX controller. However, the study's results are derived from a small-scale testbed and may not be directly applicable to larger network settings.

[47] The authors of employed flow statistics monitoring (sflow management model) and threshold-based detection on a privately self-generated dataset to detect UDP flooding attacks targeting the SDN control plane. These attacks have the potential to saturate the control plane and disrupt services. To address this challenge, the authors propose the secure flow management (SFM) model. The SFM model continuously monitors flow statistics and implements ingress and egress policies to detect and mitigate UDP flooding attacks in real time. Operating in a multicontroller environment, the SFM model offers scalability and efficiency. The evaluation results prove that the SFM model achieves a high detection rate and effectively mitigates UDP flooding attacks, restoring normal network communication. Furthermore, this model suffers minimal overhead, making it appropriate for practical deployment in SDN networks. However, evaluation is limited to a specific network topology and traffic patterns and is not effective against sophisticated or adaptive attacks.

In [4], the authors address the substantial threat posed by DDoS attacks in SDN networks, which can affect network resource saturation and service disruption. To address this challenge, the authors propose a real-time detection and mitigation system that uses sFlow, threshold-based detection, and rule-based mitigation on a privately self-generated dataset. This system is designed to detect DDoS flood and ICMP flood attacks targeting both the control and data planes via a single controller with a floodlight controller. This system analyses network traffic in real time and generates rules to mitigate DDoS attacks efficiently. The evaluation results demonstrate that the system achieves high DR and successfully restores normal network

communication by mitigating DDoS attacks. Additionally, this system suffers minimal overhead, rendering it suitable for practical deployment in SDN networks. However, it was evaluated in a specific network topology with a focus on ICMP flooding attacks.

In [38], the authors employed joint entropy calculation, dynamic rule generation, and statistical mitigation methods on the MAWI Working Group Traffic Archive public dataset to detect various DDoS attacks, including TCP-SYN, UDP, ICMP flood, SQL Slammer, Worm, DNS amplification, and NTP amplification, which target both the control and data planes. They accomplished this using a single controller with the POX controller, which is both lightweight and effective in identifying early-stage threats. This approach analyses the entropy of source IP addresses in incoming packets to detect DDoS attacks. This paper evaluates the efficacy of this approach via eight simulation scenarios, which represent various attack scenarios and traffic rates. The results prove that the entropy-based approach improves DR for high-rate DDoS attacks compared with low-rate DDoS attacks. Additionally, it reduces the FPR for detecting high-rate DDoS attacks compared with low-rate DDoS attacks. However, the solution of this study, which is based on statistical analysis, is not effective in detecting advanced attacks that can bypass statistical detection. It also introduces memory overhead in the switch and traffic on the control channel between the switch and the controller.

In [48], a novel detection mechanism for low-rate DDoS attacks in SDN networks was introduced. This mechanism uses GE and ID metrics to identify abnormalities in traffic patterns, signalling potential low-rate DDoS attacks. This proposed detection mechanism is evaluated via simulated traffic (self-generated private dataset) to detect low-rate DDoS attacks that target both the control and data planes with a single controller via the POX controller. The results demonstrate enhanced accuracy compared with existing methods. However, it was limited to low-rate DDoS attacks and was not effective against high-rate DDoS attacks.

In [49], an innovative method was introduced for detecting DDoS attacks in SDN control planes. This method uses entropy metrics, specifically GEs, on a self-generated private dataset to detect DDoS attacks on the control plane with a single controller via the POX controller. Compared with the conventional Shannon entropy approach, the GE-based method effectively reduces the FP, particularly for low-rate DDoS attacks. This method is evaluated with a simulated SDN network and exhibits promising results in terms of detection accuracy and efficiency. However, the proposed method assumes that attack traffic has a distinct entropy distribution but lacks experiments on an SDN testbed. While it shows quick detection capabilities, it may not be effective against all types of DDoS attacks. Furthermore, setting the optimal threshold for the GE proves to be challenging.

Table 8 summarizes studies that utilized entropy-based methods for cyber-attack detection in SDN.

TABLE VIII. ENTROPY-BASED DETECTION

Ref.\ Year	Method	Dataset Used/Private, Public	Type of Attacks	Target Plane	No. of Controller	Controller Type	Scope	Results
[39] 2023	Entropy, CNN-BiLSTM.	DDoS attack SDN dataset/public	SYN flood and ICMP flood	Data and Control plane	Single controller	--	Detection	ACC 99.54%, FAR 0.50%, DR 99.33%
[16] 2023	Entropy algorithm, RNN, MLP, GRU, LSTM	CICDDoS2019 dataset/public	TCP SYN flood, UDP flood, ICMP flood, DNS amplification, NTP amplification.	Control plane	Multicontroller	POX	Detection and Classification	ACC 99.42%, PREC 99.40%, Recall 99.46%, F1-score 99.42%
[40] 2023	Entropy algorithm	Self-generated dataset/private	DDoS attacks (low-rate and high-rate)	Control plane	Single controller	POX	Detection	This approach significantly improves DR for high-rate DDoS attack traffic detection compared to low-rate traffic.
[15] 2023	Entropy, GA, DL (CNN, GRU)	Not mention	DDoS attacks	Control plane	Multicontroller	RYU	Detection	ACC 99.63%, PREC 99.77%, Recall 99.58%, F1-score 99.67%, FPR 0.30%
[41] 2022	Entropy (Renyi Joint), Entropy Exponentially Weighted Moving Average (EWMA)	Self-generated dataset/private	DDoS attacks (low-rate and high-rate) on single or multiple victims	Control plane	Single controller	POX	Detection	This approach showcases high DR and low FPR across all attack scenarios.

Ref.\ Year	Method	Dataset Used/Private, Public	Type of Attacks	Target Plane	No. of Controller	Controller Type	Scope	Results
[42] 2021	Renyi entropy, Dynamic thresholding algorithm.	Self-generated dataset/private	DDoS attack	Control plane	Single controller	Floodlight	Detection	This approach exhibits robustness to various initial thresholds and effectively decreases the number of calls to the detection algorithm without compromising the quality of detection results.
[43] 2021	Entropy, RF.	Not mention	ICMP flood, SYN flood.	Control and data plane	Multicontroller	RYU	Detection	This approach is effective in detecting attacks.
[44] 2020	Entropy, CNN, SVM, DNN, DT.	CICIDS2017 dataset/public	DDoS attacks	data plane	--	POX	Detection and mitigation	ACC 98.98%, PREC 98.99%, Recall 98.96%, F1-score 98.97%
[45] 2020	Entropy, Auto Regressive Integrated Moving Average (ARIMA), Chaotic Theorem, Exponential Filter	MAWI Working Group Trac Archive/public	DDoS attacks	Control plane	Single controller	RYU	Detection and mitigation	ACC 98.82%, F1-score 98.8%, FPR 0.8%, TPR 98.46%
[46] 2019	Entropy, SVM.	DDoS attack 2007 dataset/public	DDoS flooding attacks.	Control and data plane	Multicontroller	ONOS	Detection and mitigation	This approach exhibits high DRs and low FPR
[3] 2019	Shannon Entropy	Self-generated dataset/private	DDoS, UDP flooding.	Data plane	Single controller	POX	Detection and mitigation	--
[47] 2019	Sflow management model, Threshold-based detections	Self-generated dataset/private	UDP flooding.	Control plane	Multicontroller.	RYU	Detection and mitigation	This approach achieves high DR and effectively mitigates UDP flooding attacks.
[4] 2018	Sflow, Threshold-based detection, Rule-based mitigation	Self-generated dataset/private	DDoS flood, ICMP flood.	Control and data plane	Single controller	Floodlight	Detection, mitigation	This approach achieves high DR and successfully restores normal network communication by mitigating DDoS attacks.
[38] 2018	Joint entropy calculation, Dynamic rule generation, Statistical mitigation.	MAWI Working Group Traffic Archive/public	TCP SYN Flood, UDP Flood, ICMP Flood, SQL Slammer, Worm, DNS Amplification, NTP Amplification.	Control and Data plane	Single controller	POX	Detection and mitigation	This approach improves DR for high-rate DDoS attacks compared to low-rate and reduces FPR for detecting high-rate DDoS attacks
[48] 2018	GE, ID	Self- generated dataset/public	Low-rate DDoS attacks.	Control and Data plane	Single controller	POX	Detection	This approach demonstrates enhanced accuracy compared to existing methods.
[49] 2017	Entropy calculation, GE calculation	Self-generated dataset/private	DDoS attacks.	Control plane	Single controller	POX	Detection	This approach proves the GE method is effective in reducing FP, particularly for low-rate DDoS attacks.

7.2 Cyber attack detection via the machine learning technique

ML techniques employed for detecting cyber-attacks in SDN have proven efficient by various researchers. This section analyses various ML algorithms to detect cyberattacks, as shown by the authors in [50]. This study presents a multistage learning model that combines 1D-CNN and DT-based classification on a privately self-generated dataset to detect DDoS attacks in SDN-based SCADA systems on both the control and data planes with multiple controllers. The model achieves a high accuracy rate of 97.80% in detecting DDoS attacks, highlighting its effectiveness in identifying such malicious activities. However, the study was conducted on a specific network topology and may not be generalizable to all SDN-based SCADA systems. The efficiency of the framework depends on the variety and complexity of attack scenarios in the dataset. Additionally, the 1D-CNN model employed in the study may have challenges capturing data patterns because of its restricted consideration of local structures.

In [5], the DL technique was introduced for detecting DDoS attacks in both the control and data planes of the SDN environment. This technique uses novel features derived from traffic statistics and employs a DL model based on an AE with a BGRU on the publicly available NSL-KDD dataset and a privately self-generated dataset to detect various attacks on both planes with a single POX controller, achieving an accuracy rate of 99.87%. Moreover, it monitors the average arrival bit rate of switches with unknown destination addresses in the data plane and incorporates a trust value mechanism to mitigate the impact of DDoS attacks. However, the FloodDefender system, which uses SVM, KNN, DT, NB, and RF for detection, may have a high deployment cost and could be unsuitable for real-time detection because of its computational complexity and reliance on large datasets for training DL models.

In [73], a framework that uses intelligent techniques for predicting attacks in SDN systems was introduced. The prime focus of the framework is to enable early detection of abnormal attacks on the SDN environment. When malicious traffic is identified, the AI module employs ML or DL models to identify and stop the source of the attack. The architecture allows for assessment of various ML and DL classification techniques to identify different types of network attacks. The proposed framework is evaluated via the InSDN dataset, which employs several learning models such as the RF, AB, KNN, NB, DT, and LR classifiers, which are applied to the public InSDN dataset to detect multiple attacks on the control and data planes with a single controller. Additionally, DL models, including deep CNNs and long short-term memory (LSTM) are considered. The results highlight that the proposed Deep CNN model for multiclass attack data classification achieves high accuracy of 99.85% compared with the RF, AB, KNN, NB, DT, LR, and LSTM classifiers which achieve accuracy levels of 95%, 88%, 97%, 93%, 90%, 98%, and 88.31%, respectively.

In [51], the authors address the serious threats posed by DDoS attacks on SDN environments. To address this issue, the authors propose an ML-based model that uses a public DDoS attack SDN dataset to detect DDoS attacks that specifically target the SDN control plane. The model incorporates a range of ML algorithms, including RF, SVM, KNN, and LSTM, and investigates how feature selection affects model performance. The experimental results demonstrate that the proposed model achieves a high accuracy rate of 99.5% in detecting DDoS attacks. However, the model's performance may be constrained by the limited number of DDoS attack scenarios in the dataset used for evaluation. However, the study's reliance on a limited number of DDoS attack scenarios on the dataset utilized in the study may restrict the ability of the suggested method to detect novel DDoS attack variants.

In [52], an ML-based approach was introduced for the detection of DDoS attacks in SDN environments. The approach employs three ML algorithms NB, SVM, and NN on a privately self-generated dataset to detect DDoS attacks that specifically target the control plane with a single RYU controller. These algorithms are trained on datasets containing network traffic features such as the host time and number of requests. The trained models are then used to classify new network traffic as normal or malicious. The evaluation results reveal that the SVM algorithm achieves an accuracy rate of 80%, whereas the NN algorithm also achieves an accuracy rate of 80%. On the other hand, the NB algorithm has lower accuracy rate of 70%. The proposed approach is implemented via an RYU controller and a Mininet emulator, and the results demonstrate its efficacy in detecting DDoS attacks within SDN networks. However, the dataset size was small.

In [53], a comparative analysis of five ML/DL algorithms for detecting DDoS attacks in SDN was presented. The algorithms under evaluation include SVM, KNN, DT, MLP, and CNN. To assess their performance, the authors employ real-world datasets, specifically the public CICIDS2017 and CICDDoS2019 datasets, to detect volumetric DDoS attacks on both the data and control planes. The results demonstrate that SVM outperforms the other algorithms in terms of accuracy (94.01% for the CICIDS2017 dataset and 95.57% for the CICDDoS2019 dataset), indicating its superiority in accurately detecting DDoS attacks. However, exploring and fine-tuning threshold values for the number of hosts is crucial to improve the approach's effectiveness. Additionally, using MATLAB instead of an SDN environment and the relatively lower predictive accuracy of the CNN model highlight areas that could be improved for enhanced performance.

[74] compared the performance of two open-source SDN controllers, POX and RYU, via Dijkstra's algorithm and custom topologies in the MiniNet emulator. The results show that POX performs better than RYU in terms of different metrics, making it the ideal choice for deployment in diverse scenarios. This study highlights the importance of selecting the right

SDN controller on the basis of network parameters and objectives. However, the study's evaluation is limited to only two controllers and two assessments. The custom topology may not accurately represent real-world networks.

In [17], the authors introduced a novel approach for DDoS flood attack detection and classification via ML algorithms. The study assesses the performance of four widely used ML algorithms, including the QDA, GNB, k-NN, and CART algorithms, on a private self-generated dataset. The dataset is used to detect HTTP, TCP, and UDP flooding attacks on both planes with a single floodlight controller. Among the evaluated algorithms, CART demonstrates superior performance in terms of prediction accuracy, prediction speed, training time, and robustness. Consequently, CART is a promising algorithm for the effective detection and classification of DDoS flooding attacks in SDNs. However, the research dataset is generated from a controlled simulation environment, which may restrict its ability to encompass real-world attack scenarios such as novel or zero-day DDoS flooding attacks.

In [54], a research paper presented an ML-based NIDS specifically designed for SDN. This study focuses on three tree-based ML techniques, namely, DT, RF, and XGBoost, to demonstrate their effectiveness in attack detection. The public NSL-KDD dataset is utilized for training and testing the proposed methods to detect various attacks on the control plane with a single controller. Advanced preprocessing techniques are applied to the dataset to extract the most informative data, resulting in remarkable outcomes compared with those of other systems. By utilizing only five out of the 41 features of the NSL-KDD dataset, a multiclass classification task is performed, detecting the presence of an attack and accurately classifying its type (DDoS, PROBE, R2L, and U2R), achieving an impressive accuracy rate of 95.95%. However, the proposed system is not effective in detecting novel or zero-day attacks.

In [18], a novel approach was introduced for DDoS attack detection in SDN through the application of ML. The approach involves the construction of a customized public DDoS attack SDN dataset and employs the hybrid ML models SVC-RF, LR, KNN, and ANN to classify network traffic as either benign or malicious on multiple planes with a single RYU controller. The model achieves an accuracy of 98.8% by utilizing important features extracted from SDN statistics, and the model achieves the highest level of accuracy in classifying traffic. However, the dataset used in the research was emulated rather than created in real time, which may impact the validity of the results.

In [27], a novel framework for DDoS attack detection and mitigation in SDN was presented. The framework uses an SVM as a prime classifier while employing KPCA and a GA for feature extraction and parameter optimization. To evaluate the efficacy of the proposed model, two public datasets are used: a refined version of the KDD’99 dataset and the NSL-KDD dataset, which cover various types of attacks on the control plane with a single POX controller. The model achieves an accuracy of 98.55% for the KDD Cup 1999 dataset and 92.38% for the NSL-KDD dataset. The results demonstrate the potential of the proposed model in addressing vital security concerns posed by DDoS attacks in SDNs. However, the model performs well in detecting attack traffic in a single-controller environment, and it may encounter challenges in identifying attack traffic in a multicontroller environment and high training times of SVMs.

In [55], a novel method was introduced for the detection of low-rate DDoS attacks that target the data plane of an SDN. The method uses a privately collected dataset with a single controller via the RYU controller. The method extracts four efficient features from flow rules and employs the FM algorithm to combine these features and enhance the accuracy of detection. The experimental findings demonstrate that the proposed method achieves a high level of detection accuracy, reaching 95.80%, surpassing existing detection methods. Furthermore, the method provides fine-grained detection capabilities, making it well suited for identifying low-rate DDoS attacks that are often challenging to detect owing to their covert nature. However, the proposed method has limitations such as potential overhead on the SDN controller, limited effectiveness against sophisticated low-rate DDoS attacks, and scalability issues in large SDN networks with high traffic volumes.

In [75], a novel approach is presented for detecting DDoS attacks in SDN via ML algorithms. The authors implemented and evaluated four different ML algorithms, namely, SVM, MLP, DT, and RF, within a simulated SDN environment. For classification purposes, a comprehensive set of 23 features from the OpenFlow switch flow table was utilized. The results revealed that the RF algorithm achieved the highest accuracy rate of 99.8% and lowest processing time of 0.0001 s in detecting various types of DDoS attacks, including TCP-SYN, UDP flood, flow-table attack, HTTP, ICMP, and bandwidth attacks on both control and data planes, with a single controller using the POX controller for detection. The study also identified key features, including byte count, packet count, and flow duration, that were deemed most significant for the categorization of DDoS attacks. However, while RF has high accuracy, it also has a longer processing time. The implemented algorithms are not designed for real-time detection of new types of attacks.

In [6], a novel technique called the ASVM was proposed for detecting DDoS attacks in SDN networks. The ASVM extends the SVM algorithm to handle multiclass classification and uses the public SDNTrafficDS dataset to reduce training and testing times. The technique uses five features, including volumetric and asymmetric features, to detect UDP and SYN flooding attacks effectively within a multicontroller via an OpenDaylight controller. The proposed system achieves high detection accuracy, approximately 97%, while maintaining fast training and testing times, making it well suited for real-

time DDoS attack detection in SDN networks. The evaluation results further validate the efficacy of the ASVM technique in mitigating DDoS attacks on SDN networks.

In [56], a novel ML-based IDS was proposed for anomaly detection in SDN. The proposed IDS employs a mix of supervised and unsupervised ML techniques, such as KNN, SVM, and ANN, to detect both known and unknown attacks. The system was evaluated via the public KDD Cup1999 dataset to detect various DDoS, flood, and bandwidth attacks on the control plane. To validate the efficacy of the proposed IDS, real-world network traffic data are utilized, and the results demonstrate high accuracy and low FPR in detecting DDoS attacks. However, the study may not effectively detect sophisticated DDoS attacks that can evade detection. Additionally, the hybrid approach used in the study may introduce computational complexity and overhead.

In [57], the paper addresses the issue of DDoS attacks in SDN by presenting a two-layered security solution. The first layer uses Snort for signature-based detection, focusing on known attack patterns. The second layer incorporates ML algorithms, specifically an SVM and a DNN, which are trained on the publicly available KDD Cup1999 dataset to identify abnormal traffic behavior that may indicate DDoS attacks with a single RYU controller. The evaluation results highlight the efficacy of this approach, with the DNN performing better than the SVM, with an accuracy of 92.3%. However, the study is evaluated only in a simulated SDN environment. The training process is time-consuming, and the dataset used is not representative of SDN, which could limit its effectiveness.

In [58], a novel method for detecting DDoS attacks in SDN networks was proposed. The technique extracts six tuple characteristic values from switch flow table information and employs an SVM classification algorithm with a self-generated private dataset to identify TCP-SYN, UDP, and ICMP flood attacks with a single controller via the floodlight controller. The approach achieves a high level of accuracy, up to 95.24%, while requiring only a minimal amount of flow collection. However, the study acknowledges the need for a large amount of training data, and no mitigation measures were implemented in the study.

In [19], two methods for detecting DDoS attacks via SOM in the context of SDN technology were presented. The first approach is a hybrid ML technique called SOM+k-NN, which combines the accuracy of k-NN with the speed of the SOM. This approach was applied to the public DDoS Attack 2007 dataset to detect various types of DDoS attacks, such as SYN, UDP, and ICMP floods, specifically on the control plane with a single controller using the POX controller. The second approach is the SOM distributed center algorithm, which is designed to achieve fast processing without compromising acceptable accuracy. Both proposed algorithms are assessed in a testbed environment, and the experimental results demonstrate their ability to reduce the processing time while maintaining an appropriate detection rate of 99.05%. However, the SOM distributed center algorithm has a higher FPR. Importantly, the experiments were conducted on a limited topology, which may limit the generalizability of the results.

In [59], FADM, a framework specifically designed for detecting and mitigating DDoS flooding attacks in SDN networks, was presented. The FADM utilizes entropy-based feature extraction and SVM classification techniques and uses a self-generated private dataset to achieve accurate and timely detection of SYN, UDP, and ICMP flood attacks in both the data and control planes. The system operates with a single controller via the POX controller. Additionally, it incorporates a white list and traffic migration mechanism for efficient mitigation of these attacks. The FADM is implemented as a lightweight and protocol-independent prototype system and is evaluated via real DDoS attack traffic. The evaluation results demonstrate the effectiveness of the FADM in detecting and mitigating multiple DDoS flooding attacks at an early stage, thereby enabling the network to recover rapidly. However, the proposed method has vulnerabilities against sophisticated attacks that can evade detection.

Table 9 summarizes studies that have utilized ML techniques for cyber-attack detection in SDN.

TABLE IX. MACHINE LEARNING-BASED DETECTION

Ref.\ Year	Method	Dataset Used\ Private, Public	Type of Attacks	Target Plane	No. of Controller	Controller Type	Scope	Results
[50] 2024	1D-CNN, DT.	Self-generated dataset\private	DDoS attacks	Control and data plane	Multicontroller.	--	Detection	ACC 97.80%, PREC 97.80%, Recall 97.80%, F1-score 97.79%, SPEC 99.35%
[5] 2024	AE, BGRU, SVM, KNN, DT, NB, RF.	NSL-KDD dataset\ public, and Self-generated dataset\private	DDoS, probe, R2L, U2R.	Control and Data plane	Single controller.	POX	Detection and mitigation	ACC 99.87%, PREC 98.99%, Recall 99.48%, F1-score 99.18%

Ref.\ Year	Method	Dataset Used\ Private, Public	Type of Attacks	Target Plane	No. of Controller	Controller Type	Scope	Results
[73] 2023	RF, AdaBoost, KNN, NB, DT, LR, Deep CNN, LSTM.	InSDN dataset\ public	DDoS, Probe, DoS, Bruteforce, Exploitation, R2L.	Control and data plane	Single controller.	--	Detection	ACC 99.85%, PREC 98%, Recall 89%, F1-score 91%
[51] 2023	LR, SVM, KNN, RF, LSTM.	DDoS attack SDN dataset\ public	DDoS attacks	Control plane	--	--	Detection	ACC 99.5%, PREC 99.5%, Recall 99.4%, F1-score 99.4%
[52] 2023	NB, SVM, NN.	Self-generated dataset\ private	DDoS attacks	Control plane	Single controller.	RYU	Detection	SVM ACC 80%, NN ACC 80%, NB ACC 70%
[53] 2023	SVM, KNN, DT, MLP, CNN.	CICIDS2017 and CICDDoS2019 dataset\ public	Volumetric DDoS attacks	Data and control plane	--	--	Detection	CICIDS2017 ACC 94.01%, F1-score 97%, MCC 94% CICDDoS2019 ACC 95.57%, F1-score 97%, MCC 94%
[74] 2023	Dijkstra's Algorithm for optimal path, Iperf for throughput measurement.	Not mention	--	Data Plane	Multicontroller	POX and RYU	Detection	In this approach, POX performs better than RYU in terms of various metrics.
[17] 2021	QDA, GNB, KNN, CART, RF-SVC.	Self-generated dataset\ private	HTTP, TCP, and UDP flooding attacks.	Control and Data plane	Single controller.	Floodlight	Detection	This approach proves CART to be a promising algorithm for the effective detection and classification of DDoS flooding attacks in SDNs.
[54] 2021	DT, RF, XGBoost.	NSL-KDD dataset\ public	DoS, U2R, R2L, Probe attacks.	Control plane	Single controller.	--	Detection and mitigation	ACC 95.95%, PREC 92%, Recall 98%, F1-score 95.55%
[18] 2021	LR, SVC, KNN, RF, ANN, Hybrid Model (SVC-RF).	DDoS attack SDN dataset\ public	Application plane attack, Control plane attack, communication link attack, Table-overflow attack	Application, control, communication link, data plane.	Single controller.	RYU	Detection	ACC 98.8%, PREC 98.27%, F1-score 97.65%, FAR 0.02%, SPEC 98.18%, DR 97.91%.
[27] 2020	SVM, GA.	KDD Cup 1999 and NSL-KDD dataset\ public	Volumetric attacks, protocol-exploitation attacks, application-layer attacks.	Control plane	--	POX	Detection and mitigation	KDD Cup 1999 ACC 98.55% NSL-KDD ACC 92.38%
[55] 2020	FM	Collected dataset\ private	Low-rate DDoS attacks	Data plane	Single controller.	RYU	Detection and mitigation	ACC 95.8%, PREC 95%, Recall 94.6%, AUC 93.8%
[75] 2020	SVM, MLP, DT, RF	Not mention	TCP SYN Flood, Flow-table attack, HTTP, ICMP, UDP Flooding, Bandwidth attack, HTTP Flooding	Control and data plane	Single controller.	POX	Detection	ACC 99.8%
[6] 2019	ASVM	SDNTrafficDS dataset\ public	UDP flooding, SYN flooding.	Control plane	Multicontroller.	OpenDaylight	Detection	ACC 97%, FAR, 0.03%, DR 97%
[56] 2019	KNN, ANN, SVM	KDD Cup 1999 dataset\ public	DDoS attacks, flood attacks and bandwidth attacks.	Control plane	--	--	Detection	This approach demonstrates high accuracy and low FPR in detecting DDoS attacks.
[57] 2018	SVM, DNN, Snort: for signature-based detection.	KDD Cup 1999 dataset\ public	DDoS attacks	Control plane	Single controller.	RYU	Detection	ACC 92.3%, PREC 90%, Recall 100%
[58] 2018	SVM	Self-generated dataset\ private	TCP SYN flood, UDP flood, and ICMP flood.	Control plane.	Single controller.	Floodlight.	Detection	ACC 95.24%, FAR 2.77%

Ref.\ Year	Method	Dataset Used\ Private, Public	Type of Attacks	Target Plane	No. of Controller	Controller Type	Scope	Results
[19] 2018	SOM, KNN	DDoS Attack 2007 dataset\ public	DDoS attacks, SYN flood, UDP flood, and ICMP flood.	Control plane	Single controller.	POX	Detection and mitigation	FPR 2.74%, DR 99.05%
[59] 2017	SVM, Shannon Entropy (feature selection), White-list	Self-generated dataset\ private	SYN flood, UDP flood, ICMP flood.	Data and Control plane	Single controller.	POX	Detection and mitigation	This approach is effective in detecting and mitigating multiple DDoS flooding attacks at an early stage.

7.3 Cyber attack detection using deep learning techniques

DL techniques offer a promising approach for detecting cyber-attacks in SDN. This section analyses various DL algorithms to detect cyber-attacks, as shown by the authors in [37], and the authors present DLADSC, a DL-based approach that incorporates IGR and chi-square-based cross-feature selection to identify informative features for DDoS detection. The RNN model is subsequently trained on a public DDoS attack SDN dataset via these selected features to detect UDP, TCP, and ICMP attacks on the control plane, with a single controller using the RYU controller used to simulate DDOS attacks in the SDN environment. The proposed approach achieves high levels of detection accuracy (94.18%), precision (92.14%), and F1-measure (94.27%) while maintaining a low false positive rate (8.11%). DLADSC effectively addresses the limitations of existing approaches and provides robust solutions for detecting DDoS attacks on SDN controllers. However, it was not effective against novel or zero-day DDoS attacks and was computationally expensive during training and deployment.

[33] introduces a novel hybrid DL approach designed to detect and defend against DDoS attacks in SDNs. This approach combines three algorithms, namely, the 1D-CNN, GRU, and DNN, to achieve high detection accuracy for both volumetric and low-rate DDoS attacks on both the control and data planes with a single controller. The effectiveness of the suggested approach is evaluated via two different datasets: a public CICDDoS2019 dataset and a self-generated private dataset. The evaluation results demonstrate remarkable accuracy rates of 99.81% and 99.88%, respectively, demonstrating the efficacy of the hybrid DL approach in detecting and defending against DDoS attacks in SDNs. However, it is susceptible to adversarial attacks and requires a large amount of labelled data for training.

In [34], the authors propose an adversarial approach for detecting and mitigating DDoS attacks in SDN environments. This approach combines DBN-LSTM with a GAN. The objective of this approach is to improve the system's resilience against adversarial attacks and enhance the efficacy of the feature extraction process. Experiments are conducted on the publicly available CICDDoS2019 dataset, with an achieved accuracy of 96.55% for detection. However, it is time-consuming and not effective against all types of adversarial attacks.

In their study [35], the authors investigated the effects of balancing strategies and imbalanced learning approaches on intrusion data in SDN. They proposed custom DL architectures using GANs and a Siamese NN for generative modelling and similarity-based intrusion detection. To evaluate performance, they benchmark results from a classification using ROS, SMOTE, GANs, weighted RF, and Siamese-based one-shot learning on the InSDN public dataset to detect various types of attacks, including DDoS, Probe, DoS, Bruteforce, and Exploitation (R2L), on the control plane. The results indicate that the RF outperforms the DL models in classifying minority class instances. Additionally, the authors observed that commonly used balancing techniques, such as ROS and SMOTE, significantly reduce the FPR but increase the FNR when minority classes are classified. However, these studies do not address the problem of class overlap, which can affect classification performance.

In their paper [22], the authors present a DL-based approach for the detection and mitigation of botnet attacks in SDNs. The main focus is on feature selection and the utilization of DL models to classify both normal and attack flows. The proposed approach is evaluated via a custom self-generated private dataset to detect Botnet and DDoS attacks on both the control and data planes, with a single controller using the POX controller to simulate DDOS attacks in the SDN environment. yielding promising results. Specifically, the CNN achieves a 99% detection rate for normal flows and a 97% detection rate for attack flows for detection and mitigation. Additionally, this paper presents a mitigation strategy that relies on graph theory and dynamic flow deletion. However, this research focuses on botnet-based flooding DDoS attacks in SDN environments and restricts its relevance to other types of attacks.

In [60], the authors introduced a novel approach for defending against packet injection attacks in SDNs. The approach covers two primary components: detection and mitigation modules. The detection module employs a GCN on a private self-generated dataset and a public CICIDS2017 dataset to detect high-rate, low-rate, Discontinuous, DDoS, and PortScan

attacks on the control plane, with a single controller using the RYU controller to simulate DDOS attacks in the SDN environment. The detection module aims to identify malicious hosts by analysing their traffic patterns. The mitigation module subsequently installs blocking rules at switches to prevent malicious traffic from accessing the network. The achieved accuracy was 99% for both detection and mitigation. However, there is a concern regarding the computational overhead of DL models in real-time network monitoring and the necessity for ongoing updates to address evolving attack techniques.

In [61], the authors introduced a DDoS attack detection model specifically designed for SDNs. The model utilizes optimized DNNs trained via the SIMFO algorithm on public datasets such as CIC-DDoS2019 and DDOS attack SDN datasets to detect DDoS attacks on the control plane with a single controller via the RYU controller for simulating DDOS attacks in the SDN environment. By analysing flow-based and statistical features, the model accurately identifies the existence and type of attacks. The model achieved an accuracy of 94.83% for detection and mitigation. Once an attack is detected, a mitigation process based on the baiting approach is implemented to remove the attacker's node from the network. SDN Defend is a lightweight online system designed for detecting and mitigating DDoS attacks in SDN networks [1]. The system incorporates a CNN-ELM intrusion detection module along with IP traceback-based mitigation. The system is evaluated on public datasets such as CICIDS-2017 and InSDN to detect various attacks, including Packet_in flooding, CrossPath, and flow table overflow attacks on both the control and data planes with multiple controllers via the RYU controller for simulating attacks in the SDN environment. By eliminating abnormal flows from the source, SDN Defend achieves high detection accuracy and effectively mitigates DDoS attacks. The accuracy was 99.86% for the CICIDS-2017 dataset and 99.91% for the InSDN dataset. This contribution to the field of SDN security offers a lightweight and efficient defense mechanism against DDoS attacks.

In their research [62], IDS was proposed on the basis of DL ensemble methods. The IDS utilizes CNNs, RNNs, and DNNs, along with feature selection techniques, to identify DDoS attacks effectively. The ensemble model is trained on the public CICIDS2017 dataset to detect various types of DDoS attacks, such as UDP, SYN, and HTTP floods, on the control plane, achieving an impressive detection accuracy of 99.05%. However, the model is computationally expensive and requires a large amount of data for training. In [63], the authors proposed a two-level DDoS attack detection approach in SDN networks. This approach involves using entropy to detect faked switch ports and employing a CNN as a classifier to improve accuracy and efficiency while reducing training costs. The approach is evaluated on the public CICIDS2017 dataset to detect DDoS attacks on the control plane with a single controller using the POX controller for simulating attacks in the SDN environment. The primary objective of this method was to effectively combat DDoS attacks. As a result, the presented strategy achieved the highest accuracy of 98.79% for the CNN model. However, the accuracies of the information entropy method and the two-level method were relatively lower, reaching 92.37% and 96.97%, respectively. However, it requires additional software installation on the controller and is not suitable for all types of DDoS attacks.

In [26], the authors present a hybrid neural network structure called DDoSTC, which combines efficient and scalable transformers with a CNN for detecting DDoS attacks in SDN. The proposed method is evaluated on a public dataset, CICDDoS2019. The experimental results demonstrate that DDoSTC achieves an average accuracy of 99.70%, surpassing the existing optimal model by 2.52%, making it a more effective approach for DDoS attack detection. However, lightweight virtualization has limitations in terms of security and isolation. The performance of lightweight virtualization can be limited for complex test scenarios.

In [13], the authors proposed a DNN solution for real-time detection of DDoS attacks in SDNs. The model is trained on the public CICIDS 2017 dataset, specifically on the control plane with a single controller, achieving an impressive accuracy of 97.67% in detecting DDoS attacks. A highly effective solution provides strong protection for SDN environments against these malicious attacks. Furthermore, the proposed model is comparable with other state-of-the-art models, highlighting its superior performance. However, the study focuses on DDoS attack detection in SDN environments, may not be generalizable to other types of attacks or network architectures, and needs an SDN-based dataset.

In [76], the authors propose and assess the effectiveness of DL algorithms in addressing the vulnerability of SDN controllers. They introduced promising solutions for detecting DoS attacks in SDN via three DL algorithms: RNN, LSTM, and GRU. These algorithms are assessed via the public InSDN dataset to detect DDoS, DoS, probe, botnet, and web attacks; password brute-force attacks; and the exploitation of user-to-root attacks on the control plane, which includes real-world attack scenarios. The evaluation of the DL algorithms on the InSDN dataset demonstrated their efficacy in detecting DoS attacks, with all three algorithms achieving high accuracy levels. Specifically, the LSTM achieves the highest accuracy of 99.99%, followed by the GRU with 99.98% and the RNN with 99.97%.

In [64], the authors utilized several DL techniques, including CNN, LSTM, SVM-SOM, and SAE-MLP, to detect DDoS attacks in SDN environments. The study focused on detecting DDoS attacks that involve flooding from multiple sources (such as TCP-SYN, UDP, and ICMP attacks) on the data plane with multiple controllers. The DL methods were applied to a public DDoS attack SDN dataset to classify incoming network traffic as either benign or malicious on the basis of the dataset's features. They developed a stacked autoencoder multilayer perceptron (SAE-MLP) model and focused on both

the detection and mitigation of DDoS attacks. The results of the study demonstrated an impressive accuracy of 99.97% in detecting and mitigating DDoS attacks via the SAE-MLP model.

In [65], the authors presented a novel DL framework for detecting DDoS attacks in SDNs. The framework uses Ensemble CNN, Ensemble RNN, Ensemble LSTM, and Hybrid RL models to enhance detection accuracy, specifically for flow-based data. The proposed approach is assessed via the public CICIDS2017 dataset and a state-of-the-art flow-based SDN private dataset. The evaluation focuses on detecting flooding from multiple sources, including TCP-SYN, UDP, and ICMP attacks on the control plane via the floodlight controller. The results achieve an impressive detection accuracy of 99.45%, highlighting the effectiveness of the DL framework in identifying and mitigating DDoS attacks in SDN environments. This framework is scalable, cost-effective, and outperforms existing detection approaches in terms of performance. However, the study evaluated the proposed framework via a single dataset, and the high computational complexity of DL models may hinder their deployment in resource-constrained environments.

In [66], the authors propose the LSTM-FUZZY system, which is a modular system developed for anomaly detection and mitigation in SDN environments. The system comprises three phases: characterization, anomaly detection, and mitigation. The characterization phase uses LSTM to predict the normal behavior of network traffic. The anomaly detection phase employs fuzzy logic and the Bienaymé Chebyshev inequality to identify anomalies. The mitigation phase applies countermeasures to minimize the impact of detected attacks. The system is assessed via the public CICDDoS2019 dataset to detect various DDoS attacks to detect NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebD-DoS (ARME), SYN, TFTP, and portscan attacks on the application plane with multiple controllers via the floodlight controller to simulate attacks in the SDN environment. It is specifically designed to aid in network management, detect and mitigate attacks in near real time, and learn the network's normal behavior, thus achieving an accuracy of 93.13%. However, the approach is not effective against zero-day DDoS attacks and requires a trained LSTM model for each flow attribute.

In [77], the authors presented DDoSNet, a DL model designed for detecting DDoS attacks in SDN environments. DDoSNet combines AE and RNN deep neural networks. The model is evaluated via a recent public CICDDoS2019 dataset, which offers a diverse range of attacks to detect DDoS attacks, including volumetric attacks such as ICMP, UDP, and TCP-SYN floods, as well as application layer attacks such as HTTP and DNS on the data plane, and fills gaps in existing datasets. The detection achieved the highest accuracy rate of 99%. However, it was not evaluated on real-world SDN deployments.

In [20], the authors introduce DEEPGUARD, a creative anomaly detection framework specifically designed for SDN-based networks. DEEPGUARD uses a deep reinforcement learning technique known as DDQN to learn traffic flow matching strategies for detecting DDoS attacks on the data plane with a single controller using the ONOS controller used to simulate attacks in the SDN environment. These strategies aim to maximize traffic flow granularity while safeguarding the SDN data plane from overload. The framework then implements the learned optimal traffic flow matching control policy to obtain the most valuable traffic information for real-time anomaly detection. The effectiveness of DEEPGUARD is thoroughly evaluated through extensive experiments, which demonstrate significant performance improvements over existing traffic flow matching mechanisms. However, the approach's inflexible flow matching strategies, potential performance degradation with dynamic traffic flows, and inadequate adjustment of flow granularity affect anomaly detection.

In [67], the authors presented a collaborative anomaly detection scheme (CAD) designed specifically for distributed SDN environments. This paper addresses scalability and reliability issues that arise in centralized SDN, which is vulnerable to cyberattacks. In distributed SDN, each controller manages a small portion of the network and has limited information. The proposed CAD scheme enables multiple distributed SDN controllers to collectively train a global detection model for the whole network without directly sharing raw data. This approach resolves biased flow problems and improves detection accuracy. CAD leverages the MD-GAN and EGBAD techniques to train the discriminator, generator, and encoder on each controller. The effectiveness of CAD is evaluated via the use of a public KDD99 dataset to detect various types of attacks, including DoS, U2R, R2L, and probing attacks on the control plane with multiple controllers. The evaluation results demonstrate the superiority of CAD over individual and centralized methods, achieving an accuracy rate of 98.28%, lower computational cost, and improved detection rates for various types of attacks.

In [68], the authors explored the use of a DL-based mechanism for detecting slow DDoS attacks in an SDN-based network. They proposed a hybrid model that combines a CNN and LSTM called CNN-LSTM to identify slow DDoS attacks effectively. The performance of their method is assessed via custom datasets, which yield impressive results. Compared with other DL models such as the MLP and standard ML models such as the 1-class SVM, the hybrid CNN-LSTM model exhibits superior performance. In the experimental scenarios, privately generated synthetic traffic flows are used to detect slow DDoS attacks on the data plane with the ONOS controller, this approach achieves the highest accuracy rate of 99.99%. However, the datasets used are synthetically generated, so they do not represent real-world traffic patterns and are not able to be generalized well to unseen data.

In [31], the authors developed DL-based IDPS to mitigate brute force and DDoS attacks that target SSHs in SDN environments. The DL-IDPS uses the public ISCX2012 dataset to detect SSH brute-force attacks and DDoS attacks on the data plane with a single controller via the RYU controller. Additionally, the authors applied four models, namely, SAE, LSTM, MLP, and CNN, to compare the performance of their approach against these existing models. The authors claimed that their proposed model achieved an accuracy of 99.3% when the MLP was employed in DL-IDPS. However, these studies did not investigate the influence of different network configurations and traffic patterns on system performance.

In [69], the authors introduced LionIDS, a novel IDS designed specifically for detecting DDoS and DoS attacks in the SDN control plane with a single controller. LionIDS combines the LOA for feature selection with a CNN for classification. This integration results in a high accuracy rate of 96% and a low FAR of 4%. To assess the performance of LionIDS, the authors utilize the public NSL-KDD and self-generated private datasets collected in real time via Mininet. However, the paper does not evaluate the performance of the proposed IDS on a large-scale network or with other state-of-the-art IDSs.

In [28], the authors proposed an ensemble solution for efficiently detecting DDoS attacks in the SDN control plane via a DL model that is based on a CNN. The authors utilize the public CICIDS2017 dataset (formerly known as the ISCX2012 dataset) in their work and claim that the empirical results obtained from their study demonstrate the highest detection accuracy of 99.48% with low computation time. However, the proposed solution is evaluated on a single dataset, and the impact of different network configurations on the performance of the solution has not been explored.

In [25], the authors presented a method for detecting botnets in SDNs via MLP, C4.5, DT, SVM, and Bayesian networks. The method is applied to public CTU-13, ISOT, and self-generated private datasets to detect botnet attacks, including centralized (IRC, HTTP) and decentralized (P2P) botnet structures on the application plane with a single controller using the RYU controller. The proposed method uses the MLP model to classify traffic as either malicious or normal on the basis of extracted features from each session. The proposed method achieves a detection accuracy of 99.2%, surpassing the performance of existing methods. Furthermore, the results demonstrate the use of SDN to isolate infected machines through the execution of firewalls and VLANs. However, the proposed method was not tested on real-world bot-infected machines, only on precaptured datasets. The system relies solely on session-based features and could benefit from incorporating time interval-based features for improved detection of internal botnet propagation. However, the proposed method was not tested on real-world bot-infected machines, only on precaptured datasets. The system relies solely on session-based features and could benefit from incorporating time interval-based features for improved detection of internal botnet propagation.

In their research [14], the authors propose a DL-based system for detecting and defending against DDoS attacks in OpenFlow-based SDN. The system uses a DL model to learn patterns from sequences of network traffic and trace historical network attack activities. The model integrates RNN, LSTM, and CNN to enhance its detection capabilities. The proposed method is assessed via the public ISCX2012 dataset to detect flooding from various sources, including TCP-SYN, UDP, ICMP, HTTP, and DNS, on the network's data plane. The results demonstrate its effectiveness, with a verification accuracy of 99% for training data and 98% for test data, respectively, in identifying DDoS attacks. However, it was not effective in handling low-rate attacks.

Table 10 summarizes studies that have utilized DL techniques for cyber-attack detection in SDN.

TABLE X. DEEP LEARNING-BASED DETECTION

Ref.\ Year	Method	Dataset Used\ Private, Public	Type of Attacks	Target plane	No. of Controller	Controller Type	Scope	Results
[37] 2023	RNN, IGR, Chi-square (feature selection).	DDoS attack SDN dataset\public	UDP, TCP, and ICMP attacks.	Control plane	Single controller	RYU	Detection and mitigation	ACC 94.18%, PREC 92.14%, F1-measure 94.27%, low FPR 8.11%.
[33] 2023	1D-CNN, GRU, DNN.	CICDDoS2019 dataset\ public and self-generated dataset\ private	Low-rate DDoS attacks, and Volumetric DDoS attacks.	Control and Data plane	Single controller	RYU	Detection	ACC 99.81% and 99.88% for the two datasets used PREC 99.96%, Recall 99.9%, F1-score 99.93%, ROC 99.9%
[34] 2023	GAN, DBN, LSTM	CICDDoS2019 dataset\ public	DDoS attacks	Control plane	Single controller	--	Detection	ACC 96.55%, PREC 96.44%, Recall 98.53%,
[35] 2023	ROS, SMOTE, GANs, RF, Siamese-based one-shot learning, Siamese NN.	InSDN dataset\ public	DDoS, Probe, DoS, Bruteforce, Exploitation (R2L).	Control plane	--	--	Detection	This approach indicates that RF outperforms DL models.

Ref.\ Year	Method	Dataset Used\ Private, Public	Type of Attacks	Target plane	No. of Controller	Controller Type	Scope	Results
[22] 2023	RNN, CNN, MLP, LSTM, DNN, graph theory, dynamic flow deletion.	Self-generated dataset\ private	Botnet, DDoS attacks.	Control and Data plane	Single controller	POX	Detection and mitigation	CNN ACC 99%, PREC 99.37%, F1-score 98.61%, FPR 0.53%, TPR 97.87%.
[60] 2023	GCNs	CICIDS2017 dataset\ public, and Self-generated dataset\ private	High-rate attacks, Low-rate attacks, Discontinuous attacks, DDoS, and PortScan attacks.	Control plane	Single controller	RYU	Detection and mitigation	ACC 99%.
[61] 2023	DNN, SIMFO.	CIC-DDoS2019 dataset, and DDOS attack SDN dataset\ public	DDoS attacks.	Control plane	Single controller	RYU	Detection and mitigation	ACC 94.83%, PREC 91.99%, Recall 92%, F1-score 90.18%, FPR 0.44%, FNR 0.90%, MCC 85.84%, SPEC 93.46%
[1] 2022	CNN-ELM	CICIDS2017 dataset, and InSDN dataset\ public	Packet_in flooding, CrossPath attacks, flow table overflow attacks.	Control and Data plane	Multicontroller.	RYU	Detection and mitigation	InSDN dataset: ACC 99.91%, PREC 99.92%, Recall 99.89%, F1-score 99.91%. CICIDS-2017: ACC 99.86%, PREC 99.89%, Recall 99.78%, F1-score 99.84%
[62] 2022	CNN, LSTM, GRU, voting ensemble	CICDDoS2019 dataset\ public	UDP floods, SYN floods, and HTTP floods.	Control plane	--	--	Detection	Ensemble ACC 99.05%, Recall 99.40%, ROC 99.30%,
[63] 2022	CNN, Information entropy analysis	CICIDS2017 dataset\ public	DDoS attacks.	Control plane	Single controller	POX	Detection	Entropy ACC 92.37%, PREC 92.95%, Recall 91.13%, F1-score 92.03%, DT 2.15%. CNN ACC 98.79%, PREC 98.67%, Recall 98.93%, F1-score 98.80%, DT 10.23%
[26] 2021	CNN, GRU, CNN, B-GRU, RNN, LSTM+GRU, LSTM.	CICDDoS2019 dataset\ public	DDoS attacks.	Control plane	--	--	Detection	ACC 99.70%, PREC 99.98%, Recall 99.70%, F1-score 99.84%, AUC 99.95%
[13] 2021	DNN.	CICIDS2017 dataset\ public	DDoS attacks.	Control plane	Single controller	--	Detection	ACC 96.67%., PREC 97.21%., Recall 97.29%, F1-score 97.25%.
[76] 2021	RNN, LSTM, GRU.	InSDN dataset\ public	DDoS, DoS, probe, botnet, web attacks, password brute-force attacks, and exploitation user-to-root attacks.	Control plane	--	--	Detection	LSTM ACC 99.99%, PREC 99.91%, Recall 99.97%, F1-score 99.94% GRU ACC 99.98%, and RNN ACC 99.97%
[64] 2021	CNN, LSTM, SVM-SOM, SAE-MLP	DDOS attack SDN dataset\ public	DDoS attacks involving flooding from multiple sources (TCP-SYN, UDP, ICMP) attack.	Data plane	Multicontroller	--	Detection and mitigation	SAE-MLP ACC 99.97%, PREC 99.96%, Recall 99.77%, F1-score 99.87%, FPR 0.05%, FNR 0.22%

Ref.\ Year	Method	Dataset Used\ Private, Public	Type of Attacks	Target plane	No. of Controller	Controller Type	Scope	Results
[65] 2020	Ensemble CNN, Ensemble RNN, Ensemble LSTM, Hybrid RL.	CICIDS2017 dataset\ public, and state-of-the-art Flow-based SDN dataset\ private	HTTP flood, SYN flood, UDP flood, and ICMP flood.	Control plane	--	Floodlight	Detection	ACC 99.45%, PREC 99.57%, Recall 99.64%, F1-score 99.61%, CPU usage 6.02%.
[66] 2020	LSTM, Fuzzy logic Shannon Entropy.	CICDDoS2019 dataset\ public	NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebD-DoS (ARME), SYN, TFTP Portscan attacks	Application plane	Multicontroller	Floodlight	Detection and mitigation	ACC 93.13%, PREC 97.89%, FPR 2.2%
[77] 2020	RNN, AE, SoftMax regression.	CICDDoS2019 dataset\ public	DDoS attacks. Volumetric attacks (e.g., ICMP, UDP, TCP-SYN flood) Application layer attacks (e.g., HTTP, DNS)	Data plane	--	--	Detection	ACC 99%, PREC 99%, Recall 99%, F1-score 99%
[20] 2020	DDQN, SVM, fine-grained traffic flow monitoring mechanism	Not mention	DDoS attacks	Data plane	Single controller	ONOS	Detection and mitigation	This approach demonstrates significant performance improvements compared to existing traffic flow matching mechanisms.
[67] 2020	MD-GAN	KDD Cup 1999 dataset\ public	DoS, U2R, R2L, Probing.	Control plane	Multicontroller	--	Detection	ACC 98.28%, PREC 94.91%, Recall 96.42%, F1-score 95.66%
[68] 2020	(CNN-LSTM), MLP, 1-Class SVM.	Self-generated dataset\ private	Slow DDoS attacks.	Data plane	--	ONOS	Detection	ACC 99.99%, PREC 99.98%, Recall 100.00%, SPEC 99.99%, F1-score 99.99%
[31] 2020	MLP, CNN, LSTM, SAE.	ISCX2012 dataset\ public	SSH brute-force attacks, DDoS attacks.	Data plane	Single controller	RYU	Detection and prevention	ACC 99.3%
[69] 2019	CNN, LOA: A meta-heuristic algorithm.	NSL-KDD dataset\ public and Self-generated dataset\ private	DDoS and DoS attacks.	Control plane	Single controller.	--.	Detection.	ACC 96%, FPR 4%
[28] 2019	CNN, Ensemble learning	CICIDS2017 dataset\ public	DDoS attacks	Control plane	--	--	Detection.	ACC 99.48%, PREC 99.51%, Recall 99.74%, F1-score 99.63%
[25] 2019	MLP, C4.5, DT, SVM, and Bayesian networks.	CTU-13 dataset, ISOT dataset\ public, and Self-generated dataset\ private	botnet including both centralized (IRC, HTTP) and decentralized (P2P) botnet structures.	Application plane	Single controller.	RYU	Detection.	ACC 99.2%
[14] 2018	CNN, RNN, LSTM, GRU	ISCX2012 dataset\ public	UDP Flood, HTTP Flood, SYN Flood, ICMP Flood, and DNS Flood.	Data plane	--	--	Detection and mitigation	ACC 98%

7.4 Cyber attack detection via the federate learning technique

Recent security research has employed FL models to improve security measures and handle distributed network traffic, with a precise focus on emerging attack types, as shown by the authors in [70], and the authors introduce a novel approach using FL for detecting DDoS attacks in SDN environments. FL enables the raining of models on distributed devices without the need to send data to a central server, ensuring user privacy and reducing resource consumption. The proposed FL models utilized DNN, CNN, and LSTM models to achieve high accuracy in detecting DDoS attacks, with the LSTM model

performing best among the evaluated techniques on the public CICIDS2019 dataset to detect flood (TCP-SYN, UDP, DNS) attacks on the data plane with a single controller, achieving an impressive accuracy of 99.99%. However, it requires multiple clients to participate in the FL process, the training time can be longer than that of centralized models, and security and privacy concerns need to be addressed.

In [78], the authors propose FL for adaptive DDoS attack detection (FLAD), a novel FL approach that employs federated learning DDoS (FLDDoS), and a lightweight residual network (LwResNet) specifically designed for detecting DDoS attacks. FLAD addresses limitations found in existing algorithms, such as Federated Averaging (FEDAVG), by dynamically modifying the workload assigned to each client on the basis of their local validation accuracy. This adjustment improves both the convergence time and accuracy without the need for test data at the server. Moreover, FLAD ensures privacy by allowing clients to collaborate on DDoS attack detection without sharing their training or test data. The effectiveness of FLAD is demonstrated through significant enhancements in convergence time and accuracy compared with existing FL algorithms. The evaluation is accomplished via the use of the public CIC-DDoS2019 dataset to detect UDP, SYN, and HTTP flood attacks on the control plane with a single controller.

In [29], the authors focused on addressing security challenges in FL systems by integrating SDN with FL. The main objective is to create a secure and efficient distributed learning environment while ensuring the reliability of the learned models and the protective privacy of the local data. To achieve this goal, an architecture that uses SDN to detect attacks on FL is proposed. ML algorithms, including RF, DT, and KNN, are employed for model training. The public N-BaIoT dataset is utilized for IDSs in SDN networks to detect Mirai and Bashlite attacks on both the control and data planes with a single controller. The detection achieved high accuracy, with an RF of 99.6%, a DT of 99.8%, and a K-NN of 99.3%. This paper emphasizes the importance of integrating cybersecurity measures into FL and SDN to safeguard sensitive data and prevent unauthorized access. However, communication overhead and privacy concerns may impact performance.

In [24], the authors presented a novel intrusion detection system based on FL specifically designed for SDNs. The proposed model addresses challenges related to data privacy and attack detection. By leveraging FL, the system trains the global LSTM model on various public datasets, such as UNSW-NB15, NF-UQ-NIDS-v2, and CICIDS2017, to detect attacks on both the control and data planes with a single controller via the RYU controller. This approach ensures data privacy while allowing collaborative learning. The evaluation results demonstrate the system's effectiveness in achieving high accuracy in anomaly detection for the datasets, ranging from 97% to 99%. However, it requires a large amount of data for training and is not suitable for real-time applications.

In [71], the authors investigated the effectiveness of FL with a GAN for intrusion detection in SDN environments. The study specifically focuses on the difficult task of selecting threat-specific features in non-IID data. The authors utilize a public InSDN dataset to detect various types of attacks, including DoS, DDoS, brute force, probes, web, and botnet attacks, on the control plane with multiple controllers. After data preprocessing, PCA is employed to analyse the effect of non-IID data on the importance of features. The simulation results provide detailed insights, revealing significant variations in the importance of features for non-IID data, both in terms of quantity and distribution across different threat types. However, the study does not address data imbalance or overfitting in non-IID scenarios and does not evaluate the performance of the proposed FL-based IDS on real-world networks.

In [32], the authors presented a novel framework named MiTFed that combines FL, secure multiparty computation (SMPC), and blockchain technologies to enhance security measures in SDN. The framework is applied to the public NSL-KDD dataset to detect DDoS, probe, U2R, and R2L attacks on both the control and data planes with multicontrollers in SDN environments. MiTFed aims to address emerging security threats on a large scale and is designed as a distributed, efficient, trustworthy, flexible, and privacy-preserving framework specifically for managing DDoS collaboration processes across multiple SDN domains. The experimental results demonstrate that MiTFed outperforms existing centralized ML and DL models in terms of accuracy (89%) and F1 score (89%) for both binary and multiclass classification tasks, all while ensuring the protection of collaborating parties. However, it requires a trusted third party for secure aggregation, limited scalability for large-scale deployments, and high computational overhead.

In [79], the authors introduced a novel solution called FEAR (federated deep reinforcement learning-based adaptive cyber-attack reaction) to effectively defend against cyber-attacks in distributed SDN scenarios. The Federated DQN and the Q-learning algorithm were used to detect DoS, TCP-SYN flood, and link layer flood attacks on both the control and data planes with multiple controllers via the ONOS controller. The FEAR framework comprises an aggregation server and multiple FEAR agents, each of which is equipped with local datasets. Each FEAR agent performs its local update and transmits it to the aggregation server. The aggregation server then updates the global model via received local updates from all FEAR agents. The updated global model is subsequently distributed to all the FEAR agents. To evaluate the FEAR framework, researchers employ the MaxiNet tool to emulate a distributed SDN scenario that includes six SDN networks. The performance of the FEAR framework is compared with that of existing solutions, namely, CARS and GATE, in terms of attack defense and the ratio of QoS-violated traffic flows. The results demonstrate that the FEAR framework effectively

safeguards victims from malicious packets and significantly decreases the ratio of QoS-violated traffic flows compared with both the CARS and GATE solutions. However, the study's performance evaluation is limited to DoS attacks.

In [72], the authors presented a novel approach for cyber threat hunting in SDN-enabled networks by leveraging the FL (FedPlus), LSTM, GRU, and CNN methods on the public NF-UQ-NIDS dataset to detect APTs on the control plane with a single controller via the ONOS controller. This approach enables collaborative parties to individually train DL-based IDSs via their data, thus preserving data privacy. To provide context for threat hunting, the IDS system is further enriched with additional data from the SDN network, including flow data and network topology. The proposed approach demonstrates high accuracy in anomaly detection and improves the detection rate for unknown threats. However, careful selection of enrichment data is needed to avoid overwhelming or insufficient information, high computational and communication overhead associated with FL, limited real-time capabilities, and OpenCTI systems not yet implemented.

In [80], the authors introduced a novel system designed to identify threats in networks via SDN technology. The system incorporates methods known as FL with differential privacy (DP), homomorphic encryption (HE), and LSTM methods on the public CIC-TON-IOT 2018 dataset and a self-generated private dataset to detect DoS, DDoS, botnet, and XSS attacks on the data plane with a single controller via the ONOS controller. Multiple organizations can collaborate on training models without compromising the protection of their respective data. This approach improves the accuracy of threat detection while maintaining information security. Additionally, the system leverages big data tools to efficiently handle the substantial volume of information involved. As a result, this approach presents a guaranteed solution for ensuring network safety while protecting data privacy. The detection achieved an accuracy of 94.21%. However, the study's evaluation focuses on accuracy and related metrics but does not consider other aspects, such as detection speed and resource consumption, and does not evaluate the performance of the proposed FL-based IDS on real-world networks.

Table 11 summarizes studies that have utilized FL-based approaches for detecting cyber-attacks in SDN.

TABLE XI. FEDERATE LEARNING-BASED DETECTION.

Ref.\ Year	Method	Dataset Used/Private, Public	Type of Attacks	Target Plane	No. of Controller	Controller Type	Scope	Results
[70] 2024	DNN, CNN, LSTM.	CICIDS 2019 dataset/public	UDP Flood, TCP SYN, and DNS Flood.	Data plane.	Single controller	--	Detection	ACC 99.99%, PREC 100%, Recall 99.99%, F1-score 99.99%
[78] 2024	FEDAVG algorithm, Federated Learning DDoS (FLDDoS) and lightweight residual network (LwResnet)	CICIDoS2019 dataset/public	UDP floods, SYN floods, and HTTP floods.	Control plane	Single controller	--	Detection and mitigation	In this approach, significant improvements in convergence time and accuracy are demonstrated.
[29] 2024	DT, RF, KNN, RSA encryption	N-BaIoT dataset/public	Mirai, Bashlite attacks	Control and data plane	Single controller	--	Detection	RF ACC 99.6%, DT ACC 99.8%, and K-NN ACC 99.3%
[24] 2023	FL, LSTM	UNSW-NB15, NF-UQ-NIDS-v2, CICIDS2017 dataset/public	--	Control and data plane	Single controller	RYU	Detection	UNSW-NB15 ACC 99%, NF-UQ-NIDS ACC 99%, CICIDS2017 ACC 97%
[71] 2023	FL with GAN	InSDN dataset/public	DoS, DDoS, Brute force, Probes Web, Botnet attacks	Control plane	Multiple controller	--	Detection	--
[32] 2023	FL, SMPC, Blockchain	NSL-KDD dataset/public	DDoS, Probe, U2R, R2L	Control and data plane	Multicontroller	--	Detection	ACC 89%, PREC 90%, Recall 89%, F1-score 89%
[79] 2022	Federated DQN, Q-learning algorithm	Not mention	DoS, TCP SYN flood, Link layer flood	Control and data plane	Multicontroller	ONOS	Detection and mitigation	This approach effectively safeguards victims from malicious packets and significantly reduces ratio of QoS-violated traffic flows.
[72] 2022	FL:(FedPlus), LSTM, GRU, CNN.	NF-UQ-NIDS dataset/public	APTs	Control plane	Single controller	ONOS	Detection	This approach achieves high accuracy and

Ref.\ Year	Method	Dataset Used/Private, Public	Type of Attacks	Target Plane	No. of Controller	Controller Type	Scope	Results
								enhances the effectiveness of FL in improving the DR of unknown threats.
[80] 2022	FL with DP, HE, LSTM	CIC-TON-IOT 2018 dataset/public, and Self-generated dataset/private	DoS, DDoS, Botnet. and XSS attacks.	Data plane	Single controller	ONOS	Detection	ACC 94.21%, PREC 94.85%, Recall 94.12%, F1-score 94.45%

8. DATASET SOURCE

This section introduces a repository of datasets utilized in the literature to validate detection approaches, as shown in Table 12. Many research papers did not make available the sources of their dataset. Moreover, some highlighted the absence of comprehensive datasets comprising both benign and malicious traffic to assess their proposed models. This resulted in some work adopting two distinct datasets for their experiments. Some datasets seem outdated and were collected for conventional networks. Hence, it is essential to obtain a new recent dataset for SDN-based networks to assess the newly proposed cyber-attack detection method. They were considering the rapid development of technology and the growing number of cyberattacks.

TABLE XII. DATASET SOURCE.

Ref.	Dataset	URL	Size of Traffic/No. of Instances
[6]	SDNTrafficDS Dataset	https://my.pcloud.com/publink/show?code=XZYm5P7ZXWd1JwSha2XTmPMtkfv2wzdXp5my	--
[16]	CICDDoS2019 Dataset	https://www.unb.ca/cic/datasets/ddos-2019.html	50,063,112 records/80 Features
[45]	Mawi working group traffic Archive Dataset	http://mawi.wide.ad.jp/mawi	1.1GB
[46]	CAIDA DDoS2007 Dataset	http://www.caida.org/data/passive/ddos20070804dataset.xml	7000
[48]	Self-generated Dataset	https://www.researchgate.net/publication/292967044	2,160,668 records/27 Features
[5]	NSL-KDD Dataset	https://www.unb.ca/cic/datasets/nsl.html	108,400 records/41 Features
[18]	DDoS attack SDN Dataset	https://data.mendeley.com/datasets/jxpfjc64kr/1	1,04,345 records/23 Features
[56]	KDD Cup 1999 Dataset	http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html	4,900,000 records/41 Features
[13]	CICIDS 2017 Dataset	https://www.unb.ca/cic/datasets/ids-2017.html	2,830,743 records/80 Features
[25]	CTU-13 Dataset	https://www.stratosphereips.org/datasets-ctu13	1.8GB
[14]	ISCX2012 Dataset	http://www.unb.ca/research/iscx/dataset/iscx-IDS-dataset.html	1000 packets
[29]	N-BaIoT Dataset	https://archive.ics.uci.edu/dataset/442/detection+of+iot+botnet+attacks+n+baiot	7,062,606 records\115 Features
[24]	UNSW-NB15 Dataset	https://research.unsw.edu.au/projects/unsw-nb15-dataset	2,540,044 records/49 Features
[72]	NF-UQ-NIDS Dataset	https://rdm.uq.edu.au/files/e2412450-ef9c-11ed-827d-e762de186848	75,987,976 records
[73]	InSDN Dataset	https://aseados.ucd.ie/datasets/SDN/	5.33 GB/80 Features
[80]	CIC-TON-IOT Dataset	https://rdm.uq.edu.au/files/127784c0-ef9d-11ed-a964-b70596e96ad5	5,351,760 records\83 Features.

9. EVALUATION METRICS

The evaluation metrics discussed in this paper are derived from the literature in Sections 8.1–8.4. There are two main categories of performance evaluation metrics: detection and computation. The detection performance metrics encompass commonly used measures employed by researchers to validate their approaches. These measures are based on a confusion matrix, which evaluates the performance of classification algorithms, including TP, TN, FP, and FN. A total of 14 performance metrics fall under this category and have been widely adopted in existing studies.

Table 13 shows that accuracy (ACC) is the most frequently used evaluation metric, appearing in 62 studies. It is followed by recall (37), precision (PREC) (38), F-measure (40), FPR (22), ROC (16), DR (17), and FAR (9). Furthermore, AUC, TPR, and FNR are used in 8, 5, and 5 studies, respectively. However, fewer studies have incorporated other evaluation metrics, such as specificity (SPEC), sensitivity (SEN), and TNR.

On the other hand, the computational performance metrics assess the computational aspects of the proposed approaches. The literature studies identify 21 distinct computational performance metrics. For example, Table 13 highlights that 5 studies evaluated their approaches on the basis of training and testing times. Additionally, the CPU utilization, detection

time, mitigation time, processing time, and MCC were examined in 4, 3, 2, 2, and 2 studies, respectively. Moreover, the remaining computational performance metrics were investigated in only a few studies, indicating their uniqueness as computational performance metrics. In conclusion, it is recommended that the research community employ various evaluation metrics rather than relying solely on one or a few conventional metrics, as they may no longer be sufficient criteria for evaluating contributing approaches.

TABLE XIII. PERFORMANCE EVALUATION USED IN THE ARTICLES.

Re.	Detection	Computational
[39]	ACC, DR, FAR.	--
[16]	ACC, Recall, PREC, F1-score.	--
[40]	DR, FPR.	--
[15]	ACC, Recall, PREC, F1-score, FPR, AUC, ROC.	--
[41]	DR, FPR.	--
[42]	ACC.	Number of detection algorithm invocations.
[43]	ACC, FPR.	CPU utilization.
[44]	ACC, Recall, PREC, F1-score.	Training time.
[45]	ACC, recall, F1-score, TPR, FPR, ROC, AUC.	--
[46]	DR, FPR, ROC.	Timeliness of recovery.
[3]	DR, FPR.	--
[47]	DR, FPR.	Response time, CPU utilization.
[4]	ACC	Detection time, Mitigation time.
[38]	ACC, Recall, PREC, F1-score.	--
[48]	ACC, FNR.	--
[49]	TPR, FPR.	--
[50]	ACC, Recall, PREC, F1-score, ROC, TPR, FPR	--
[5]	ACC, DR, FAR, F1-score.	--
[73]	ACC, Recall, PREC, F1-score.	--
[51]	ACC, Recall, PREC, F1-score, TPR, FPR.	--
[52]	ACC, Recall, PREC.	--
[53]	ACC, Recall, PREC, F1-score.	MCC, training time, detection time.
[74]	ACC	Throughput, packet loss, and packet delivery ratio.
[17]	ACC, Recall, PREC, F1-score, TPR, FPR, ROC, DR, FAR.	Prediction speed, training time.
[54]	ACC, Recall, PREC, F1-score, ROC, AUC.	--
[18]	ACC, PREC, F1-score, DR, FAR, SPEC.	--
[27]	ACC, Recall, PREC, F1-score, FAR.	Training time.
[55]	ACC, Recall, PREC, F1-score, AUC, ROC.	Training time.
[49]	ACC, ROC.	--
[6]	ACC, FAR, DR.	--
[56]	ACC, FPR.	--
[57]	ACC, PREC, recall.	--
[58]	ACC, DR, FAR.	--
[19]	ACC, FPR.	Processing time.
[59]	DR, FAR.	CPU utilization.
[37]	ACC, PREC, F1-score, FPR.	--
[33]	ACC, Recall, PREC, F1-score, SPEC, ROC.	--
[34]	ACC, Recall, PREC, F1-score.	--
[35]	ACC, PREC, recall, F1-score.	--
[22]	ACC, PREC, F1-score, DR, FPR.	--
[60]	ACC, Recall, PREC, F1-score.	Computational overhead.
[61]	ACC, Recall, PREC, F1-score, SEN, SPEC, FDR, FNR, FPR.	MCC.
[1]	ACC, Recall, PREC, F1-score.	--
[62]	ACC, Recall, F1-score, AUC, ROC.	Log loss.
[63]	ACC, FPR.	Detection time.
[26]	ACC, Recall, PREC, F1-score, AUC	--
[13]	ACC, Recall, PREC, F1-score.	--
[76]	ACC, Recall, PREC, F1-score.	--
[64]	ACC, Recall, PREC, F1-score, FPR, FNR.	--

Re.	Detection	Computational
[65]	ACC, Recall, PREC, F1-score, ROC.	--
[66]	ACC	Mitigation time.
[77]	ACC, Recall, PREC, F1-score, ROC, AUC.	--
[20]	ACC	Traffic flow granularity, Forwarding performance degradation.
[67]	ACC, Recall, PREC, F1-score, DR.	--
[68]	ACC, Recall, PREC, F1-score, SPEC.	--
[31]	ACC, Recall, PREC, F1-score.	Processing time.
[69]	ACC, FAR, FPR.	Iteration (execution time in ms).
[28]	ACC, PREC, Recall, F1-score, FPR, FNR, ROC.	--
[25]	ACC, Recall, PREC, F1-score.	--
[14]	ACC, Recall, PREC, F1-score, FPR, FNR.	CPU utilization.
[70]	ACC, PREC, recall, and F1-score.	--
[78]	ACC	Convergence time, Communication overhead.
[29]	ACC, Recall, PREC, F1-score, AUC, ROC.	--
[24]	ACC.	--
[71]	ACC	--
[32]	ACC, F1-score, DR.	--
[79]	ACC	Attack mitigation rate and the ratio of QoS violated traffic flows.
[72]	ACC, DR.	--
[80]	ACC, Recall, PREC, F1-score.	--

Metrics such as sensitivity, specificity, accuracy, TPR, and recall are vital for assessing cyber-attack detection systems in SDN. [15] highlighted the impact of attack traffic intensity on recall, with a high volume reducing recall and a low volume also affecting recall negatively. Achieving a high TPR is crucial for effective detection, while specificity aids in TN identification. The detection of low-rate DDoS attacks is challenging because of their subtle nature, which impacts both specificity and sensitivity and ultimately influences accuracy for accurate detection.

Different models, such as entropy, ML, and DL, have varying effects on metrics when detecting cyberattacks. Studies by [51][18] and [15] have applied different models, each showing different performances in these metrics. To improve these metrics, optimization techniques such as feature engineering, hyperparameter optimization, and hybridization can be utilized. Feature engineering, as demonstrated by [18] and [37], can enhance model performance. Hyperparameter optimization, as shown in [48], can increase the specificity, sensitivity, TPR, recall, and accuracy. Hybridization (combining models), as demonstrated by authors such as [18] and [33], can result in a reduced FPR, improved precision, and F1 score compared with those of individual models in various studies.

Imbalanced datasets, as shown by [37], can skew metrics such as accuracy towards the majority class, potentially overlooking the minority class and leading to inaccurate performance assessment. Accuracy may not distinguish between FPs and FNs, which are crucial distinctions in cyber-attack detection. Misclassification metrics such as FPR, FNR, precision, and the F1 score are essential for assessing classification errors in SDN cyberattack detection. Achieving high precision and addressing classification errors are critical for accurate detection in SDN environments.

The ROC-AUC metric is based on sensitivity and specificity, whereas DR indicates the system's ability to detect attacks. High attack traffic levels can affect DR, with studies demonstrating better performance under increased traffic. Some ML studies combine models to increase detection rates, although the ROC-AUC and DR may not provide a complete performance picture in imbalanced data scenarios where FPs are critical. Additionally, CPU utilization is used to measure cyber-attack detection system performance in SDN, with studies showing reduced resource consumption with specific detection approaches. [59] reported high CPU usage during DDoS attacks but noted that mitigation modules help return CPU usage to normal levels.

10. RESEARCH STRENGTHS AND WEAKNESSES

In this section, a comprehensive review of 60 articles that focused primarily on cyber-attack detection in SDN is presented. The objective was to assess the strengths and weaknesses of these articles. The evaluation of methods considers factors such as effectiveness in attack detection, scalability, adaptability, robustness, and the ability to address emerging cyberattacks. Strengths were identified when a method demonstrated superior performance in these aspects, whereas weaknesses were identified when a method showed limitations or shortcomings in achieving the desired outcomes. The findings are summarized in Table 14.

TABLE XIV. STRENGTHS AND WEAKNESSES OF ARTICLES.

Re.	Strengths	Weakness
[16]	High accuracy, low false positive rate, scalability, and compatibility with any type of SDN controller.	It is susceptible to false positive attacks, where the entropy-based detection algorithm may mistakenly identify normal traffic as malicious. Additionally, the solution is limited in its ability to detect and classify only DDoS attacks in multicontroller SDN networks. Its absence of capability to detect further types of attacks, such as malware or phishing attacks.
[40]	Shows high DR and low FPR in identifying both low-rate and high-rate DDoS attacks aiming at either single or multiple victims.	Relies on a static threshold to distinguish between normal and attack traffic, which may not be optimal for all scenarios and requires tuning based on network environments. Moreover, the approach solely depends on the source IP address for entropy computation, potentially constraining its effectiveness in detecting sophisticated DDoS attacks that utilize evasion techniques.
[15]	Comprehensive defense framework, Adaptive detection trigger, Switch migration for overload prevention, Two-stage anomaly detection, and Cross-domain mitigation	The CC-Guard defense framework may not be fully effective in countering highly sophisticated DDoS attacks.
[42]	The proposed dynamic thresholding approach in DDoS attack detection adapts to changing network conditions, reduces resource consumption while maintaining accuracy, and exhibits robustness to initial threshold values.	The proposed system requires careful selection of the initial threshold value and not be effective against all types of attacks beyond DDoS.
[43]	Effective DDoS attack detection is possible with this technique, which also lessens controller workload, southbound communication overhead, and attack detection time.	Only two DDOS attack methods were simulated and not be a good generalization for other DDOS attacks.
[44]	High accuracy, and two-level detection.	Not effective against all types of DDoS attacks.
[45]	Employs time series analysis to detect anomalies in traffic patterns, incorporates various anomaly detection methods for enhanced accuracy, and identifies the source of attack, and appropriate countermeasures. It effectively resolves constant thresholding problems associated with statistical-based detection techniques.	Requires careful parameter tuning and considers a limited feature vector. It may not be effective against sophisticated DDoS attacks using advanced evasion techniques.
[46]	Demonstrates real-time detection capabilities and achieves high accuracy in identifying such attacks. It also enables early detection, allowing for timely recovery of the network once an attack is detected.	Two features were considered and may easily result in misjudgment The recovery algorithm is not efficient.
[3]	The detection algorithm is known for its simplicity, efficacy, and effectiveness in identifying DDoS attacks in SDN environments. It exhibits a speedy detection time of 3 to 10 seconds and maintains a high level of accuracy in its outcomes.	The detection algorithm may be prone to false positives if the entropy threshold is not accurately calibrated. Furthermore, the study's findings are based on a small-scale testbed and may not be directly applicable to larger network environments.
[47]	Provides real-time detection and mitigation of DDoS attacks, secure flow management, and supports multiple controllers for effective SDN network management.	Evaluation is limited to a specific network topology and traffic patterns, and not be effective against sophisticated or adaptive attacks.
[4]	Provides real-time detection and mitigation of DDoS attacks with low overhead. It is scalable, flexible, and easy to deploy in diverse network environments.	Relies on device support for sflow and has been evaluated in a specific network topology with a focus on ICMP flooding attacks. It may not be effective against sophisticated or adaptive attacks.
[38]	Utilizes statistical detection and mitigation techniques with dynamic attribute selection and tunable accuracy. It is effective in detecting both known and unknown attacks, including multiple attacks simultaneously. The solution also includes mitigation strategies for unfamiliar attacks.	A solution based on statistical analysis may not be effective against sophisticated attacks that can evade statistical detection. It also introduces memory overhead in the switch and traffic on the control channel between the switch and the controller.
[48]	Focuses on the early detection of low-rate DDoS attacks. It achieves this by leveraging ID metrics, which enhances the accuracy of the detection process.	Limited to low-rate DDoS attacks, and not effective against high-rate DDoS attacks.
[49]	Demonstrates high accuracy in detecting DDoS attacks while maintaining a low false positive rate, especially for low-rate attacks. By utilizing the GE metric and combining GE with ID, the method effectively identifies attacks at the early stages.	The proposed method assumes that attack traffic will have a distinct entropy distribution but lacks experiments on an SDN testbed. While it shows quick detection capabilities, it may not be effective against all types of DDoS attacks. Furthermore, setting the optimal threshold for GE proves to be challenging.
[50]	The study utilizes a newly developed dataset specifically tailored for SDN-based SCADA systems.	The study was conducted on a specific network topology and may not be generalizable to all SDN-based SCADA systems. The effectiveness of the framework depends on the variety and complexity of attack scenarios in the dataset. Additionally, the 1D-CNN model employed in the study may have challenges capturing data patterns due to its restricted consideration of local structures.
[5]	Utilizes new features and DL models to achieve effective detection in both the control and data planes. It incorporates a trust-based mitigation mechanism, resulting in improved accuracy and detection rates compared to related work.	Requires further improvements to enhance its detection performance and accuracy. The FloodDefender system, which relies on SVM for detection, may have a high deployment cost and may not be suitable for real-time detection due to computational complexity and reliance on large datasets for training DL models.
[51]	High accuracy in detecting DDoS attacks, and utilizes ML and feature selection.	The restricted number of DDoS attack scenarios within the dataset utilized in the study may restrict the capability of the suggested method to detect novel DDoS attack variants.

Re.	Strengths	Weakness
[52]	Detects DDoS attacks by classifying incoming requests, and it achieves a higher accuracy compared to the other two methods.	Small dataset size.
[53]	SVM is a strong algorithm used in the proposed methodology for managing high-dimensional data. The method analyses packet headers on the fly, reducing the necessity for extensive data storage and processing capacity. This strategy is computationally efficient, conserves space, and does not require specialized equipment.	It is important to explore and refine threshold values about the number of hosts to improve the effectiveness of the approach. Moreover, the utilization of MATLAB instead of SDN environment and the comparatively lower predictive accuracy of CNN model underscore aspects that could be improved for potential improvement.
[74]	The study conducts a comprehensive performance comparison of the POX and RYU controllers, using well-defined metrics. It provides practical recommendations for selecting the appropriate controller based on the observed performance results.	Evaluation of only two controllers and two assessments. The employment of custom topology may not precisely represent real-world networks, and there is an absence of detailed analysis of the performance distinctions detected among controllers.
[17]	Offers several advantages such as being low-cost, easy to implement, and compatible with existing SDN controllers.	The research dataset is generated from a controlled simulation environment, which may restrict its ability to encompass real-world attack scenarios like novel or zero-day DDoS flooding attacks.
[54]	Demonstrates improved accuracy and efficiency in detecting network attacks. It enables real-time monitoring and classification of attacks.	The study specifically focuses on tree-based ML algorithms. Consequently, the proposed system may not be as effective in detecting novel or zero-day attacks.
[18]	Ability to differentiate flash crowd events. The evaluation dataset is derived from SDN emulated environment. The approach attains high accuracy by use of novel features and implementation hybrid model.	The dataset is used in the research is restricted to only three variants of DDoS traffic, which may not be representative of all DDoS attack scenarios. The findings may not apply to all SDN platforms, and dataset was emulated rather than generated in real-time, which may impact the validity of results.
[128]	The research improves accuracy and generalization by minimizing noise through KPCA and optimizing SVM parameters using GA. The model also uses N-RBF kernel function and allows for minimized training time and testing in a multicontroller context.	The model demonstrates well in detecting attack traffic in a single controller environment, it may encounter challenges in detecting attack traffic in a multicontroller environment and high training time of SVM.
[55]	The study focuses on achieving fine-grained detection for low-rate DDoS attacks and emphasizes high detection accuracy. .	The proposed method has limitations such as the requirement for access to flow table information, potential overhead on the SDN controller, limited effectiveness against sophisticated low-rate DDoS attacks, and scalability issues in large SDN networks with high traffic volumes.
[75]	Detects three types of DDoS attacks to protect the controller from being a single point of failure. The study reveals that DT is efficient, while RF rest achieves the highest accuracy among the models evaluated. Additionally, both DTs and RF can select optimal parameters for the classification process.	The study finds that while RF has high accuracy, it also has a longer processing time. The implemented algorithms are not designed for real-time detection of new types of attacks. The study focuses on three types of attacks and reveals that accuracy for detecting controller attacks is relatively low due to the equal distribution of the training database.
[56]	The paper proposes a real-time anomaly detection algorithm based on traffic flow analysis.	The study may not effectively detect sophisticated DDoS attacks that can evade detection. Additionally, the hybrid approach used in the study may introduce computational complexity and overhead.
[57]	The study presents a two-level security system that utilizes ML algorithms. The system is evaluated using a standard dataset.	The system proposed in the study is evaluated only in a simulated SDN environment. The training process is time-consuming and the dataset used may not be representative of real-world DDoS attacks, which could limit its effectiveness.
[58]	The study implemented a detection method for DDoS attacks using SDN architecture and SVM. The method achieved high accuracy with a low false alarm rate by combining a six-tuple feature vector and utilizing a small number of flows for detection.	The method does not effectively detect sophisticated DDoS attacks that can modify attack patterns. The study acknowledges the need for a large amount of training data and recognizes the limitations of the simulated legitimate traffic. Furthermore, no mitigation measures were implemented in the study.
[19]	The research introduces lightweight and effective algorithm deployment on SDN platform for elastic deployment. The method is assessed using a real-world dataset and utilizes SOM algorithm to detect flooding attacks. The research also handles other controller concerns such as performance and load distribution.	The SOM + k-NN algorithm has reduced accuracy in contrast to using k-NN unaccompanied. The SOM distributed center algorithm demonstrates higher FPR. It is vital to reflect that the experimentation was conducted on limited topology, which may limit the generalizability of results.
[59]	The research aims to attain accurate and timely detection of security risks with a focus on efficient mitigation strategies. The suggested method is lightweight and protocol-independent, making it suitable for diverse network environments.	The proposed method has vulnerabilities against sophisticated attacks that can evade detection.
[37]	High detection accuracy, Low FPR, and robust feature selection mechanism.	The approach is not effective against novel or zero-day DDoS attacks, computationally expensive during training and deployment.
[33]	High accuracy in detecting DDoS attacks, Combination of multiple DL algorithms, and handling of both volumetric and low-rate DDoS attacks.	Susceptible to adversarial attacks, and requires a large amount of labelled data for training.
[34]	High accuracy and robustness, and novel approach that combines GAN with DBN-LSTM.	Requires large training data and can be time-consuming and not effective against all types of adversarial attacks.
[35]	The study provides a complete analysis of methods employed to address the class imbalance problem within SDN. The insights presented are valued by both researchers and individuals in the field of network security.	The study has constrained in exploration of diverse balancing techniques and classification models, possibly missing out on effective approaches. The problem of class overlaps, which can affect classification performance, is also not handled in the study.
[22]	Comprehensive feature selection and effective mitigation strategy based on graph theory.	The research focuses on botnet-based flooding DDoS attacks in SDN environments and limits its relevance to other types of attacks. The efficiency of

Re.	Strengths	Weakness
		the suggested mitigation strategy and its computational difficulty are lacking in the study.
[60]	A novel approach to packet injection attack detection and mitigation in SDN networks.	There is a concern regarding the computational overhead of DL models in real-time network monitoring and the necessity for ongoing updates to address evolving attack techniques.
[62]	The EDL-IDS is highly accurate and effective at detecting DDoS attacks. The ensemble approach helps to improve the robustness of the model.	The model is computationally expensive and requires a large amount of data for training.
[63]	Two-level detection method that integrates information entropy analysis and DL methods. This approach attains high accuracy and effectiveness in identifying several types of attacks.	The document does not provide details on the dataset used for training and evaluation, requires additional software installation on the controller, and not be suitable for all types of DDoS attacks.
[26]	The model is shown to be more effective than traditional detection methods.	The study has limitations regarding evaluation information, generalizability, and explanation of the hybrid architecture. Important computational requirements are also lacking, and further research is needed to validate the model's effectiveness and practicality in real-world scenarios.
[13]	High detection accuracy, real-time detection, and low resource consumption.	The study focuses on DDoS attack detection in SDN environments and may not generalize to other types of attacks or network architectures.
[76]	The power of DL algorithms for DoS attack detection, and the use of the InSDN dataset precisely designed for SDN is a strength.	The performance of the models is affected by the limited number of samples for certain attack classes in the dataset. The size and quality of the dataset also play a role in influencing the performance of the proposed framework.
[65]	High detection accuracy, scalability, and cost-effectiveness.	The study evaluates the proposed framework using a single dataset, raising concerns about its generalizability to other datasets and network environments. Furthermore, the high computational complexity of DL models may hinder their deployment in resource-constrained environments.
[66]	Detects multiple types of DDoS attacks, can learn normal network behavior, and near real-time detection.	The approach not be effective against zero-day DDoS attacks, and requires a trained LSTM model for each flow attribute.
[77]	Uses a novel DL approach, evaluated on a comprehensive dataset, and achieves high accuracy in attack detection.	The model is not evaluated on real-world SDN deployments.
[20]	Improving anomaly detection through fine-grained traffic flow monitoring. It utilizes deep reinforcement learning for efficient policy optimization. Additionally, the approach proactively protects SDN switches from experiencing degradation in forwarding performance.	Restricted views of individual nodes, inflexible flow matching strategies, potential performance degradation with dynamic traffic flows, and inadequate adjustment of flow granularity affecting anomaly detection.
[68]	The proposed model is highly accurate and does not require any predefined threshold.	The datasets used are synthetically generated, so not represent real-world traffic patterns, and not be able to generalize well to unseen data.
[31]	High accuracy in detecting and preventing attacks, Low processing delay, and adaptability to different network environments.	The study does not investigate the influence of different network configurations and traffic patterns on the system's performance. Furthermore, the generalizability of the system to all types of attacks is not thoroughly explored.
[69]	Presents a novel IDS for detecting DDoS attacks in SDN. By combining LOA and CNN, the proposed system achieves high accuracy and reduces the false alarm rate, making it effective for DDoS detection.	The paper does not evaluate the performance of the proposed IDS on a large-scale network and with other state-of-the-art IDS.
[28]	High detection accuracy, low computational complexity, and utilizes a state-of-the-art dataset.	The proposed solution is evaluated on a single dataset. The impact of different network configurations on the performance of the solution is not explored.
[25]	Session-based feature extraction lets for the detection of diverse botnet structures and protocols. SDN integration enables efficient isolation and quarantine of infected machines. A dynamic and programmable approach offers flexibility and adaptability.	The proposed method was not tested on real-world bot-infected machines, only on precaptured datasets. The system relies solely on session-based features and could benefit from incorporating time-interval-based features for improved detection of internal botnet propagation.
[14]	Leveraging the power of DL for accurate and adaptable DDoS attack detection, Integrating the solution into an SDN environment for real-time mitigation, and validating the approach through experimental results and real-time attack scenarios.	The approach is not effective in handling low-rate attacks.
[70]	Protects user data privacy, reduces bandwidth usage, latency, and server overhead, and achieves high accuracy in detecting DDoS attacks.	Requires multiple clients to participate in the FL process, training time can be longer than centralized models, and security and privacy concerns need to be addressed
[29]	Enhanced security, efficiency, and overall performance of FL systems, as well as improved network management and control through SDN.	Communication overhead and privacy concerns may impact performance.
[24]	Preserves data privacy, achieves high accuracy in anomaly detection, and can be applied to complex SDN topologies.	Requires a large amount of data for training, and is not suitable for real-time applications
[71]	This study provides valuable insights into the impact of non-IID data on FL-based IDS. It identifies a set of features that are most important for intrusion detection in the InSDN dataset.	The study does not address the challenges of data imbalance, or overfitting in non-IID scenarios, and does not evaluate the performance of the proposed FL-based IDS on real-world networks.
[32]	Privacy-preserving collaborative learning, high accuracy and efficiency, decentralized and efficient collaboration management.	Requires a trusted third party for secure aggregation, limited scalability for large-scale deployments, potential vulnerabilities in underlying technologies, and high computational overhead.
[79]	Rapid policy learning through federated DQN, scalability to distributed SDN deployments, effective mitigation of DoS attacks, and reduced QoS violations for legitimate traffic.	The performance evaluation is limited to DoS attacks. The scalability of FEAR with a larger number of SDN nodes needs further investigation.
[72]	Data privacy preservation through FL, high accuracy in threat detection, and scalability to large SDN networks.	Requires careful selection of enrichment data to avoid overwhelming or insufficient information, high computational and communication overhead

Re.	Strengths	Weakness
		associated with FL, limited real-time capabilities, and OpenCTI system not yet implemented.
[80]	A novel approach to privacy-preserving threat hunting using FL, effective combination of DP and HE for enhanced privacy protection, efficient big data processing using Apache Spark, and evaluation on both public dataset and real network traffic.	The paper lacks a comprehensive comparison with existing threat-hunting systems for SDN. The evaluation focuses on accuracy and related metrics but does not consider other aspects such as detection speed and resource consumption, and does not evaluate the performance of the proposed FL-based IDS on real-world networks.

11. RESEARCH GAPS AND FUTURE DIRECTIONS

In this section, we examine the current challenges encountered by research publications that focus on cyber-attack detection and mitigation techniques. By discussing these challenges researchers in the field can identify areas for further exploration and propose appropriate solutions for enhancing cyber-attack detection.

While SDN offers numerous benefits, it also faces several issues. One notable concern is the security threats arising from the separation of the data and control planes within SDN. This separation has made SDN planes susceptible to cyberattacks allowing attackers to target any plane within the SDN infrastructure. Safeguarding SDN from these attacks is a crucial issue that must be addressed to ensure uninterrupted access to services for legitimate users. Moreover, addressing this issue is vital for fully harnessing the potential benefits of SDN. We highlight certain gaps that researchers can explore to conduct further investigations and develop novel approaches to enhance security of SDN.

1. Standard dataset: Article [27][54][57] and other reviewed articles have relied primarily on traditional and publicly available datasets such as the KDDCUP99 dataset, CAIDA 2007 dataset, and DARPA dataset. However, these datasets may not be suitable for detecting attacks in SDN environments, as they were designed for traditional networks. Some authors, as mentioned in [6][58], have created datasets for implementing their detection approaches, but these datasets are not publicly available for further validation. On the other hand, an SDN-based dataset was created and made publicly accessible, but the dataset included only three types of attacks during its creation. Given the increasing sophistication and stealth of cyberattack techniques, there is a persistent need to develop a more comprehensive SDN-based dataset that encompasses various forms of attacks. In general, the availability of cyber-attack datasets specifically tailored for SDN is limited. Future research should focus on developing standardized datasets for SDN, as identified in this review, where current datasets such as KDD-CUP99 are outdated and not reflective of modern network traffic.

2. Feature Engineering in SDN-based cyber-attack detection: In the context of SDN-based cyber-attack detection, feature selection plays a crucial role in determining the performance of such detection systems. Surprisingly, only a limited number of systems prioritize feature selection as their primary methodology. The identification of cyber-attacks can be greatly facilitated by the utilization of specific features. By analysing and incorporating these features into a dataset, the detection of attacks can be significantly improved. Notably, [27][6][58] explored several key features for cyber-attack detection. These commonly studied features extracted from SDN include the duration of flows, the entropy of source/destination IP addresses, the number of packets per flow, the number of bytes per flow, and the entropy of source/destination ports. While it is crucial to have informative features, it is equally important to possess the best and most concise feature set for effective detection. This is where feature selection methods come into play, as they facilitate the identification of the most relevant features for detection purposes. Future research should encourage researchers to create an SDN-specific dataset based on essential features and then apply feature selection techniques to detect attacks better, where many researchers tend to overlook the significance of feature selection when developing cyber-attack detection techniques and not accurately reflect their effectiveness in detecting and mitigating cyber threats.

3. Low-rate DDoS detection: In the realm of DDoS detection, it is important to note that the majority of research focuses on identifying high-rate DDoS attacks, while few works address low-rate DDoS attacks. However, recent incidents, such as attacks on AWS, indicate a shift towards more covert low-rate attacks than high-volume DDoS attacks. It is crucial to detect low-rate DDoS attacks promptly, as they have the potential to cause significant harm. Detecting these attacks poses a challenge because their average traffic volume closely resembles regular traffic flow, making them difficult to distinguish. Therefore, it is imperative not to overlook these attacks, as they can gradually disrupt benign traffic over time. Several research papers, including [48][55], have proposed detection strategies specifically tailored for low-rate DDoS attacks. The challenge lies in identifying the distinguishing characteristics of these attacks and developing a detection system with a low FPR and TPR. Addressing this challenge is crucial for the effective mitigation of low-rate DDoS assaults. Future research in the field of DDoS detection should develop techniques specifically tailored for detecting low-rate DDoS attacks, where current detecting attacks may not be sufficient for detecting low-rate attacks owing to the lower packet count.

4. Distributed SDN controllers: The majority of the current literature focuses on security approaches that are based on a topology with a single network controller, as is evident in studies such as [40][46][17][22]. However, this topology is susceptible to single points of failure in the event of cyberattacks. On the other hand, a network that employs distributed

controllers in either a flat or hierarchical design offers numerous advantages, including improved load distribution, consistency, and scalability. Moreover, as the severity of cyberattacks continues to increase, the presence of distributed controllers enables the maintenance of network efficiency even when the central controller becomes a bottleneck. These distributed controllers can effectively mitigate the impact of cyber-attacks, reduce communication overhead, eliminate single points of failure, and facilitate load balancing of traffic flow across multiple controllers. Consequently, the operation of distributed SDN controllers remains an open security challenge that warrants further investigation to increase network resilience and defend against cyberattacks. Future research should focus on developing innovative security approaches for distributed SDN controllers. The existing topology is based on a single network controller topology which is vulnerable to single points of failure during cyber-attacks and does not facilitate load balancing of traffic flow across controllers.

5. Detecting a wide range of attacks: Most existing studies focus on DDoS attacks. Other significant attacks such as MitM attacks, insider threats, zero-day exploits, data exfiltration, DNS spoofing, and cache poisoning, receive comparatively less attention. To address this future research should focus on developing advanced detection mechanisms to enhance defensive capabilities against a wide range of cybersecurity attacks that have not been adequately explored in SDN environments. Current attacks such as DDoS are well known and do not reflect the strengthening of network resilience and security posture.

6. Hyperparameter tuning (hyperparameter optimization): Hyperparameter tuning is critical aspect of the ML and DL approaches, as it involves adjusting various parameters to optimize performance. Fine-tuning these models is essential to achieve optimal parameters for effective training and to minimize potential negative impacts. By appropriately adjusting hyperparameters, the ML and DL approaches can achieve enhanced performance. Future research should focus on performing hyperparameter tuning via algorithms such as Bayesian optimization, the GA, particle swarm optimization, grid search, random search, and reinforcement learning. These algorithms can aid in efficiently searching and optimizing the hyperparameters of ML and DL models to improve their performance in detecting, mitigating, or preventing cyberattacks in SDN. Further exploration and implementation of these hyperparameter tuning techniques can significantly increase the effectiveness and efficiency of ML and DL approaches in the cybersecurity domain.

7. Optimization of detection: Optimization of detection in SDN networks presents a significant opportunity to enhance cyber-attack detection capabilities. One approach to achieve this is by exploring the integration of DL approaches with 'relevance feedback'. By incorporating 'Relevance Feedback', which allows the system to learn from user interactions and feedback, the detection capabilities can be strengthened through continuous improvement and adaptation to evolving threats. Additionally, introducing deep neural networks with innovative new activation functions and a change in the operation method of kernel filters can further increase the efficiency of cyberattack detection in SDN settings. By leveraging advanced activation functions and kernel filters, the deep neural network can better capture complex patterns and anomalies in network traffic data, leading to improved accuracy and effectiveness in detecting cyber threats. Finding the optimization of detection in SDN is a significant challenge that requires focused research efforts.

8. Prevention approach for cyber-attacks: It is evident from the reviewed literature, including studies such as [27][45][55][19], that the majority of research has focused primarily on the detection and mitigation of SDN cyber-attacks rather than prevention. There is a notable lack of approaches that specifically address prevention measures. However, emphasizing the importance of prevention in safeguarding the functionality of the SDN network is crucial. Preventing cyberattacks is more urgent and crucial than merely detecting and mitigating them, as it aims to halt their propagation into the network and prevent the consumption of valuable network resources. Future research should prioritize the development of comprehensive strategies that integrate effective prevention, detection, and mitigation of cyberattacks in SDN environments. Addressing this challenge requires focused attention and further investigation in the field.

9. Actual testbed for simulation: Notably, several cyber-attack detection techniques have utilized either simulation or emulation as a means to validate their detection approaches. However, it is important to consider that the use of simulated SDN environments under virtual host machines with limited resources, as observed in studies such as[46], may not accurately reflect real-life implementations. These simulations often employ small network topologies, which may not adequately represent the vast internet resources and high bandwidth from which cyberattacks are typically launched. Therefore, there is a clear need for research that employs real testbeds with large network topologies to effectively demonstrate and validate cyber-attack detection approaches more realistically and comprehensively. Future research should focus on developing cyber-attack detection techniques that are validated in real-world testbeds with large network topologies.

10. Collecting traffic statistics: Many cyber-attack detection approaches, such as the one proposed by [59], rely on the traditional OpenFlow protocol for gathering traffic features. However, it is important to consider that using OpenFlow for collecting traffic statistics on large-scale networks may result in data plane overhead. Additionally, in the case of high-rate DDoS attacks, the controller data bandwidth can become overwhelmed, leading to potential connection disruptions between switches and delayed responses from the controller. While flow management mechanisms have been utilized as alternatives, they often cannot gather comprehensive packet details. Future research should focus on developing cyberattack detection

approaches that enable the collection of traffic statistics without causing additional overhead on the SDN architecture. The development of efficient and lightweight data collection techniques is crucial for improving the scalability and performance of cyber-attack detection systems in SDN environments.

11. Security solutions for other planes in SDN: It is evident from the comprehensive review conducted that the majority of researchers have focused primarily on providing security solutions for the control plane to combat cyber-attacks such as [16][45][51][57]. However, few studies address security solutions for the data plane and application plane in SDN environments. Focusing on enhancing security measures for these planes is crucial to ensure comprehensive protection against cyber threats across all aspects of the network infrastructure. By directing research efforts towards developing innovative security solutions for the data plane and application plane, researchers can improve the overall security posture of SDN environments and enhance their resilience against evolving cyberattacks. Future research should focus on developing security solutions for the data plane and application plane in SDN environments.

As a result, there is an important research gap in terms of providing comprehensive security solutions for these other planes and SDN switches. Addressing this gap and developing effective security measures for all planes within SDN architecture is an open research challenge that requires attention and investigation.

12. CONCLUSION

The study provides an overview of the SDN architecture model, the OpenFlow forwarding process, and cyberattacks on SDN networks. It also constructed six research questions concerning entropy, ML, DL, and FL approaches for detecting SDN cyberattacks. To address these questions, a systematic literature review method was employed to conduct an in-depth analysis and synthesis of literature spanning seven years from 2017--2024. This process led to the selection of 69 primary studies deemed pertinent to the research inquiries, following strict inclusion and exclusion criteria to ensure high-quality research selection.

Moreover, the significant findings of this SLR show that the number of publications is progressively increasing, particularly from 2020 onwards, and that cyberattacks have recently been around for a few years. Additionally, the analysis indicates that the majority of the literature studies employed DL techniques (36%) for their analysis, followed by ML at 28%, entropy at 23%, and FL methods at 13%. Notably, ML and DL techniques have emerged as the most promising for the detection and mitigation of SDN cyberattacks. The review also considered the network simulators and tools utilized in employing and evolving these approaches, revealing that many researchers utilize the Mininet network emulator as an SDN testbed environment, with POX and RYU controllers for capturing and processing network traffic flows.

Furthermore, this SLR highlights that a majority of the examined studies desire to generate datasets due to the limited availability of realistic publicly accessible datasets. While numerous researchers have proposed cyber-attack detection solutions, only a few have explored the application of feature selection algorithms on SDN datasets. Further research is needed to identify better feature subsets for more efficient cyber-attack detection. Additionally, the review highlights the performance assessment metrics utilized by researchers to assess and validate their methodologies. The evaluation metrics fall into two main categories: detection performance metrics encompassing measures such as the confusion matrix, ROC, AUC, and detection accuracy, which are commonly employed in the literature. On the other hand, computational performance metrics assess their approaches on the basis of factors such as training and testing times, as well as CPU utilization. Finally, the systematic literature review identifies the strengths and weaknesses of the reviewed articles, illuminates research gaps, and proposes future research directions aimed at advancing cyber-attack detection in SDN environments.

The implications of this study are significant for the academic and professional community in the field of cybersecurity. It provides a robust foundation for understanding the challenges and opportunities in cyber-attack detection in SDN environments, as well as identifying key areas that need further research and development. This systematic review offers a clear direction for future research and highlights the importance of addressing existing gaps in cyber-attack detection in SDN. These outlined research directions provide a roadmap for researchers to conduct further studies and develop new methods to safeguard SDN environments.

Funding

The authors had no institutional or sponsor support.

Conflicts of interest

The author’s disclosure statement confirms the absence of any conflicts of interest.

Acknowledgement

The author would like to thank Mustansiriyah University (https://uomustansiriyah.edu.iq/) in Baghdad–Iraq for its support in the present work.

References

[1] J. Wang and L. Wang, "SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN," Sensors, vol. 22, no. 21, 2022, doi: 10.3390/s22218287.

[2] Z. S. Younus and M. Alanezi, "A Survey on Network Security Monitoring: Tools and Functionalities," Mustansiriyah J. Pure Appl. Sci., vol. 1, no. 2, pp. 55–86, Jul. 2023, Accessed: Aug. 05, 2024. [Online]. Available: https://mjpas.uomustansiriyah.edu.iq/index.php/mjpas/article/view/33

[3] T. Omar, A. Ho, and B. Urbina, "Detection of DDoS in SDN Environment Using Entropy-based Detection," 2019 IEEE Int. Symp. Technol. Homel. Secur. HST 2019, pp. 1–6, 2019.

[4] B. Lawal and N. At, "Real-Time Detection and Mitigation of Distributed Denial of Service (DDoS) Attacks in Software Defined Networking (SDN)," no. May, pp. 1–5, 2018, doi: 10.1109/SIU.2018.8404674.

[5] W. G. Gadallah, H. M. Ibrahim, and N. M. Omar, "A deep learning technique to detect distributed denial of service attacks in software-defined networks," Comput. Secur., vol. 137, no. February, p. 103588, 2024, doi: 10.1016/j.cose.2023.103588.

[6] M. Myint Oo, S. Kamolphiwong, T. Kamolphiwong, and S. Vasupongayya, "Advanced Support Vector Machine-(ASVM-) based detection for Distributed Denial of Service (DDoS) attack on Software Defined Networking (SDN)," J. Comput. Networks Commun., vol. 2019, 2019, doi: 10.1155/2019/8012568.

[7] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," J. Inf. Secur. Appl., vol. 50, p. 102419, Feb. 2020, doi: 10.1016/J.JISA.2019.102419.

[8] A. M. Aleesa, B. B. Zaidan, A. A. Zaidan, and N. M. Sahar, "Review of intrusion detection systems based on deep learning techniques: coherent taxonomy, challenges, motivations, recommendations, substantial analysis and future directions," Neural Comput. Appl., vol. 32, no. 14, pp. 9827–9858, Jul. 2020, doi: 10.1007/S00521-019-04557-3.

[9] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," J. Netw. Comput. Appl., vol. 169, p. 102767, Nov. 2020, doi: 10.1016/J.JNCA.2020.102767.

[10] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Trans. Emerg. Telecommun. Technol., vol. 32, no. 1, Jan. 2020, doi: 10.1002/ETT.4150.

[11] S. Gupta and D. Grover, "A Comprehensive Review on Detection of DDoS Attacks using ML in SDN Environment," Proc. - Int. Conf. Artif. Intell. Smart Syst. ICAIS 2021, pp. 1158–1163, Mar. 2021, doi: 10.1109/ICAIS50930.2021.9395987.

[12] B. Alhijawi, S. Almajali, H. Elgala, H. Bany Salameh, and M. Ayyash, "A survey on DoS/DDoS mitigation techniques in SDNs: Classification, comparison, solutions, testing tools and datasets," Comput. Electr. Eng., vol. 99, p. 107706, Apr. 2022, doi: 10.1016/J.COMPELECENG.2022.107706.

[13] A. Makuvaza, D. S. Jat, and A. M. Gamundani, "Deep Neural Network (DNN) Solution for Real-time Detection of Distributed Denial of Service (DDoS) Attacks in Software Defined Networks (SDNs)," 2021. doi: 10.1007/s42979-021-00467-1.

[14] C. Li et al., "Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN," Int. J. Commun. Syst., vol. 31, no. 5, p. e3497, Mar. 2018, doi: 10.1002/DAC.3497.

[15] J. Wang, L. Wang, and R. Wang, "A Method of DDoS Attack Detection and Mitigation for the Comprehensive Coordinated Protection of SDN Controllers," 2023.

[16] T. G. Gebremeskel, K. A. Gemeda, T. G. Krishna, and P. J. Ramulu, "DDoS Attack Detection and Classification Using Hybrid Model for Multicontroller SDN," Wirel. Commun. Mob. Comput., vol. 2023, pp. 1–18, 2023, doi: 10.1155/2023/9965945.

[17] A. O. Sangodoyin, M. O. Akinsolu, P. Pillai, and V. Grout, "Detection and Classification of DDoS Flooding Attacks on Software-Defined Networks: A Case Study for the Application of Machine Learning," IEEE Access, vol. 9, pp. 122495–122508, 2021, doi: 10.1109/ACCESS.2021.3109490.

[18] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, "Automated DDOS attack detection in software defined networking," J. Netw. Comput. Appl., vol. 187, no. May, 2021, doi: 10.1016/j.jnca.2021.103108.

[19] T. M. Nam et al., "Self-organizing map-based approaches in DDoS flooding detection using SDN," Int. Conf. Inf. Netw., vol. 2018-Janua, pp. 249–254, 2018, doi: 10.1109/ICOIN.2018.8343119.

[20] T. V. Phan, T. G. Nguyen, N. N. Dao, T. T. Huong, N. H. Thanh, and T. Bauschert, "DeepGuard: Efficient Anomaly Detection in SDN with Fine-Grained Traffic Flow Monitoring," IEEE Trans. Netw. Serv. Manag., vol. 17, no. 3, pp. 1349–1362, 2020, doi: 10.1109/TNSM.2020.3004415.

[21] M. Z. Abdullah, A. K. Jassim, F. N. Hummadi, and M. M. M. Al Khalidy, "NEW STRATEGIES FOR IMPROVING NETWORK SECURITY AGAINST CYBER ATTACK BASED ON INTELLIGENT ALGORITHMS," J. Eng. Sustain. Dev., vol. 28, no. 3, pp. 342–354, May 2024, doi: 10.31272/JEASD.28.3.4.

[22] M. W. Nadeem, H. G. Goh, Y. Aun, and V. Ponnusamy, "Detecting and Mitigating Botnet Attacks in Software-Defined Networks Using Deep Learning Techniques," IEEE Access, vol. 11, no. May, pp. 49153–49171, 2023, doi: 10.1109/ACCESS.2023.3277397.

[23] F. Kamil, H. Mihna, M. A. Habeeb, and Y. L. Khaleel, "Using Information Technology for Comprehensive Analysis and Prediction in Forensic Evidence," Mesopotamian journal of Cybersecurity, vol. 2024, pp. 4–16, 2024, doi.org/10.58496/MJCS/2024/002.

[24] A. A. Abd Al-Ameer and W. S. Bhaya, "Enhanced Intrusion Detection in Software-Defined Networks Through Federated Learning and Deep Learning," Ing. des Syst. d'Information, vol. 28, no. 5, pp. 1213–1220, 2023, doi: 10.18280/isi.280509.

[25] S. Maeda, A. Kanai, S. Tanimoto, T. Hatashima, and K. Ohkubo, "A Botnet Detection Method on SDN using Deep Learning," 2019 IEEE Int. Conf. Consum. Electron. ICCE 2019, pp. 1–6, 2019, doi: 10.1109/ICCE.2019.8662080.

[26] H. Wang, W. Li, J. H. Yi, and G.-J. Ahn, "DDosTC: A Transformer-Based Network Attack Detection Hybrid Mechanism in SDN," Sensors 2021, Vol. 21, Page 5047, vol. 21, no. 15, p. 5047, Jul. 2021, doi: 10.3390/S21155047.

[27] K. S. Sahoo et al., "An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks," IEEE Access, vol. 8, pp. 132502–132513, 2020, doi: 10.1109/ACCESS.2020.3009733.

[28] S. Haider, A. Akhunzada, G. Ahmed, and M. Raza, "Deep Learning based Ensemble Convolutional Neural Network Solution for Distributed Denial of Service Detection in SDNs," 2019 UK/China Emerg. Technol. UCET 2019, Aug. 2019, doi: 10.1109/UCET.2019.8881856.

[29] H. Babbar, S. Rani, A. Singh, and G. Gianini, "Detecting Cyberattacks to Federated Learning on Software-Defined Networks," pp. 120–132, 2024, doi: 10.1007/978-3-031-51643-6_9.

[30] I. A. Shakir, P. A. A. A. Saleh, and P. H. M.El-Bakry, "Use of Singular Value Decomposition for a Deep Learning-Based Fast Intrusion Detection System," J. Coll. Basic Educ., vol. 30, no. 123, pp. 73–87, Apr. 2024, doi: 10.35950/CBEJ.V30I123.11337.

[31] T. H. Lee, L. H. Chang, and C. W. Syu, "Deep learning enabled intrusion detection and prevention system over SDN networks," 2020 IEEE Int. Conf. Commun. Work. ICC Work. 2020 - Proc., Jun. 2020, doi: 10.1109/ICCWORKSHOPS49005.2020.9145085.

[32] Z. A. El Houda, A. S. Hafid, and L. Khoukhi, "MiTFed: A Privacy Preserving Collaborative Network Attack Mitigation Framework Based on Federated Learning Using SDN and Blockchain," IEEE Trans. Netw. Sci. Eng., vol. 10, no. 4, pp. 1985–2001, 2023, doi: 10.1109/TNSE.2023.3237367.

[33] H. Elubeyd and D. Yiltas-Kaplan, "Hybrid Deep Learning Approach for Automatic DoS/DDoS Attacks Detection in Software-Defined Networks," Appl. Sci., vol. 13, no. 6, 2023, doi: 10.3390/app13063828.

[34] L. Chen, Z. Wang, R. Huo, and T. Huang, "An Adversarial DBN-LSTM Method for Detecting and Defending against DDoS Attacks in SDN Environments," Algorithms 2023, Vol. 16, Page 197, vol. 16, no. 4, p. 197, Apr. 2023, doi: 10.3390/A16040197.

[35] S. M. H. Mirsadeghi, H. Bahsi, R. Vaarandi, and W. Inoubli, "Learning From Few Cyber-Attacks: Addressing the Class Imbalance Problem in Machine Learning-Based Intrusion Detection in Software-Defined Networking," IEEE Access, vol. 11, no. November, pp. 140428–140442, 2023, doi: 10.1109/ACCESS.2023.3341755.

[36] L. A. E. Al-saeedi et al., "Artificial Intelligence and Cybersecurity in Face Sale Contracts : Legal Issues and Frameworks," Mesopotamian journal of Cybersecurity, vol. 4, no. 2, pp. 129–142, 2024., doi:org/10.58496/MJCS/2024/0012.

[37] A. Mansoor, M. Anbar, A. A. Bahashwan, B. A. Alabsi, and S. D. A. Rihan, "Deep Learning-Based Approach for Detecting DDoS Attack on Software-Defined Networking Controller," Syst. 2023, Vol. 11, Page 296, vol. 11, no. 6, p. 296, Jun. 2023, doi: 10.3390/SYSTEMS11060296.

[38] L. Altay, "JESS : Joint Entropy Based DDoS Defense Scheme in SDN," vol. 8716, no. c, pp. 1–15, 2018, doi: 10.1109/JSAC.2018.2869997.

[39] H. Zhou and J. Ling, "A Cooperative Detection of DDoS attacks based on CNN-BiLSTM in SDN," J. Phys. Conf. Ser., vol. 2589, no. 1, 2023, doi: 10.1088/1742-6596/2589/1/012001.

[40] M. A. Aladaileh et al., "Effectiveness of an Entropy-Based Approach for Detecting Low- and High-Rate DDoS Attacks against the SDN Controller: Experimental Analysis," Appl. Sci., vol. 13, no. 2, 2023, doi:

10.3390/app13020775.
[41] M. A. Aladaileh, M. Anbar, A. J. Hintaw, I. H. Hasbullah, A. A. Bahashwan, and S. Al-sarawi, "applied sciences Renyi Joint Entropy-Based Dynamic Threshold Approach to Detect DDoS Attacks against SDN Controller with Various Traffic Rates," 2022.
[42] T. Wang, Y. Feng, and K. Sakurai, "Improving the Two-stage Detection of Cyberattacks in SDN Environment Using Dynamic Thresholding".
[43] S. Yu, J. Zhang, J. Liu, X. Zhang, Y. Li, and T. Xu, "A cooperative DDoS attack detection scheme based on entropy and ensemble learning in SDN," EURASIP J. Wirel. Commun. Netw., 2021, doi: 10.1186/s13638-021-01957-9.
[44] L. Wang and Y. Liu, "A DDoS Attack Detection Method Based on Information Entropy and Deep Learning in SDN," Proc. 2020 IEEE 4th Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2020, pp. 1084–1088, Jun. 2020, doi: 10.1109/ITNEC48623.2020.9085007.
[45] R. Fadaei and O. Ermi, "A DDoS Attack Detection and Defense Scheme Using Time-series Analysis for SDN," 2020.
[46] J. Cui, M. Wang, Y. Luo, and H. Zhong, "DDoS detection and defense mechanism based on cognitive-inspired computing in SDN," Futur. Gener. Comput. Syst., vol. 97, pp. 275–283, 2019, doi: 10.1016/j.future.2019.02.037.
[47] U. Gurusamy and M. Msk, "Detection and mitigation of UDP flooding attack in a multicontroller software defined network using secure flow management model," no. April, pp. 1–11, 2019, doi: 10.1002/cpe.5326.
[48] K. S. Sahoo, D. Puthal, M. Tiwary, J. J. P. C. Rodrigues, B. Sahoo, and R. Dash, "An Early Detection of Low Rate DDoS Attack to SDN Based Data Center Networks using Information Distance Metrics," Futur. Gener. Comput. Syst., 2018, doi: 10.1016/j.future.2018.07.017.
[49] K. S. Sahoo, "Detection of Control Layer DDoS Attack using Entropy metrics in SDN : An Empirical Investigation," 2017 Ninth Int. Conf. Adv. Comput., pp. 281–286, 2017.
[50] O. Polat et al., "Multi-Stage Learning Framework Using Convolutional Neural Network and Decision Tree-Based Classification for Detection of DDoS Pandemic Attacks in SDN-Based SCADA Systems," Sensors, vol. 24, no. 3, 2024, doi: 10.3390/s24031040.
[51] H. Alubaidan, R. Alzaher, M. AlQhatani, and R. Mohammed, "DDoS Detection in Software-Defined Network (SDN) Using Machine Learning," Int. J. Cybern. Informatics, vol. 12, no. 04, pp. 93–104, 2023, doi: 10.5121/ijci.2023.120408.
[52] Musmuharam and Suharjito, "Detection of Distributed Denial of Service Attacks in Software Defined Networks by Using Machine Learning," Int. J. Commun. Networks Inf. Secur., vol. 15, no. 3, pp. 13–25, 2023, doi: 10.17762/ijcnis.v15i3.6214.
[53] T. E. Ali, Y. W. Chong, and S. Manickam, "Comparison of ML/DL Approaches for Detecting DDoS Attacks in SDN," Appl. Sci., vol. 13, no. 5, 2023, doi: 10.3390/app13053033.
[54] A. O. Alzahrani and M. J. F. Alenazi, "Designing a network intrusion detection system based on machine learning for software defined networks," Futur. Internet, vol. 13, no. 5, 2021, doi: 10.3390/fi13050111.
[55] W. Zhijun, X. Qing, W. Jingjie, Y. Meng, and L. Liang, "Low-rate DDoS Attack Detection Based on Factorization Machine in Software Defined Network," vol. XX, 2020, doi: 10.1109/ACCESS.2020.2967478.
[56] G. Kaur and P. Gupta, "Hybrid Approach for detecting DDOS Attacks in Software Defined Networks," 2019 12th Int. Conf. Contemp. Comput. IC3 2019, pp. 1–6, 2019, doi: 10.1109/IC3.2019.8844944.
[57] B. V. Karan, D. G. Narayan, and P. S. Hiremath, "Detection of DDoS Attacks in Software Defined Networks," Proc. 2018 3rd Int. Conf. Comput. Syst. Inf. Technol. Sustain. Solut. CSITSS 2018, pp. 265–270, 2018, doi: 10.1109/CSITSS.2018.8768551.
[58] J. Ye, X. Cheng, J. Zhu, L. Feng, and L. Song, "A DDoS Attack Detection Method Based on SVM in Software Defined Network," Secur. Commun. Networks, vol. 2018, Apr. 2018, doi: 10.1155/2018/9804061.
[59] D. Hu, P. Hong, and Y. Chen, "FADM : DDoS Flooding Attack Detection and Mitigation System in Software-Defined Networking," 2017.
[60] A. T. Phu et al., "Defending SDN against packet injection attacks using deep learning," Comput. Networks, vol. 234, pp. 1–15, 2023, doi: 10.1016/j.comnet.2023.109935.
[61] K. Perumal and K. Arockiasamy, "Optimized deep neural network based DDoS attack detection and bait mitigation process in software defined network," Concurr. Comput. Pract. Exp., vol. 35, no. 12, pp. 1–23, 2023, doi: 10.1002/cpe.7692.
[62] U. Mbasuva and G. A. L. Zodi, "Designing Ensemble Deep Learning Intrusion Detection System for DDoS attacks in Software Defined Networks," Proc. 2022 16th Int. Conf. Ubiquitous Inf. Manag. Commun. IMCOM 2022, 2022, doi: 10.1109/IMCOM53663.2022.9721785.
[63] Y. Liu, T. Zhi, M. Shen, L. Wang, Y. Li, and M. Wan, "Software-defined DDoS detection with information entropy analysis and optimized deep learning," Futur. Gener. Comput. Syst., vol. 129, pp. 99–114, Apr. 2022, doi:

10.1016/J.FUTURE.2021.11.009.
[64] N. Ahuja, G. Singal, and D. Mukhopadhyay, “DLSDN: Deep learning for DDOS attack detection in software defined networking,” Proc. Conflu. 2021 11th Int. Conf. Cloud Comput. Data Sci. Eng., pp. 683–688, Jan. 2021, doi: 10.1109/CONFLUENCE51648.2021.9376879.
[65] S. Haider et al., “A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks,” IEEE Access, vol. 8, no. March, pp. 53972–53983, 2020, doi: 10.1109/ACCESS.2020.2976908.
[66] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proenca, “Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment,” IEEE Access, vol. 8, pp. 83765–83781, 2020, doi: 10.1109/ACCESS.2020.2992044.
[67] L. Zhou, J. Shu, and X. Jia, “Collaborative Anomaly Detection in Distributed SDN,” Proc. - IEEE Glob. Commun. Conf. GLOBECOM, 2020, doi: 10.1109/GLOBECOM42002.2020.9322364.
[68] B. Nugraha and R. N. Murthy, “Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks,” 2020 IEEE Conf. Netw. Funct. Virtualization Softw. Defin. Networks, NFV-SDN 2020 - Proc., pp. 51–56, Nov. 2020, doi: 10.1109/NFV-SDN50289.2020.9289894.
[69] D. Arivudainambi, V. K. Varun, and S. Sibi Chakkaravarthy, “LION IDS: A meta-heuristics approach to detect DDoS attacks against Software-Defined Networks,” Neural Comput. Appl., vol. 31, no. 5, pp. 1491–1501, 2019, doi: 10.1007/s00521-018-3383-7.
[70] J. Mateus, G. L. Zodi, and A. Bagula, “Federated Learning-Based Solution for DDoS Detection in SDN,” pp. 875–880, 2024.
[71] S. H. A. Kazmi, F. Qamar, R. Hassan, K. Nisar, D. P. B. Dahnil, and M. A. Al-Betar, “Threat Intelligence with Non-IID Data in Federated Learning enabled Intrusion Detection for SDN: An Experimental Study,” 2023 24th Int. Arab Conf. Inf. Technol. ACIT 2023, pp. 1–6, 2023, doi: 10.1109/ACIT58888.2023.10453867.
[72] H. T. Thi, N. D. Hoang Son, P. T. Duv, and H. Van Pham, “Federated Learning-Based Cyber Threat Hunting for APT Attack Detection in SDN-Enabled Networks,” 2022 21st Int. Symp. Commun. Inf. Technol. Isc. 2022, pp. 1–6, 2022, doi: 10.1109/ISCIT55906.2022.9931222.
[73] H. A. Hassan, E. El-Din Hemdan, M. Shokair, F. E. A. El-Samie, and W. El-Shafai, “An Efficient Attack Detection Framework in Software-Defined Networking using Intelligent Techniques,” ICEEM 2023 - 3rd IEEE Int. Conf. Electron. Eng., no. October, 2023, doi: 10.1109/ICEEM58740.2023.10319575.
[74] N. Naim, M. Imad, M. A. Hassan, M. B. Afzal, S. Khan, and A. U. Khan, “POX and RYU Controller Performance Analysis on Software Defined Network,” EAI Endorsed Trans. Internet Things, vol. 9, no. 3, pp. 1–11, 2023, doi: 10.4108/eetiot.v9i3.2821.
[75] R. Santos, D. Souza, W. Santo, A. Ribeiro, and E. Moreno, “Machine learning algorithms to detect DDoS attacks in SDN,” Concurr. Comput. Pract. Exp., vol. 32, no. 16, pp. 1–14, 2020, doi: 10.1002/cpe.5402.
[76] A. S. Alshra’A, A. Farhat, and J. Seitz, “Deep Learning Algorithms for Detecting Denial of Service Attacks in Software-Defined Networks,” Procedia Comput. Sci., vol. 191, no. 2019, pp. 254–263, 2021, doi: 10.1016/j.procs.2021.07.032.
[77] M. S. Elsayed, N. A. Le-Khac, S. Dev, and A. D. Jurcut, “DDoSNet: A Deep-Learning Model for Detecting Network Attacks,” Proc. - 21st IEEE Int. Symp. a World Wireless, Mob. Multimed. Networks, WoWMoM 2020, no. July, pp. 391–396, 2020, doi: 10.1109/WoWMoM49955.2020.00072.
[78] R. Doriguzzi-Corin and D. Siracusa, “FLAD: Adaptive Federated Learning for DDoS attack detection,” Comput. Secur., vol. 137, no. c, 2024, doi: 10.1016/j.cose.2023.103597.
[79] T. V. Phan and T. G. Nguyen, “FEAR: Federated Cyber-Attack Reaction in Distributed Software-Defined Networks with Deep Q-Network,” Wirel. Telecommun. Symp., vol. 2022-April, no. March 2022, 2022, doi: 10.1109/WTS53620.2022.9768169.
[80] N. T. Trong, H. Do Hoang, D. M. Trung, P. T. Duy, and V. H. Pham, “A federated threat hunting system with big data analysis for SDN-enabled networks,” Proc. - 2022 RIVF Int. Conf. Comput. Commun. Technol. RIVF 2022, pp. 35–40, 2022, doi: 10.1109/RIVF55975.2022.10013833.