Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-19679-7

A comprehensive analysis of threat vectors in software-defined networking

Seema Dahiya[1] · Harkesh Sehrawat[1] · Seema Kharb[2] · Vikas Siwach[1]

Received: 20 August 2023 / Revised: 7 June 2024 / Accepted: 10 June 2024

Abstract
In recent years, cyber-attacks have become more frequent and advanced, targeting critical infrastructure, businesses, homes, and government agencies. Detecting and preventing these attacks at the earliest stage possible is crucial to avoid serious harm, including data breaches. Researchers and experts in cybersecurity are looking to Software-Defined Networking (SDN) technologies as a solution to enhance real-time defense against cyber-attacks. SDN revolutionizes traditional networking by offering unprecedented flexibility and control over network resources, making it possible to adapt quickly to emerging threats. SDN provides logically centralized network control by separating the control plane from the data plane. This enables network programming and can block network activity when malicious movement is spotted. This paper presents a comprehensive analysis of threat vectors in SDN. It examines the various ways in which SDN networks are vulnerable to cyber-attacks, including network infrastructure, application layer, and SDN controller. The paper also evaluates the effectiveness of existing security measures and proposes future research directions to enhance SDN security. Overall, the paper highlights the potential of SDN as a powerful tool in the fight against cybercrime and emphasizes the importance of continued research and development to improve SDN security.

Keywords Internet of Things · SDN architecture · Traditional networking · Intrusion detection system · Threats

✉ Vikas Siwach
siwachvikas427@gmail.com

Seema Dahiya
semd14@gmail.com

Harkesh Sehrawat
sehrawat_harkesh@yahoo.com

Seema Kharb
seema016@gmail.com

[1] University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak, India

[2] Department of Computer Science and Engineering, SRM University, Delhi-NCR, Sonepat, India

Published online: 02 July 2024

1 Introduction

In the current digital world, networking systems and applications have become an essential component in everyday life for individuals and organizations. These systems are designed to meet various needs, such as communication, data storage, and processing [1]. As the number of users and applications on these systems increases, the need for efficient management and security also grows. One of the technologies that have been developed to address these concerns is Software-Defined Networking (SDN), which offers a new approach to network management. SDN is built on the principle of separating the control plane from the data plane. The control plane is responsible for managing network traffic, while the data plane handles the transmission of data packets. By separating the two, SDN provides a centralized control system that can effectively manage network traffic and provide better security. The benefits of SDN are numerous. For instance, it provides a more flexible and programmable network infrastructure that can adapt to changing network demands. SDN also allows for more efficient management of network traffic, which can result in faster data transfer and reduced latency. However, like any other technology, SDN is not without its challenges. Some of the challenges include scalability, reliability, and security. Since SDN relies on centralized software to make decisions regarding packet forwarding, it is vulnerable to security threats such as DDoS attacks, malware, and network intrusions [2]. To address these security concerns, researchers have explored various techniques that use the SDN paradigm to improve security. One of these techniques is Intrusion Detection Systems (IDS), which involves monitoring network traffic for signs of malicious activity. IDS can detect attacks in their early stages and alert network administrators, allowing them to take appropriate action to mitigate the attack [3]. This paper provides a comprehensive review of SDN architecture, comparing it to traditional networks and discussing its benefits and challenges. The paper also highlights SDN security threats and explores IDS technologies, methodologies, and approaches [4].

The main contributions of the paper are as follows:

- **Identification and Analysis of Important Datasets:**
  - Identifying and analyzing important datasets for threat, intrusion, or attack detection in SDN systems.
  - Emphasizing each dataset's concentration on particular attacks and the circumstances surrounding their creation.
- **Thorough Analysis of Performance Metrics:**
  - Providing a comprehensive analysis of the metrics employed to gauge the performance of intrusion detection systems in SDN.
- **Investigation of Immediate Threat Vectors:**
  - Investigating immediate threat vectors that present serious challenges for SDN security.
  - Offering suggestions for future research directions to address these threats.
- **Critical Review of Existing Systems:**
  - Identifying open research areas requiring attention by critically reviewing the design decisions made by existing SDN intrusion detection systems.

- **Advancement of Knowledge and Security Measures:**
  - Aiming to advance our knowledge and create stronger security measures in SDN.
  - Improving data protection methods in this rapidly evolving networking paradigm.

This research is motivated by the rising cases of Advanced Persistent Threats targeting business entities, homes, and even government installations. It is essential for these attacks to be identified and halted as early as possible to reduce or eliminate any possible consequences such as data loss. SDN technologies are being investigated as a potential strategy to improve real-time protection from cyber threats. SDN is an innovative concept in networking since it provides the customization and management of resources in a network to address new threats. SDN's potential of logically centralizing control through the separation of the control and data planes can allow more efficient and adaptive security measures needed against cybercrime. Consequently, the purpose of this research is to focus on strengthening security mechanisms within the context of the SDN paradigm to protect networks better.

The paper is organized into followings sections. Section 2 provides an overview of SDN, including its features, architecture, components, and advantages. Section 3 presents a systematic mapping of the SDN landscape, discussing the key concepts, methodologies, and techniques used in the field. Section 4 analyzes the various threat vectors that can affect SDN and the security implications associated with them. Section 5 presents a literature survey of the most recent research works related to SDN security. Section 6 discusses the challenges and issues facing SDN, including scalability, interoperability, and management complexity. Section 7 presents recent trends and observations in the SDN field, discussing the latest developments and emerging issues. Finally, Sect. 8 summarizes the conclusions drawn from the study and provides insights into the future research directions that can be explored in the field.

2 Overview of SDN

SDN is a type of network design that promises to make network management more straightforward by physically separating the control plane from the data plane. Because the control plane and data plane are so closely connected in traditional network topologies, it can be challenging to make changes to the network without negatively affecting user experience or traffic flow [5]. SDN solves this problem by centralizing control in a separate software component called the SDN controller. The SDN controller communicates with the switches in the network to program forwarding rules based on policies and network conditions. This allows network administrators to configure the network centrally and automate many tasks that would be difficult or impossible to accomplish with traditional networks. The data plane, the control plane, and the application layer are the three primary components that make up the architecture of a SDN network. The control plane is responsible for programming the rules that will be followed by the data plane when it comes to the forwarding of packets. The control plane is in charge of maintaining the network and programming the forwarding rules into the switches. Its responsibilities also include programming the routing protocols. The application layer is made up of the various network apps that are able to interface with the network. These applications run on top of the SDN controller.

Some of the key features and advantages of SDN include:

- Centralized control: SDN allows for centralized control of the network, making it easier to manage and automate network tasks.
- Programmability: SDN allows for flexible and dynamic network programming, enabling network administrators to easily modify the behavior of the network.
- Open standards: SDN is based on open standards and protocols, which promotes interoperability and avoids vendor lock-in.
- Virtualization: SDN enables network virtualization, allowing multiple logical networks to run on top of a physical network infrastructure.
- Scalability: SDN can scale to support large and complex networks, making it well-suited for enterprise and cloud environments.
- Security: By enabling more granular access control and network segmentation, SDN has the potential to significantly boost network security.

2.1 SDN architecture

The SDN architecture is a crucial component of this technology that has attracted a substantial amount of attention from researchers as well as industry practitioners. The control plane and the data plane are kept separate as shown in Fig. 1, which grants network managers the ability to manage network traffic in a more effective and versatile manner. This method of networking offers a consolidated view of the network, which, among other potential benefits, can serve to make network management easier and boost network safety. As the SDN technology continues to advance, there is a pressing need for a thorough analysis of its architecture, which should include an examination of its layers. This will allow for a better comprehension of the capabilities and restrictions of the technology.

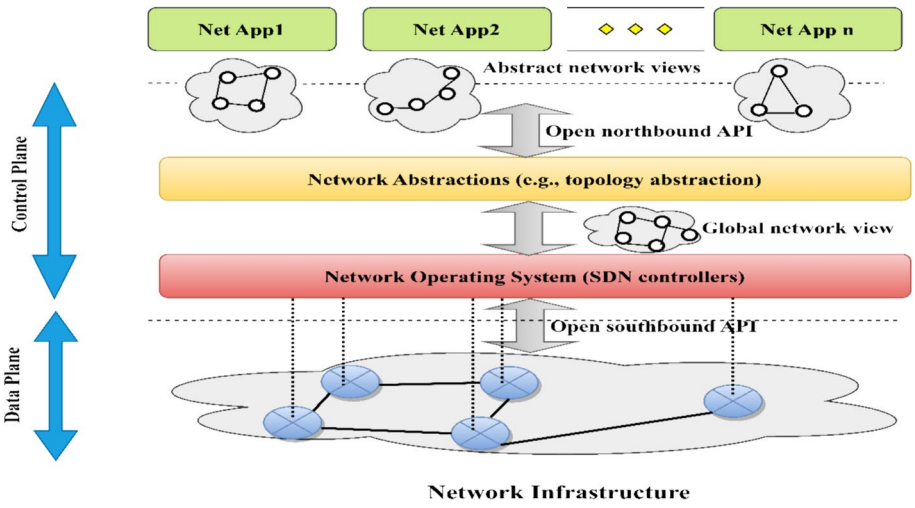

Fig. 1 SDN Architecture

A Infrastructure layer

The infrastructure layer is the bottom-most layer in the SDN architecture and is responsible for providing the physical network infrastructure, including routers, switches, and other network devices. This layer also provides the interfaces required to communicate with the underlying physical network infrastructure [6]. It is the foundation of the SDN architecture and enables the separation of the control and data planes. At this layer, traditional networking devices are replaced by OpenFlow-enabled switches that can be centrally managed by the SDN controller. The OpenFlow protocol provides the means for the controller to communicate with the underlying network devices to manage traffic flow, which allows for centralized control and management of the network. Additionally, the infrastructure layer provides interfaces for the management of network devices, including device configuration, monitoring, and status reporting. One of the key benefits of the infrastructure layer is the ability to simplify network management and increase network flexibility. With SDN, the network administrator can easily change network policies, traffic routing, and other network settings without having to manually configure individual network devices. The infrastructure layer also enables the dynamic provisioning of network resources, allowing for the allocation of resources on an as-needed basis. This flexibility and agility make the SDN infrastructure layer an essential component of modern networking architectures. However, the infrastructure layer also presents several challenges. First, SDN requires significant investments in new hardware and software infrastructure to enable OpenFlow-enabled switches and SDN controllers. Additionally, network administrators must be trained to manage the new SDN infrastructure, which may require new skills and knowledge. Finally, SDN infrastructure must be carefully planned and designed to ensure that it is scalable, secure, and reliable. Overall, the infrastructure layer is critical to the success of the SDN architecture and requires careful consideration in the design and implementation of SDN networks.

B Control layer

The Control Layer is responsible for the centralization of network control, which is a fundamental feature of the SDN architecture. It comprises the components that manage the network topology, traffic forwarding policies, and network configuration. The primary element of the Control Layer is the SDN controller, which functions as the network operating system. It runs a software application responsible for managing and forwarding network traffic based on predefined policies. The SDN controller provides a logical view of the network to the applications running on top of it. The controller communicates with the switches in the Infrastructure Layer through a standardized protocol, such as OpenFlow. The Control Layer also includes the network applications that run on top of the SDN controller. These applications interact with the controller to implement network services and policies. Examples of SDN applications include load balancers, firewalls, intrusion detection systems, and traffic engineering modules. One of the significant advantages of the Control Layer is the separation of the control plane from the data plane. This separation allows for centralizing network management, which provides better network programmability and automation. It also enables more granular control of network traffic, which enhances network security and resiliency. The Control Layer is also responsible for ensuring network scalability and reliability. SDN controllers can be deployed in a distributed manner, allowing for redundant and fault-tolerant configurations. Additionally, the Control Layer provides mechanisms for handling network failures and recovering from them.

C Application layer

The Application Layer is the very top layer of the software-defined networking architecture, and its job is to supply services to the applications that run on networks. It is made up of a collection of application programming interfaces, or APIs, which make it possible for network applications to connect with the SDN controller in order to program the network. Network applications are able to function on an abstracted view of the network topology thanks to the Application Layer, which offers a centralized view of the network and exposes this view to users. At the Application Layer, the SDN controller exposes its northbound application-programming interface (API) to the various network applications. The northbound application-programming interface (API) is a collection of interfaces that are made available to network applications so that they can communicate with the controller and program the network. Because the northbound application-programming interface (API) is vendor-neutral by design, it enables the creation of network applications that are not dependent on the underlying network hardware.

Network applications at the Application Layer can be used to implement a variety of network functions, including load balancing, traffic engineering, intrusion detection, and firewalling. These applications can be developed by third-party developers, making it possible to extend the functionality of the SDN infrastructure beyond what are provided by the controller vendor. Overall, the Application Layer of the SDN architecture provides a flexible and extensible platform for network application development, enabling the creation of innovative and customized network services.

2.2 SDN features

SDN has emerged as a new approach to designing and managing networks. It offers several unique features that set it apart from traditional networking. Firstly, SDN decouples the control plane from the data plane. This means that network devices are no longer responsible for making complex routing decisions; instead, a centralized controller handles this task [7]. This simplifies network management, allowing for greater flexibility and control over network behavior. Secondly, SDN relies on flows, rather than source and destination addresses, to make forwarding decisions. A flow is a collection of packets that share common characteristics, such as protocol type, source and destination port numbers, and time interval [8]. By basing forwarding decisions on flows, rather than individual packets, SDN can achieve more efficient and effective network utilization. Thirdly, SDN enables a global network perspective and a software encapsulation of network logic. By moving the control logic to a third-party entity, such as a network operating system or an SDN controller, SDN allows for greater network programmability and automation. This enables network operators to write software applications that can communicate with the underlying data layer hardware, and automate network management tasks, such as traffic engineering, load balancing, and security. Finally, SDN allows networks to be programmed using software applications that communicate with the underlying data layer hardware that runs on top of the network operating system [9]. This software-defined approach to network management provides greater agility and flexibility, enabling operators to quickly and easily adapt to changing network conditions and user needs. Additionally, the network can be set up with SDN's software/hardware links, allowing forwarding devices to be changed to execute applications, further increasing the network's flexibility and adaptability.

2.3 Software defined networking controller

In software-defined networking (SDN), the controller serves as the "brain" of the network, as it determines the routing path for each new flow. Without the controller, the network would be vulnerable to attacks and potential destruction. There are different types of controllers that can manage the control plane, including a single controller or multiple controllers as shown in Fig. 2. However, the architecture of the controller [10] [11]greatly impacts the performance of the SDN. Recently, three new controller architectures were introduced, which include distributed, multi-core, and logically centralized controllers. Distributed controllers synchronize their conditions with other controllers to produce an optimal global solution and maintain a comprehensive network picture. On the other hand, logically centralized controllers are composed of multiple distributed controllers, enabling them to respond more quickly to each flow request and handle more requests per second. Although exchanging data between controllers using network services is necessary, it can also create a heavy load on the network. To improve scalability and maintain data consistency, efforts must be made to lessen the load of state synchronization between controllers.

SDN controllers take the shape of certain traits, functions, and applications. In order to give a clear understanding of the distinctions between these controllers and how they might be used to different networking contexts, following are the some of the most well-known SDN controllers.

Open Daylight Controller: In brief, Open Daylight is a popular open-source SDN controller framework that supports a wide range of protocols, including Netconf and OpenFlow. This offers a framework that is modular, scalable, and adaptable for creating and implementing network management applications.

Controller Floodlight: Written in Java and released under the Apache license, Floodlight is an enterprise-class OpenFlow controller. Because of its portable and lightweight architecture, it may be deployed on small and medium-sized networks or utilized as a teaching tool for novices in SDN.

Open Network Operating System (ONOS): An excellent option for carrier-grade and mission-critical networks, the controller platform's outstanding performance and scalability provide high availability. It prioritizes scalability, performance, and compatibility with multiple protocols and services.

Ryu Controller: Ryu is a framework for SDN controllers that provides easy-to-use and adaptable APIs for creating new network control and management applications. It is highly

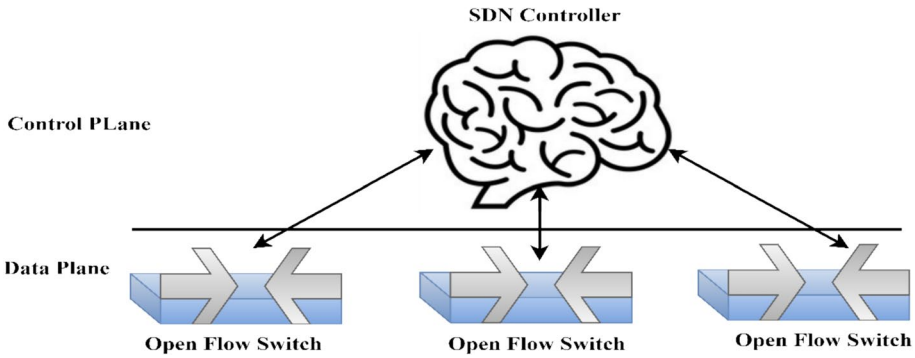

Fig. 2 Controller as Brain of SDN

perfect for researchers and developers who wish to experiment with novel SDN applications because it is versatile and easy to use.

Beacon Controller: Beacon is a lightweight, easy-to-use SDN controller and was among the first. Despite having less features than other controllers, it is still thought to be the best platform for studying or doing lab experiments on the fundamentals of SDN.

Each of these controllers has a unique set of benefits and may be used with various networks that have distinct needs. For instance, because Open Daylight and ONOS may integrate so many features and growth, they are ideally suited for extremely sophisticated large-scale networks. However, because they are easier to use and less complicated, Floodlight and Ryu might be favored over ONOS and Open Daylight for smaller projects or when someone wishes to demonstrate something for educational purposes.

2.4 Benefits of SDN framework

- **Operational Saving**: SDNs cut operational costs by automating configuration and management activities, which frees up network administrators to focus on other priorities. Application owners receive pre-packaged network services, which frees up the networking team to focus on other tasks.
- **Flexibility**: SDNs make it possible for the network to be utilized and operated in a number of different manners. Resellers have the ability to design their very own network services by utilizing the standard development tools.
- **Increased Uptime**: SDNs make it possible for resellers to prevent configuration and deployment problems, which could potentially interrupt the network, by doing away with the need for manual involvement.
- **Improved Management:** Managed Service Providers (MSPs), also known as cloud service providers, are able to manage virtual networking, computing, and storage resources using a unified point of view and toolkit.
- **Planning**: With a more complete understanding of their customers' network, compute, and storage capabilities, resellers are better able to plan IT efforts on their clients' behalf.
- **Infrastructure Cost Savings:** By decoupling, route/switching knowledge from packet forwarding, routers and switches can compete on price-performance attributes.

3 Systematic mapping

As a part of a systematic mapping study (SMS), which assisted in the development of the research project, an assessment of the level of security that is now implemented in SDN was carried out [12]. Since 2007, its use in computers, and more specifically in the field of software engineering, has seen tremendous expansion. A classification process, a graphical summary, mapping, and outcomes are all provided by a systematic mapping study, as stated by Petersen [13]. The literature on software-defined networks is evaluated and analyzed in this study, which looks at published articles between the years 2010 and the second quarter of 2022. A strategy that is utilized to discover, investigate, and gather pertinent data from the body of work that is related to the study subject is called a literature review. This particular literature study was broken up into two sections, which are depicted in Fig. 3. Phase selects the electronic database to which a search will be applied as well as the search phrases that will be used in the search. This allows Phase to build an initial list of articles.

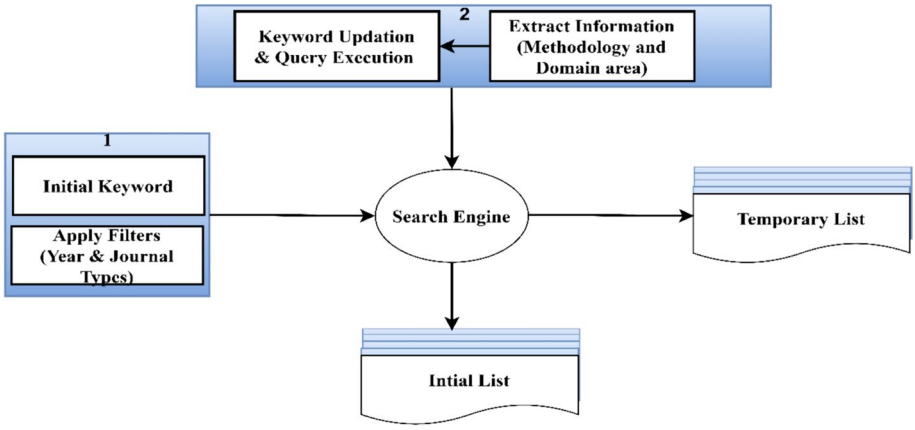

Fig. 3 Phases of Literature Review

3.1 Initial search terms

In this phase, the goal is to select appropriate keywords for our literature review. We will use the Boolean operator "AND" and the terms "Software-Defined Network" OR "SDN" OR "SDNs" combined with the terms "SDN Architecture," "SDN Controller," "Security," "Threats," or "Vulnerabilities" to create our search string.

3.2 Sources selected

In our study, we examined scholarly journal articles and research papers. We reviewed several citations from the downloaded articles and added additional relevant publications. The number of citations varied across different websites. We primarily focused on archiving sites such as IEEE ACCESS, IEEE EXPLORE, ACM Digital Library, Google Scholar, Research Gates, and Science Direct.

The inclusion and exclusion criteria for the literature review are summarized in Table 1.

Table 1 Literature Review

Criteria	Rational
Include 1. The preliminary research ought to offer a programmable network SDN solution	The SDN is the area
Include 2. Publication Period	2010 to 2022
Include 3. Attack zones in SDN must be described in the preliminary study	The study of this requires both academic and professional classifications
Exclude 1. Language	The publication that did not publish in English is removed
Exclude 2. Between 2010 and 2022, the study was never published	Following a full-text review, the papers will be rejected. If the emphasis is not on SDN

3.3 Research questions (RQ)

The research questions listed in Table 2 are taken into consideration.

Research Q1 reply can be found in Fig. 4. Despite this, research associated with software-defined networks (SDN) didn't begin until 1990; nevertheless, the pace of SDN research has stepped up significantly after 2017, when the Internet Research Task Force (IRTF) created the SDNRG (Programming Characterized Organization Exploration Gathering).

Research Q2 focuses on a well-known and significant author who published studies that were related to SDN. Even though many authors have contributed to such a subject, the authors who have had at least ten distributions are few in number and are included in Tables 3 and 4. These authors are listed in Table 3.

Table 2 Research Question

Research Q1	How are the distributions connected with SDN conveyed throughout the long term?
Research Q2	Who are the main authors who contributed?
Research Q3	Which prominent SDN conferences are there?
Research Q4	Which top Institutions publishing about SDN?
Research Q5	How SDN is different from traditional networking?
Research Q6	Which are the most used Datasets Specific to SDN Security?
Research Q7	What are Evaluation Metrics used in designing IDS in SDN?

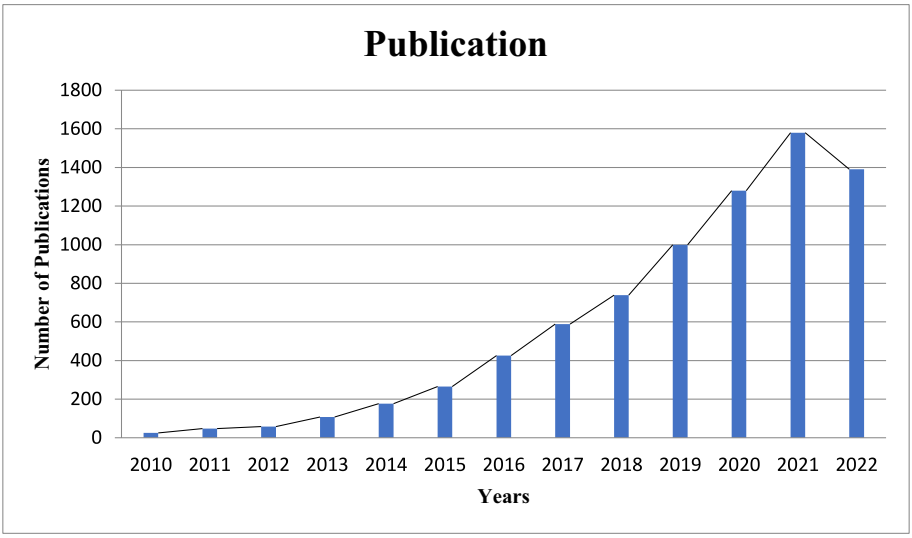

Fig. 4 Time Based Count of SDN Publication

Table 3 Top Authors in SDN (2010–2023)

Authors	Count
R. Martinez	125
R. Munoz	124
R. Casellas	121
M. Yu	65
A. Guha	27
M. Canini	25
Vassilios G. Vassilakis	21
J. Rexford	20
S. Shenker	17
N. McKeown	12

Table 4 Comparisons of SDN and Conventional Networking

Characteristics	SDN	Conventional Network
Network Control	Centralized	Distributed
Interface	Open	Close
Cost	Low	High
Programmability	Present	Absent
Flexibility of Network	Present	Absent
Structural Complexity	Low	High
Complex Control Network	Absent	Present
Extensibility	High	Low
Performance Improved	Present	Absent
Configuration	Automatic	Static/Manual
Management Enhanced	Present	Absent
Configuration Efficiency	Present	Absent
Easy to use and implement	Present	Absent
Troubleshooting and Reporting	Easy	Difficult

Research Q3 is associated with Important Gatherings That Distribute Papers Related to SDN in Figs. 5 and 6.
Research Q4 focuses on the involvement of top institutions worldwide in the development of SDN, as demonstrated in Fig. 5.
In **Research Q5**, the focus is on differentiating between conventional networking and software-defined networking.

Figure 7 represents the architecture of conventional networking, while Fig. 8 displays the planes and switches of SDN architecture. In Table 4, we compared the differences between conventional networking and SDN based on sources. Conventional networking refers to an outdated approach that uses specialized hardware devices like switches and routers to manage network traffic. In contrast, SDN has gained popularity as it offers better

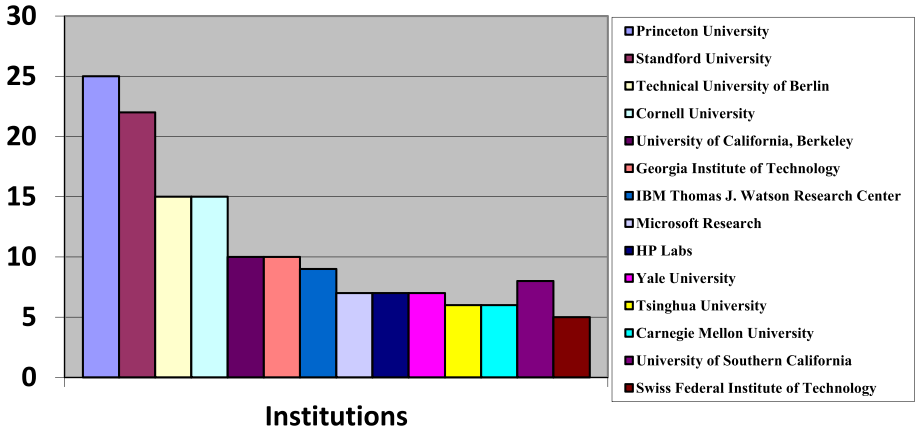

Fig. 5 Top Institutions publishing in SDN

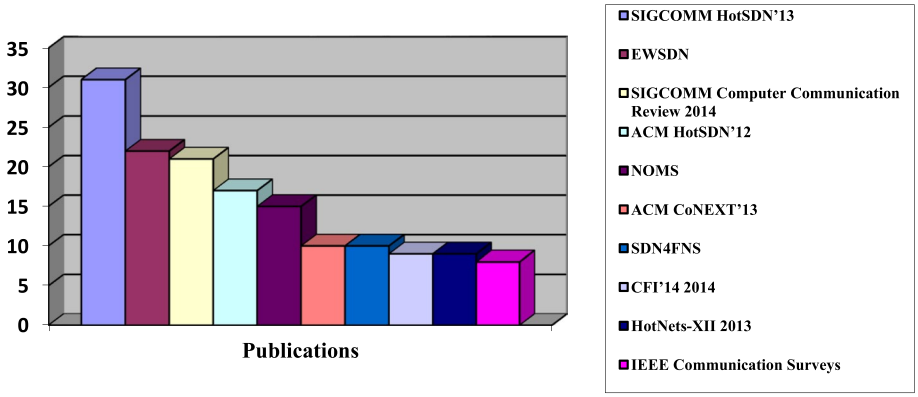

Fig. 6 Top SDN Conferences

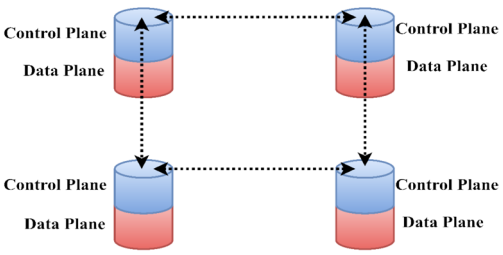

Fig. 7 Conventional Network Architecture Planes/Devices

scalability, performance, and security features, making it a suitable solution for modern business environments [5]. Unlike conventional networks, SDN architecture is flexible and dynamic, allowing for easier management of a growing number of devices. Conventional networks require manual reconfiguration of network devices when new devices are added, which can lead to scalability and security concerns. Additionally, traditional network

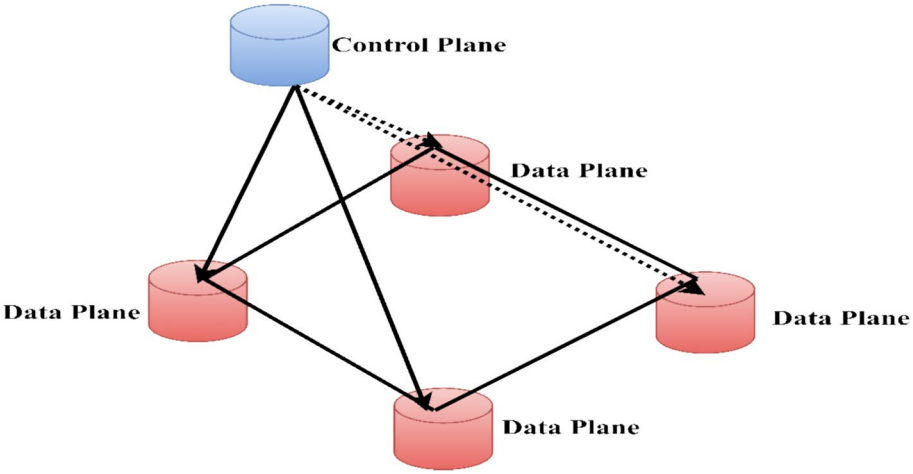

Fig. 8 SDN Architecture Devices/Planes

infrastructures require human controllers to manage the increasing number of devices, leading to higher operating expenses.

As networks grow larger, it becomes increasingly challenging humans to install and maintain switches, routers, and other network equipment. This results in scalability and security issues, as manual configurations can be prone to human errors.

Research Q6 focused on the datasets that utilized to evaluate the effectiveness of the proposed methodology, and Table 5 provided a detailed overview of the datasets used, along with the threat types contained in each dataset.

3.4 KDD cup'99

This IDS dataset is widely recognized and frequently used for evaluating intrusion detection systems. It comprises approximately five million records, with an equal number used for testing and training purposes. Each record contains 41 distinct attributes or features, which help to identify whether the record represents an attack or a legitimate network activity.

Table 5 Summary of Datasets available for attack detection

Dataset	Reference	Year	Threats
KDD Cup 99	[14]	1998	DoS, R2L, Probe, U2R
Kyoto 2006	[15]	2006	Attacks that are known, Attacks that are unknown
NSL-KDD	[16]	2009	R2L, Probe, U2R, DoS
UNSW-NB15	[17]	2015	Exploits, Fuzzers, Port scans, Reconnaissance, Backdoors, Shellcode, worms, DoS, Generic
CIC-IDS2017	[18]	2017	Infiltration, HeartBleed, Brute Force, Botnet, DoS, DDoS, Web
CSE-CIC-IDS2018	[18]	2018	HeartBleed, DoS, Botnet, DDoS, Brute Force, Infiltration, Web

3.5 Kyoto 2006 +

The dataset compiled by Kyoto University, which utilized network security technologies to gather network traffic records over a period of nine years from 2006 to 2015. The most recent version of the dataset contains records from this time period. Each record in the dataset contains 24 statistical attributes, of which 14 are sourced directly from the KDD Cup'99 dataset. The remaining 10 attributes are additional features.

3.6 NSL-KDD

The KDD Cup'99 dataset has improved by removing certain fundamental issues. The dataset includes 41 attributes that are used to describe various types of attacks. According to KDD Cup'99, these attributes can be categorized into four distinct types.

3.7 UNSW-NB15

The UNSW-NB15 dataset, which was generated by the Australian Center for Cyber Security, consists of over two million records, each of which has a total of 49 parameters.

3.8 CIC-IDS2017

The CIC-IDS2017 dataset developed by the Canadian Institute of Cyber Security (CIC) in 2017 and contains both real-world attacks and expected flows. The dataset is designed to be used for intrusion detection research. The network traffic in the dataset is analyzed by CICFlowMeter, which uses information such as timestamps, source and destination IP addresses, protocols, and attacks to identify and classify network traffic.

3.9 CSE-CIC-IDS2018

The Canadian Institute of Cyber Security (CIC) and the Communications Security Establishment (CSE) worked together in 2018 to produce the dataset. This was accomplished through a collaborative effort.

> **Research Q7** discusses the evaluation of AI algorithms for intrusion detection systems, which is assessed using common evaluation metrics provided in Table 6. These evaluation measures are based on the various attributes employed in the algorithms.

Several typical assessment metrics, such as the true positive (TP) rate, the true negative (TN) rate, the false positive (FP) rate, and the false negative (FN) rate are used in the process of evaluating artificial intelligence algorithms for use in intrusion detection systems. The term "TP rate" refers to the proportion of actual positive events that the algorithm properly identifies as positive. The TN rate is the proportion of actual negative instances that the algorithm properly identifies as being negative. It is also abbreviated as "TN rate." The false positive rate (FP rate) is the proportion of genuine negative cases that are wrongly identified as positive, while the false negative rate (FN rate) is the proportion

Table 6 Evaluation Metrics

Evaluation Metrics	Formula	Description
Precision	Precision = TP/ TP + FP	It is the proportion of accurately anticipated assaults to every one of the examples anticipated as attacks
Recall	Recall = Detection Rate = TP/ TP + FN	It is a proportion of all examples accurately delegated assaults to every one of those genuine attacks
False alarm rate	False Alarm Rate = FP /FP + TN	It is defined as the proportion of incorrectly expected attack tests to all usual cases
True negative rate	True Negative Rate = TN /TN + FP	It is characterized as the proportion of the quantity of accurately ordered typical examples to every one of the examples that are ordinary
Accuracy	Accuracy = TP + TN /TP + TN + FP + FN	It is the proportion of correctly ordered occurrences to the actual quantity. It is also known as Discovery Precision and is an important presentation metric when a dataset is balanced
F-Measure	F Measure = 2 (Precision × Recall/ Precision + Recall)	It is characterized as the harmonic mean of the Precision and Recall

of actual positive instances that are incorrectly identified as negative. These metrics are necessary for determining whether or not AI algorithms are successful in recognizing and locating potential dangers to a network's security.

4 SDN threats

Each component or layer that is a part of SDN has the potential to result in either purposeful or accidental abuse. On occasion, it was utilized to bring attention to faults in the system or expose previously hidden malevolent actions. As a result, each and every SDN component or layer that was a part of the architecture suggested a potential danger vector or attack surface. Attacks were appropriately categorized according to their effects on network policies, application architecture, and practice architecture. This was possible due to the fact that the SDN design partitions network policy definitions from functionality that is generated from technologies and practices [19]. A document published by the Open Networking Foundation with the title Standards and Practices for Secure Software-Defined Networks has a set of proposed recommendations for the protection of SDN. The security requirements for the SDN architecture are outlined in Table 7, which covers all of its protocols, components, and interface points. The controller, which serves as the system's central location for management, is a potential entry point for an assault on the SDN [20]. Information pertaining to switching, routing, and access control is contained within the flow tables of the switches. By fooling the controller into allowing malicious services to join the network and communicate with the controller as well as the web and its traffic, it is also possible to attack the North–South connection-point, the South-South point of interaction, and the East–West point of interaction. A pertinent real-world instance of an SDN threat is the 2016 cyber-attack on the domain name system provider Dyn. This Distributed Denial of Service (DDoS) attack was amplified through a botnet comprising numerous IoT devices, which exploited the centralized nature of SDN controllers in the network. Attackers orchestrated a synchronized flood of requests, overwhelming the system and disrupting services for major platforms across the internet.

The security weaknesses and risks of the SDN network architecture are illustrated in Fig. 9, which indicates several vulnerabilities and potential breaches across different layers and interfaces. Despite the fact that SDN technology is intended to provide secure network operations, topology, flow control, and access mechanisms, this section highlights its security shortcomings and hazards.

Table 7 Principles for Securing SDN

1	Characterize security reliance and trust limits obviously
2	Ensure Strong Entity
3	Build security using open standards
4	Safeguard data Availability, Integrity and Confidentiality
5	Secure Functional Reference Information
6	Establish secure system by default
7	Security must Support Responsibility and Detect ability
8	Characteristics of Reasonable Security Controls

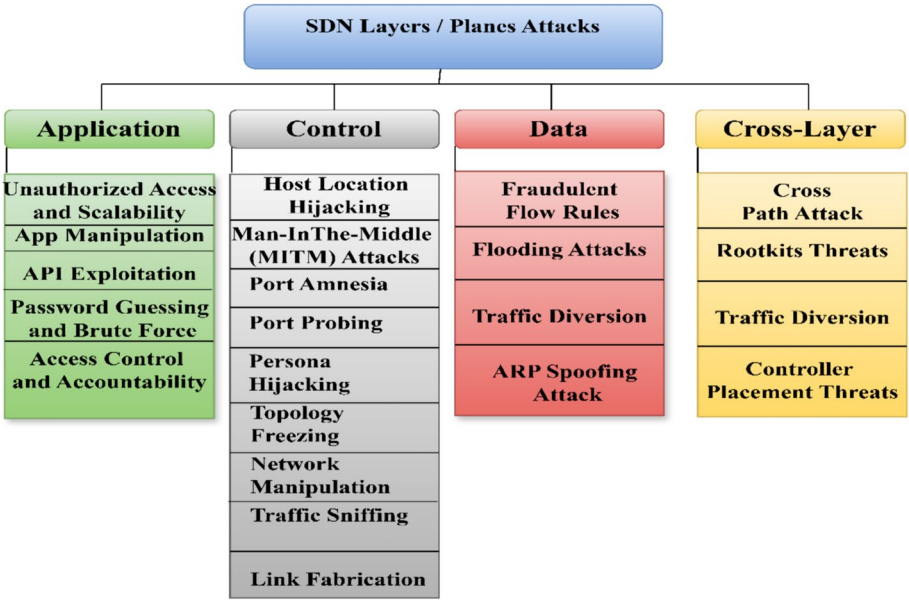

Fig. 9 SDN Layers Attacks

4.1 Application layer/plane attacks

4.1.1 Application manipulation

One of the security risks in SDN architecture is the possibility of application manipulation attacks in the SDN application plane. These attacks enable attackers to gain access to an SDN application, allowing them to cause issues and disrupt protocols. Furthermore, attackers may obtain more rights to exploit SDN applications, increasing the potential damage they can inflict.

4.1.2 API exploitation

API exploitation is another type of attack that attackers may employ by exploiting the Application Programming Interface (API) of specific software components connected to SDN systems, which can expose sensitive information without proper authorization. This may lead to a halt in the network's information flow. To mitigate such threats, it is crucial to keep the software programs running on SDN nodes up-to-date.

4.1.3 Password guessing and brute force

A brute force attack involves an attacker attempting to gain access to a system by trying out a wide range of user credentials, including all possible usernames, passwords, and combination variations. This process of trial and error can lead to the attacker successfully guessing the password and gaining unauthorized access to the system.

4.1.4 Access control and accountability

The SDN controller has the capacity to implement a diverse selection of applications in order to improve the efficiency with which the network and SDN services are utilized. On the other hand, these applications are given considerable access privileges which leaves the entire SDN network open to the possibility of security breaches. Therefore, in order to guarantee the safety of the network environment, it is essential to develop methods of stringent access control and authority implementation on these applications. The security flaws that are related with the power and accessibility control of SDN are broken down in Fig. 10, which provides a better understanding of these flaws. Unauthorized and potentially harmful apps may gain access to the control layer of the SDN through a gateway that is opened by a hacked SDN application.

4.2 Control layer/plane attacks

4.2.1 Host location hijacking

This type of attack is known as resource depletion, and it targets the resources of the SDN controller that are used to control the network's operations. In particular, this type of attack can be devastating when controlling critical infrastructure such as aircraft via data. Attackers may use the controller's resources to slow down the entire network or render it unreachable to end users as shown in Fig. 11.

4.2.2 Link fabrication

An attacker can add a spoofed connection to the SDN network to gain control over traffic, which can be a challenging task to detect due to the distributed nature of the network and scalability issues inherent in SDN. As a result, finding the origin of a link fabrication attack can be difficult.

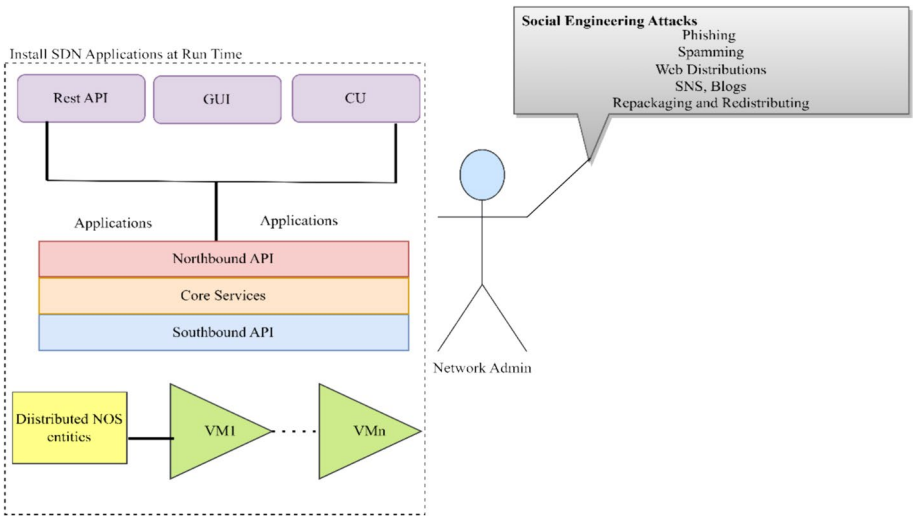

Fig. 10 SDN threatening situation

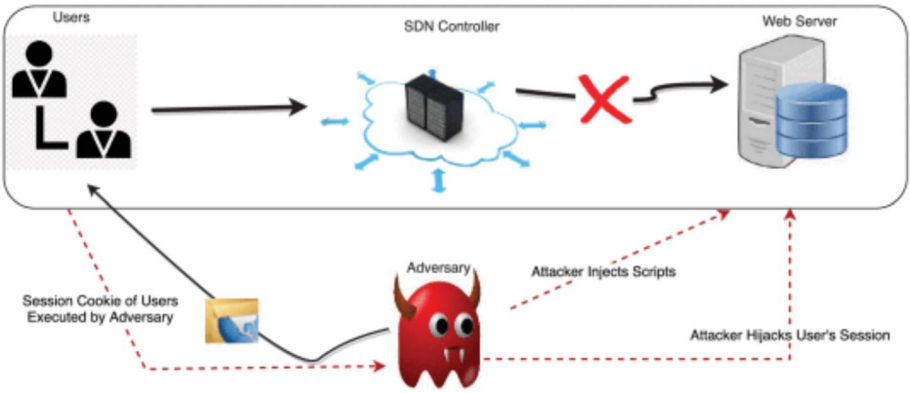

Fig. 11 Host location-based hijacking attack

4.2.3 Port amnesia

To prevent attacks based on port amnesia, various defense techniques are being developed for SDN networks. One such technique is the Port-Security feature, which helps to prevent unauthorized access to a network by limiting the number of MAC addresses allowed to communicate through a specific switch port. Another approach is to use dynamic access control, which uses a combination of port-based access control and user identity to prevent unauthorized access. Additionally, SDN controllers can use network monitoring and anomaly detection techniques to detect and block malicious traffic. These defense techniques help to ensure the integrity and security of SDN networks against topology attacks.

4.2.4 Port probing

This attack on the SDN architecture enables attackers to bypass defenses and also causes confusion in the location of the host, resulting in attempts to hijack the host.

4.2.5 Persona hijacking

It is a type of attack on the SDN architecture that can have severe consequences. This attack exploits the bindings of the layers in the SDN network and aims to deceive the SDN architecture into believing that the attacker is the legitimate owner of the network. By doing so, the attacker gains unauthorized access and permissions to the SDN network.

4.2.6 Reverse loop

A reverse loop attack is a type of network attack in which an attacker exploits the dynamic nature of the SDN by creating a reverse link that disrupts the network. By creating a fake link that connects two switches, the attacker tricks the SDN controller into believing that there is a valid connection between the switches. The controller may then redirect traffic through this fake link, causing network disruptions and potentially leading to a denial of service. This attack is accomplished by reversing the inter-switch links within a predetermined timeframe, thereby disrupting the network and potentially causing widespread damage.

4.2.7 Topology freezing

Topology freezing is a type of attack that exploits vulnerabilities in the topology service provisioning modules of the SDN controller. This attack affects how the controller views the network and prevents it from updating its topology view to reflect the dynamic changes in the network. As a result, the attacker can hide their malicious activities in the network and make it difficult for the controller to detect them. This can cause disruption in the network's performance and can also compromise the security of the network.

4.2.8 Network manipulation

This attack on SDN occurs in the control plane, where the attacker compromises the controller and introduces false information into the network. This type of attack aims to launch further attacks across the entire communication system [21]. By manipulating the network's control plane, the attacker can cause disruption in the flow of data and potentially gain access to sensitive information. It is essential to implement effective security measures to prevent such attacks, such as using secure protocols and implementing access control mechanisms.

4.2.9 Traffic sniffing

It is a type of attack in SDN that involves intercepting network traffic to obtain sensitive data by capturing and analyzing the communication interface. This type of attack is particularly effective in networks that have consistent traffic and can provide attackers with access to valuable information. Attackers who gain access through traffic sniffing can exploit network circumstances and use unencrypted data to prohibit traffic to and from the controller. To mitigate the effects of traffic sniffing attacks in SDN, strong encryption techniques and passwords should be employed to secure the network [21]. This helps to ensure that sensitive information is not transmitted in plaintext, making it more difficult for attackers to intercept and use the data. Additionally, network administrators can implement monitoring tools and intrusion detection systems that can alert them to any suspicious traffic patterns and help to identify potential attacks.

4.2.10 Man-in-middle (MIM) attacks

This attack in Open Flow systems can have more significant consequences compared to those in traditional networking setups, primarily due to the absence of identity verification controls in the Open Flow TCP/IP control layer. As depicted in Fig. 12, attackers can leverage downstream Open v Switch switching devices to carry out sophisticated eavesdropping attacks in an SDN system. These attacks involve intercepting network communication between two parties and potentially altering or stealing sensitive data. Therefore, it is crucial to implement robust identity verification measures in the Open Flow control layer to mitigate the risk of MIM attacks.

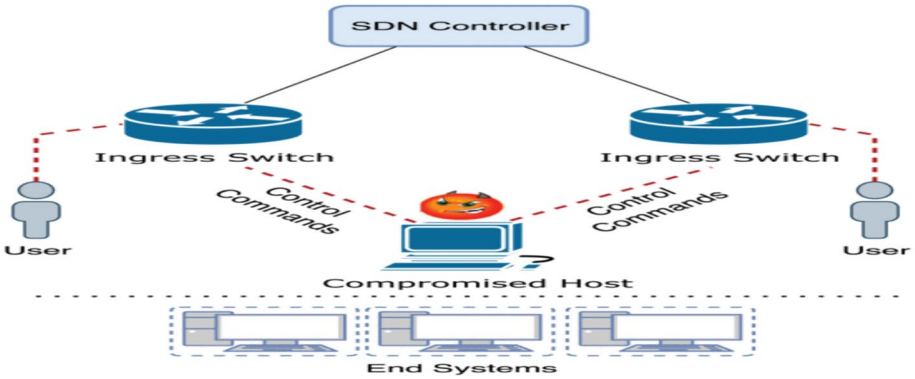

Fig. 12 Man-in-Middle Attack

4.3 Data layer/plane attacks

4.3.1 Fraudulent flow rules

In SDN networks, attackers can launch fraudulent flow rules attacks by impersonating the SDN controller and inserting invalid or malicious flow rules into OpenFlow switches. This type of attack can cause network disruptions or allow attackers to divert network traffic to malicious destinations. The attackers can also modify the existing flow rules in order to gain access to sensitive information, such as user credentials or other sensitive data. In order to prevent fraudulent flow rules attacks, SDN administrators can implement strict access control policies and ensure that flow rules are properly authenticated before they are inserted into the network.

4.3.2 Flooding attacks

These attacks are a common type of Distributed Denial of Service (DDoS) and Denial of Service (DoS) attacks in SDN-based communication systems. These attacks take advantage of the self-management and diverse access to network operation that are provided by SDN architectures. When a network packet violates the OpenFlow flow rules, the OpenFlow switch still sends it to the SDN controller. This can be seen in Fig. 13. Protocol rules are unsafe, even if data is decoupled from the control plane. Adversaries can use this vulnerability to interfere with data plane forwarding and network topology, both of which are crucial to the proper operation of SDN setups.

4.3.3 Traffic diversion

It is a security attack on the data plane of SDN where attackers can exploit vulnerabilities in network devices to divert network traffic, which allows them to intercept and manipulate data transmissions. By manipulating network components, attackers can redirect traffic

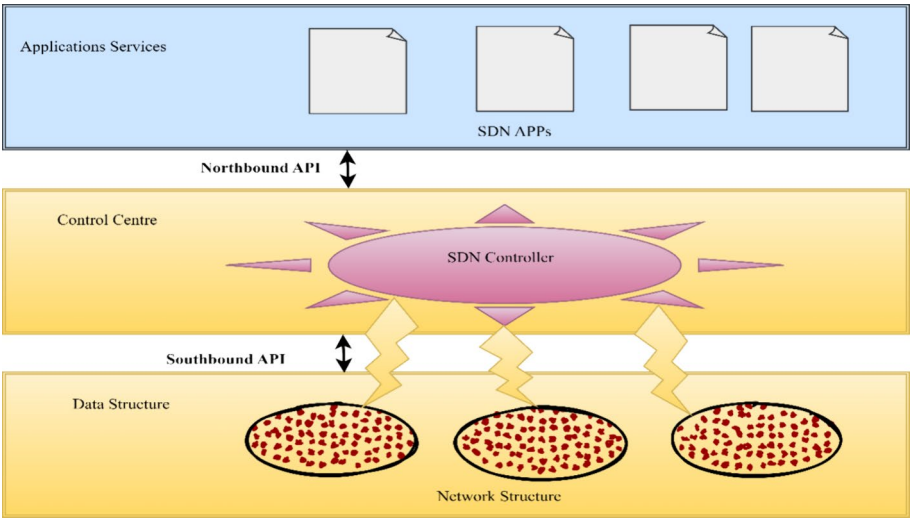

Fig. 13 DoS Threat

flows to their own systems where they can snoop on the data or reroute them to other destinations. This type of attack can result in data breaches, unauthorized access, and a loss of network availability. To prevent traffic diversion attacks, it is essential to secure network devices and use encryption techniques to protect the integrity and confidentiality of network traffic.

4.3.4 ARP spoofing

In local area networks (LANs), it takes advantage of a protocol known as Address Resolution Protocol (ARP). On a network, mapping an IP address to a physical MAC address is accomplished through the use of the ARP protocol. An attacker uses an ARP spoofing attack to associate their MAC address with the IP address of a genuine host on the network. This is accomplished by sending bogus ARP signals. This makes it possible for the attacker to intercept and control network traffic that is destined for that host, which opens the door to the possibility of Man-in-the-Middle attacks (MITM). Attacks using a spoofing technique called ARP are mainly restricted to being carried out on local network segments.

4.3.5 Side channel attacks

Side channel attacks are a type of security breach that exploits weaknesses in cryptographic systems that lack strong mathematical features. These attacks are usually non-invasive and do not cause any harm. In SDN, side channel attacks take advantage of information leaked from malicious nodes during regular communication activities to extract data. They often rely on network delays to infer network configurations. As time goes on, these attacks can become more efficient. To mitigate these threats, it is important to implement dynamic workload adjustments in the control plane.

4.4 Cross-layer attacks

4.4.1 Cross path attack

A cross-path attack refers to an attack on the control channels of an SDN where the attacker exploits shared routes between data and control signal intersections. The attacker can use manipulated information to disrupt control traffic, particularly shared connections. This attack is difficult to detect because the attacker can efficiently mix the manipulated information with the control channel signals, making it difficult for the controller to distinguish between legitimate and malicious traffic. The attacker can also exploit vulnerabilities in the SDN routing protocols to reroute traffic through malicious paths, bypassing security measures and increasing the likelihood of a successful attack. To prevent cross-path attacks, SDN networks need to implement secure routing protocols and establish separate channels for data and control traffic to avoid the sharing of routes.

4.4.2 Teleportation attacks

It takes advantage of the separation between the control and data planes in SDN. This separation creates a reliable and configurable connection, which makes SDN a promising paradigm for various types of attacks, including attacks on switches and hosts.

4.4.3 Rootkit attack

A rootkit is a malicious software program that can provide an attacker with complete control over a system. In the context of a software-defined network, rootkits can be used to infiltrate and gain control over multiple controllers, posing a significant threat to the system's security.

4.4.4 Controller placement threats

In an SDN system, the controller is responsible for implementing security policies. However, improper or incompatible configuration of multiple controllers in single-domain and multi-domain scenarios can result in internal conflicts. There is no assurance that all installed SDN controllers will be aware of network modifications or upgrades and the network's condition and resources. SDN controllers divide the overall network environment into multiple controllers, resulting in the creation of sub-networking domains. This can make it challenging or even impossible to maintain and implement security protocols and applications in each separate SDN domain, leading to Controller Placement Threats.

5 Intrusion detection system

An IDS is a security tool that monitors network traffic from outside an organization. The IDS functions as both a management console and a sensor. When the sensors detect an attack signature that matches a previously identified attack, they send an alert to the control center. IDS can detect a wide range of security threats, including spyware, critical logs, unintentional data leaks, security policy violations, unauthorized users and servers, and even configuration issues. A recent survey by the Barkley Organization and the EEF's

2018 Cyber Security for Manufacturing Report found that nearly 48% of manufacturers have been victims of cyber attacks [19]. The National Centre for Manufacturing Services (NCMS) in the United States has reported that the cost of such breaches in the manufacturing industry ranges from USD 1 million to USD 10 million. Consequently, companies have adopted improved security measures, driving the demand for intrusion detection and prevention systems. These systems analyze network traffic to identify intrusions perpetrated by unauthorized entities.

5.1 Active IDS

An Active Intrusion Detection System (IDS) is a type of IDS that can take action in response to detected intrusions or attacks. Unlike a passive IDS, which only detects and reports on intrusions, an active IDS can take measures to prevent or stop an attack in progress.

Active IDS can be classified into two main categories: reactive and proactive. Reactive active IDS responds to an intrusion or attack after it has been detected, while proactive active IDS takes preemptive measures to prevent attacks before they happen. Some common examples of active IDS actions include blocking traffic from a suspicious source, terminating a suspicious connection, and resetting connections. Active IDS may also use various techniques to confuse attackers, such as altering packet contents or delaying responses. While active IDS can be effective in preventing or stopping attacks, they also come with some potential downsides. False positives can occur, leading to legitimate traffic being blocked or disrupted. In addition, attackers may be able to detect and circumvent active IDS measures, leading to a false sense of security. Overall, active IDS can be a useful tool in protecting networks and systems from attacks, but they should be used in conjunction with other security measures and regularly evaluated for effectiveness.

5.2 Passive IDS

It is a type of IDS that monitors network traffic and events without actively interfering with it. Unlike active IDS, passive IDS do not generate any response or take any action to prevent or stop an intrusion. Instead, they analyze and report the events to a central management console or a security analyst. Passive IDS use various techniques such as signature-based detection, anomaly-based detection, and statistical analysis to identify potential threats and attacks. Signature-based detection compares network traffic to known patterns of malicious activity, while anomaly-based detection looks for unusual or abnormal behavior in the network traffic. Statistical analysis identifies patterns of activity that could indicate an attack or a security breach. It is commonly used in large organizations where network performance is critical and active IDS could cause disruptions. They are also suitable for detecting sophisticated attacks that are designed to evade active IDS. However, passive IDS have some limitations, such as the inability to prevent an attack or respond to an intrusion in real-time.

According to A. Shaghahi et al., the separation of network logic in SDN and its architecture design pose potential threats to the SDN system. The author proposes securing the data plane by controlling it through various authentications and by preventing unauthorized data flow. The author also presents a scenario to limit potential attacks by analyzing available threat vectors.

N. Shone, T. Ngoc, V. Phai, and Q. Shi proposed a system called the Non-Symmetric Deep Auto Encoder, which was tested on GPU-enabled Tensorflow using KDDCup'99 and NSL-KDD datasets.

In their research, T. Ubale and A. K. Jain examined Denial-of-Service attacks in SDN and conducted a survey to gain insight into the vulnerability of SDN. The authors also proposed a topic for future research in the networking field.

N. Marir et al. proposed using DBN and ensemble SVM to detect abnormal behavior in a distributed environment. They investigated how reducing the vast amount of network data can improve the prediction results of distributed ensemble SVM. The authors used techniques based on Apache Spark for their research.

Abbas Yazdinejadna et al. proposed a Kangaroo-based Intrusion Detection System (KIDS) which detects anomalous behavior in the data plane of SDN. The system uses a resonant pattern similar to a kangaroo's sequential hop to identify an attack on the SDN controller. The KIDS system was designed to be highly scalable and fault-tolerant.

Thomas Girdler et al. developed an SDN-based IDPS which provides protection against ARP Spoofing attacks and Blacklisted MAC Addresses. OpenvSwitch and the Python-based SDN Controller POX (OvS) were used by the authors. Their research showed that the system has a zero false positive rate and is able to recognize every packet successfully.

Kaur et al. [22] reviewed resource management in the Fog/Edge computing paradigm in great detail, with an emphasis on how it integrates with IoT applications. They looked at the difficulties in handling resources in heterogeneous, distributed contexts as well as the varying demands on computational nodes. In order to investigate AI and non-AI based solutions for computing resource provision, job offloading, scheduling, service placement, and load balancing, they examined over 490 publications, shortlisting 223 for analysis. The literature they covered spans the years 2018 to 2023. They also talked about how combining Fog/Edge computing with cutting-edge technologies like blockchain and 5G could improve the intelligence and efficiency of IoT applications.

In order to manage resources in fog and edge computing environments, Sundas et al. [23] conducted a systematic literature review on the use of AI and ML approaches. They investigated the difficulties in managing resources because of these environments' unpredictability, heterogeneity, fluctuating workloads, and resource restrictions. Through an analysis of different AI/ML techniques—particularly those able to make sequential decisions, such as reinforcement learning—the study explored the advantages and disadvantages of these technologies, addressing problems like high variance, explainability, and the need for online training. The review also evaluated current approaches, suggested a taxonomy of AI/ML-based resource management strategies, and suggested future research areas in AI/ML-based fog/edge computing.

Suman et al. [24] investigated how cloud computing systems could be subject to Distributed Denial of Service (DDoS) assaults, which are a serious risk to the CIA (confidentiality, integrity, and availability) of data sent over the internet. Because DDoS attacks can resemble legal traffic, it can be difficult to detect them. For this reason, the study focuses on implementing intrusion detection systems (IDS) that are augmented with cutting-edge machine learning (ML) approaches. Analyzing and enhancing DDoS threat identification and mitigation in cloud environments was the aim. The purpose of this effort is to further the subject of cloud security and provide fresh researchers with an avenue of inquiry.

Through optimized task offloading in a cloud-fog environment, the kumar et al. [25] built an AI-enabled framework to improve the performance of latency-sensitive Industrial Internet of Things (IIoT) applications. They incorporated fog computing to solve latency difficulties after realizing that regular cloud computing was unable to satisfy the expectations of IIoT applications. The system uses an AI-based Whale Optimization Algorithm (WOA) to optimally allocate resources with the goal of enhancing Quality-of-Service

(QoS) parameters. It also incorporates a fuzzy-based offloading controller for real-time decision-making. Compared to traditional offloading and allocation algorithms, their experimental results show a considerable improvement in makespan time, energy consumption, and execution cost.

The Fig. 14 organizes the design choices of IDS into three interconnected components. 'Intrusion Technologies' differentiates between Host-based Intrusion Detection Systems (HIDS) and Network-based Intrusion Detection Systems (NIDS), indicating the scope of monitoring each type provides. 'Intrusion Methodologies' details the detection mechanisms employed by IDS, including Signature, Anomaly, Specification, and Hybrid Based IDS, which define how the IDS identifies potential threats. Finally, 'Intrusion Detection Approaches' elaborates on the underlying principles and frameworks used for detection, such as Statistical, Rule, Heuristic, Pattern, Cloud, Machine Learning (ML), and Deep Learning (DL) Based approaches, which indicate the analytical and decision-making processes of the IDS. Each category builds upon the previous, from the deployment type to the detection mechanism, and finally, the analytical approach, together forming a comprehensive structure of IDS design choices within SDN.

5.3 IDS technologies

IDS can be broadly classified into two categories: Host-Based IDS (HIDS) and Network-Based IDS (NIDS). HIDS operates on a specific host or endpoint, analyzing the system's events and logs for signs of suspicious activity. NIDS, on the other hand, monitors network traffic to identify potential security breaches. Both HIDS and NIDS play an essential role in securing systems and networks, but each has its unique strengths and limitations. Understanding the differences between these two types of IDS is essential for selecting the appropriate solution to protect against potential security threats.

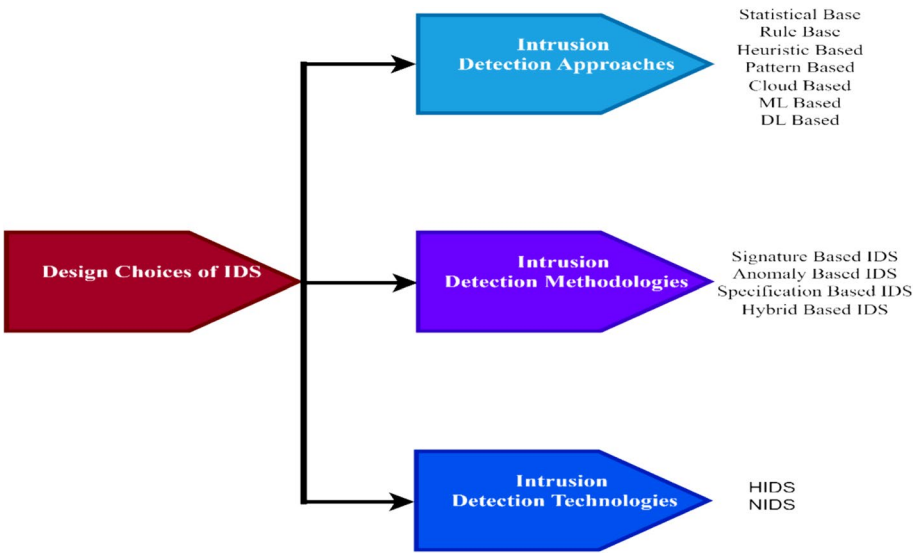

Fig. 14 Design Choices of IDS

5.4 HIDS

Dacier and Wespi introduced the concept of Intrusion Detection in 1999 to monitor system and related records based on a typical image. Ou et al. designed and implemented a host-based intrusion detection system with two detection techniques. These techniques include log file analysis and back propagation neural network technology for abuse and anomaly detection, respectively. The proposed HIDS aims to enhance the accuracy and efficiency of intrusion detection by combining both detection methods. The study results indicate that the suggested approach has increased the effectiveness and accuracy of attack detection.

5.5 NIDS

A Network-Based Intrusion Detection System (NIDS) is primarily utilized to analyze network traffic entering and leaving system nodes to identify any unauthorized access, abnormal behavior, and subsequent network attacks. The challenges that an intrusion detection system encounters regarding accuracy and false positive alarms were discussed by Z. Ahmad et al. The authors suggested using a Data Science-based system (DL/ML) in IDS to identify potential warnings with more precision and fewer false positive alerts. In developing the NIDS system, the authors proposed an approach to categorize the crucial ML/DL algorithms.

Chuanlang Yin et al. proposed a Deep Learning (DL) based Intrusion Detection System (IDS) known as RNN-IDS. This model was suggested to enhance the ability to detect intrusion types and improve the accuracy of intrusion detection.

5.6 IDS detection methods

Above, the different systematic types of IDS Detection Methods are explained, as illustrated in Fig. 14 (Wanda, 2020).

5.7 Signature-based detection techniques

It involve matching attack signatures against a database of known threats and evaluating network traffic against it. When a match is found, an alert is generated for identification. This method is quite accurate in identifying known attacks but fails to detect zero-day threats. V. Kumar and O.P. Sangwan conducted signature-based attack detection using Snort, a popular NIDS that inspects packets on a network and compares them to known attack signatures. They used the DARPA dataset to analyse the irregular connections detected by Snort. The Snort attack signature database is updated periodically. This IDS System is proven to be effective in identifying and evaluating intrusions in live network traffic.

5.8 Anomaly-based detection technique

It involves sounding an alert when any deviations above the permitted threshold are detected, without identifying the specific type of attack. To improve the effectiveness of anomaly-based detection, creating normal profiles was found to be more effective than

attempting to replicate normal and attack events using machine-learning models as behavior identification equivalents. This technique has proven to be more efficient than signature-based methods in identifying previously unseen attacks that do not fit known attack patterns in real-world networks.

S. Dwivedi, V. M., T. S., and S. A. K. proposed the use of Passban, an intelligent intrusion detection system based on anomaly detection, which can provide protection to IoT devices that are directly connected. The system is capable of utilizing edge computing to detect cyber threats close to the data source. Passban was used in the first instance as an IDS that was directly installed on the gateway responsible for processing information from IoT devices and the Internet. Based on the evaluation results, Passban was able to accurately and efficiently detect attacks.

Celyn Birkinshaw et al. designed, experimented, and assessed two connection-based methods, namely CB-TRW (Credit Based- Threshold Random Walk) and RL (Rate Limiting), as part of their IDPS. They tested the PB (Port-Bingo) technique against port scanning and extended the RL algorithm to include anomaly detection for TCP, UDP, and ICMP network traffic.

5.9 Specification-based detection techniques

It utilize a database of rules with associated deviation ranges specified by human experts to accurately record the typical response of a system and contrast it with current system observations to identify external deviations. While both anomaly identification and specification-based detection techniques share the same principles, anomaly-oriented solutions integrate normal behavior with machine learning, whereas their specification-oriented counterparts require manual specifications. As a result, the false-positive rates are reduced in specification-based detection techniques compared to their anomaly-oriented counterparts.

5.10 Hybrid detection technique

The methods combined various techniques to address shortcomings and optimize the benefits of existing and new attacks. To achieve the required processing and storage balance for both methods, hybrid intrusion detection techniques were developed by combining anomaly and signature identification techniques. Similarly, balancing the cost of storing signature recognition and the cost of estimating the anomaly equivalent price was crucial.

5.11 Intrusion detection approaches

Intrusion detection systems (IDS) are crucial components of any network security infrastructure. They aim to detect and respond to potential security breaches in a timely and effective manner. There are several different approaches to IDS that can be used, each with its own advantages and limitations. These approaches include static-based detection, rule-based detection, heuristic-based detection, machine learning based detection etc. Each of these approaches has its own unique strengths and weaknesses, and a successful IDS implementation will likely incorporate a combination of several different techniques to achieve optimal performance.

5.12 Statistical IDS

This approach is a type of intrusion detection system that uses statistical techniques to detect both known and unknown attacks. Unlike signature-based IDS, which relies on a database of known attack patterns, statistical IDS relies on statistical analysis of network traffic to identify patterns of behavior that deviate from normal activity. This approach is useful in detecting zero-day attacks, which are new and previously unknown vulnerabilities that have not yet been documented or added to a signature database. However, it requires a significant amount of time and resources to create a baseline of normal network activity and to refine the statistical models over time. Additionally, statistical IDS may produce a high number of false positives if the statistical models are not fine-tuned correctly. Overall, statistical IDS is a powerful tool for detecting a wide range of attacks, but it requires careful configuration and ongoing maintenance to ensure that it is effective and efficient.

5.13 Rule-based IDS

As the name suggests, relies on predefined rules to identify and detect malicious activities. It uses a set of predefined rules or signatures to identify known attacks. The rules are typically created based on characteristics of past attacks or by analyzing system vulnerabilities. Rule-based IDS is simple and effective in identifying a wide range of attacks and is easy to maintain. This approach requires only a few rules to identify thousands of attacks as slight modifications in attack scripts cannot evade the detection system. Moreover, it is efficient in detecting new attack variations, provided there are predefined rules for it. However, this approach may fail to detect new and unknown attacks as it solely relies on pre-defined rules. Additionally, a large number of rules are required to identify all potential cyber threats, which can result in high false positive rates if not maintained properly. Despite these limitations, rule-based IDS is still a widely used approach due to its simplicity and effectiveness in detecting known attacks.

5.14 Heuristic-based IDS

It relies on a set of rules or algorithms to identify potential intrusions, making it a useful approach for detecting both known and unknown attacks. It operates by analyzing the behavior of network traffic or system activities to identify patterns that may be indicative of an intrusion. When a new attack is detected, heuristic-based IDS may expand the list of malicious behaviors to improve its accuracy in detecting future attacks. One of the challenges of heuristic-based IDS is that some attackers may use evasion strategies to avoid detection. For example, they may modify their attack scripts or employ techniques such as fragmentation to hide their activities from the IDS. To overcome this challenge, heuristic-based IDS may use advanced techniques such as protocol analysis and anomaly detection to improve its accuracy in detecting such attacks. Overall, heuristic-based IDS is a useful approach for detecting a wide range of cyber threats, including both known and unknown attacks.

5.15 Pattern-based IDS

It is an approach that relies on detecting pre-defined patterns of attacks in network traffic or system logs. This approach is effective in quickly identifying known attacks and taking appropriate action to mitigate them. Pattern-based IDS systems can be quickly and easily implemented because they do not require complex algorithms or machine learning models to operate. However, one major limitation of pattern-based IDS is that it can only detect known attacks for which a pattern has already been defined [19]. This means that it is not effective in detecting new or unknown attacks that have not yet been discovered or have not yet had their patterns identified. Therefore, pattern-based IDS systems are often used in conjunction with other detection methods, such as heuristic-based or anomaly-based IDS systems, to provide comprehensive coverage against both known and unknown attacks.

Gunduz and Das propose an innovative method for detecting intrusions, using a set of rules as a form of pattern recognition. They implemented a pattern-based intrusion detection model to complement the existing statistically-based intrusion detection model. The model was tested on a dataset created during the study, and achieved a 75% accuracy rate. However, relying solely on signature-based attack detection may not be sufficient. Therefore, further work could be done by integrating anomaly-based intrusion detection into the system.

5.16 Cloud-based IDS

It is an emerging technology that offers numerous benefits over traditional IDS. It provides a limitless computing capability and scalability, enabling companies to handle large amounts of data at a lower cost. The 24/7 data availability and enhanced computing capacity allow for real-time detection and response to network intrusions [26]. Various IDS strategies, including signature-based, anomaly-based, and hybrid IDS, can be implemented on the cloud infrastructure. A cloud-based IDS typically consists of a signature database, service console, and analysis engine, providing a comprehensive security solution for network intrusion detection. The use of cloud-based IDS improves performance, reduces processing time, and enables faster detection of network intrusions. The technology is still in its early stages, but it is expected to become increasingly popular in intrusion detection systems in the near future. As cloud computing technology continues to evolve and improve, cloud-based IDS will become an even more essential tool for companies looking to protect their networks from cyber attacks [27].

Ahmed M. El-Shamy and colleagues proposed a monitoring algorithm that utilises SDN (Software Defined Networking) and SVM (Support Vector Machine) to detect performance anomalies and locate bottlenecks in distributed applications within cloud data centres. The SVM Algorithm performs two steps to achieve this: firstly, it employs OCSVM (One-Class SVM) to classify the response time performance of the front-end server as either normal or abnormal. Secondly, it uses MCSVM (Multi-Class SVM) to recognise the type of anomaly and identify the bottleneck's source.

5.17 Machine learning based IDS

It is an approach that automates the analysis procedure to recognize intrusions in computer networks. It utilizes various supervised, unsupervised, and semi-supervised learning techniques and a range of algorithms to identify patterns and anomalies in network traffic [28]. The ML-based IDS has the potential to learn from historical data to predict new attacks, and this can be particularly useful in detecting zero-day attacks [29].

During the supervised learning process, a model is trained by making use of labelled data in order to categorize network traffic as either normal or malicious. During the unsupervised learning process, the algorithm looks for patterns in the data that depart from the typical pattern of activity, which could point to the presence of an intrusion. During the training phase of semi-supervised learning, the model is educated with the assistance of both labelled and unlabeled data. ML-based intrusion detection systems provide various advantages over traditional intrusion detection systems. These advantages include the capacity to adapt to new threats without the need for manual updates, better accuracy in spotting complicated intrusions, and lower false positive rates. However, the efficacy of the ML-based IDS is contingent on the quality and quantity of data used for training the model, as well as the model's capacity to adapt to shifting attack patterns.

N. Satheesh et al. discussed the use of machine learning algorithms for anomaly detection in the SDN environment. They also examined the possibility of using the available network throughput for fault tolerance and fast routing, which could help mitigate security breaches in the SDN architecture.

G. Karatas, O. Demir, and O. K. Sahingoz applied six different machine learning models to implement the CSE-CIC-IDS2018 dataset. They tested the system using Keras/TensorFlow and programmed it using Python.

The MSML (Multilevel Semi-Supervised Machine Learning) framework was introduced by H. Yao et al. as an approach that incorporates multiple levels of semi-supervised machine learning. The performance of the framework was evaluated using the KDDCUP99 dataset. Additionally, the MSML framework utilizes a combination of supervised and unsupervised learning techniques to improve the accuracy of intrusion detection systems. By incorporating both labeled and unlabeled data, the MSML framework can detect known and unknown attacks, making it an effective approach for enhancing the security of computer systems.

Catania and C. Garino emphasized the importance of automation in identifying network hazards through signatures. Automation has been a guiding principle in the development of signature-based intrusion detection systems used by many industry leaders. The authors presented a perspective to detect attacks by integrating the capabilities of AI/ML into current practices. The article aims to expand the landscape of current practices by incorporating machine learning capabilities to make the system more robust.

R. Ravi et al. introduced a new approach named SDN-oriented prevention technique for detecting phishing attacks in cyberspace. This method involves using DML (Deep Machine Learning) with the CANTINA strategy, known as DMLCA. The DMLCA technique was tested with different settings, and it achieved high identification accuracy. By applying an ML strategy that addresses the challenges of phishing attacks, such as SVM, this approach effectively resolved the problem of classification complexity.

In their research, S.K. Dey et al. investigated the potential security threats and vulnerabilities in network systems when applied directly to SDN. The authors suggested alternative methods to address these challenges. One such method is the use of a random forest approach to employ machine learning for feature selection. In addition, they presented the use of a DNN for feature selection using recursive feature elimination and the ANOVA test.

5.18 Deep learning-based IDS

It is an innovative approach that uses multiple layers to learn and recognize different types of attacks. With the help of deep learning techniques, it can identify data anomalies, recognize massive datasets, and efficiently handle time-varying datasets. Marcos V.O. de Assis, Luiz F. Carvalho, Joel J.P.C. Rodrigues, Jaime Lloret, and Mario L. Proenasa Jr conducted an experiment to detect DDoS attacks using Convolution Neural Network (CNN), which is a popular deep learning method. By using CNN, they were able to accurately identify DDoS attacks with a high level of precision. This approach can be very effective in detecting various types of attacks in a network, making it an essential tool for network security professionals.

M. Albahar explained the significance of the SDN framework and how it reduces networking costs. However, there is a glitch in the framework that compromises the network's security. To address this issue, the author conducted an experiment using the RNN-SDR model to demonstrate its potential in ensuring network security without compromising performance.

In their work, M. Wang et al. proposed a set of guidelines aimed at encouraging researchers to integrate machine learning and deep learning techniques into their networking-related work. They emphasized the importance of adopting a standard framework for developing machine learning network (MLN) applications, as well as the need to incorporate the latest ML/DL algorithms and technologies. The authors argued that by leveraging data science, researchers could unlock new opportunities to advance the field of networking and address some of its most pressing challenges. For instance, ML/DL can be used to optimize network performance, enhance security, and reduce costs. By following the proposed guidelines, researchers can develop more effective MLN applications and contribute to the ongoing evolution of networking technology.

T.A. Tang et al. explained that the SDN network flow is managed by a central command, which makes it more secure and resilient. To detect intrusion in the SDN, the authors recommended a deep learning method based on GRU-RNN. They tested this technique on data collected from the SDN controller, which included measuring the network's latency and throughput. By employing this approach, they aimed to improve the SDN's security measures against possible attacks. Tables 8 and 9 is shown summarize comparison of papers with future direction and evaluation comparison with attacks respectively.

6 SDN limitation and Issues

Despite its potential to enable, create, sustain, and provide automation to network administration, SDN's operation and performance in networking organizations may be restricted by technical obstacles [46]. Currently, only small test beds for research prototypes are in use for SDN deployments. Figure 15 provides context for our focused analysis on security within SDN by illustrating how security is intrinsically linked to other

Table 8 Comparison analysis of previous papers

Sr. No	Authors	Technique	Methodology	Outcomes	Future Direction	Publication
1	C. Catania and C. Garino (2012) [30]	ML	Signature-based approach	The author spoke about the value of automated processes in utilising network signatures to identify harmful occurrences	Fully atomization Intrusion Detection Process issues like labelling of network traffic and its associated resources are analysed deeply	ScienceDirect
2	Salah Eddine Benaicha et al. (2014) [31]	NN(DL)	GA	GA is applied to improve search time in the audit data	DARPA dataset need to be enriched	2014 Science and Information Conference
3	Marwan Ali Albahar (2019) [32]	DL	RNN-SDR	RNN-SDR had considerable real-time detection potential	In order to make the network significantly more secure or quicker, it needs to be optimised to meet its requirements	Hindawi
4	Marcos V.O. de Assis, Luiz F. Carvalho, Joel J.P.C. Rodrigues, Jaime Lloret, Mario L. ProenÃ§a Jr (2020) [33]	CNN (DL)	CICDDoS 2019 Datasets, Anomaly Detection approach	Low positive rates, more accuracy, and greater outcomes precision were all attained by CNN. The mitigation method successfully resumed regular SDN operations	Investigate how RDL affects SDN classification issues	ScienceDirect
5	Wenjuan Li et al (2021) [34]	CIDN	Trust management scheme to detect insider attackers	To log an occurrence of CIDNs in respect to the SDN architecture, a more tolerant and robust method was designed	Determining a reasonable threshold and how to use Blockchain-based CIDN in SDN are still unanswered questions	ScienceDirect
6	Abbas Yazdinejadna et al. (2021) [35]	KIDS in SDN	DARPA and NSL-KDD Datasets	The ultimate product is highly scalable and resilient to failure	Utilize the KIDS architecture when detecting attacks by using NFV and P4 structure	ScienceDirect
7	Garcia et al. (2021) [36]	ML	K-mean and GMM	The results were 0.98% accurate, practical, and high-performing to successfully and promptly identify SlowDoS cyber attacks	Adapt or expand our network flow processor, an AI-based model for spotting cyber attacks, must include 5G traffic	Science Direct

Table 9 Comparisons analysis of attack focused and evaluation metrics

Sr. No	Authors and Reference	Year	AI	Datasets	Methodology	Attack focused	Evaluation Metrics	Simulation Tools	Publication
1	Samrat Kumar Dey, Md. Mahbubur Rahman [37]	2019	ML DL	NSL-KDD Tf. Train Adam Optimizer	Flow Based Anomaly Detection	DoS Probing U2R R2L	Precision Recall F-Score Accuracy	Open Flow Controller	MDPI Symmetry
2	Iqbal H. Sarker et.al. [38]	2020	ML		Predictive Data Analysis with IntruD Tree Cyber Security Model		Precision Recall F-Score Accuracy True Positive Rate(TPR), False Positive Rate(FPR)	Python Programming with Scikit Learn	MDPI Symmetry
3	Yin et al [39]	2017	DL	NSL-KDD	Recurrent Neural Network	DoS Probing U2R R2L Normal Anomaly	Accuracy False Alarm Rate(FAR)		IEEE ACCESS
4	Naila Marir et al. [40]	2018	ML,DL	NSL-KDD UNSW-NB15 CICIDS2017	Deep Belief Network, Support Vector Machine	DoS Probing, U2R, R2L,Normal	Precision Recall F-Measure,Area under ROC Curve	Hadoop version 2:7:3., Spark version 2.3.0	IEEE ACCESS
5	Shone et al. [41]	2018	ML DL	NSL-KDD KDD Cup'99	Non Symmetric Deep Auto Encoder with Random Forest			Tensor-Flow	IEEE
6	Yao et al. [42]	2018	ML	KDD Cup'99	Multilevel Model based on K-Means Clustering and Random Forest		Accuracy Precision Recall F-Measure		IEEE Internet of Things Journal

Table 9 (continued)

Sr. No	Authors and Reference	Year	AI	Datasets	Methodology	Attack focused	Evaluation Metrics	Simulation Tools	Publication
7	Karatas et al. [43]	2020	ML	CSE-CIC-IDS2018	SMOTE	Benign BoT DoS Brute-Force Infiltration SQL-Injection	Accuracy Precision Recall F-SCORE	Keras/Tensor flow using Python programming language	IEEE ACCESS
8	Celyn B. et al. [44]	2019			Anomaly Based IDPS	DoS Port-Scanning (TCP, UDP, ICMP based attacks)	Threshold Parameters	POX Controller and Open Flow 1.0	Elsevier/JNCA
9	Ahmed M. El-Shamy et al. [45]	2021	DL		Anomaly Detection SVM Algorithm	Normal vs. Abnormal	Precision Recall F-Score	MiniNet Emulator	Science Direct/ Egyptian Informatics Journal
10	Thomas Girdler et al. [44]	2021				ARP Spoofing Attacks and Blacklisted MAC addresse		POX controller OpenvSwitch(OvS) Virtual Box:HypervisorWire-Shark	Elsevier/ Science Direct/ Computers and Electrical Engineering

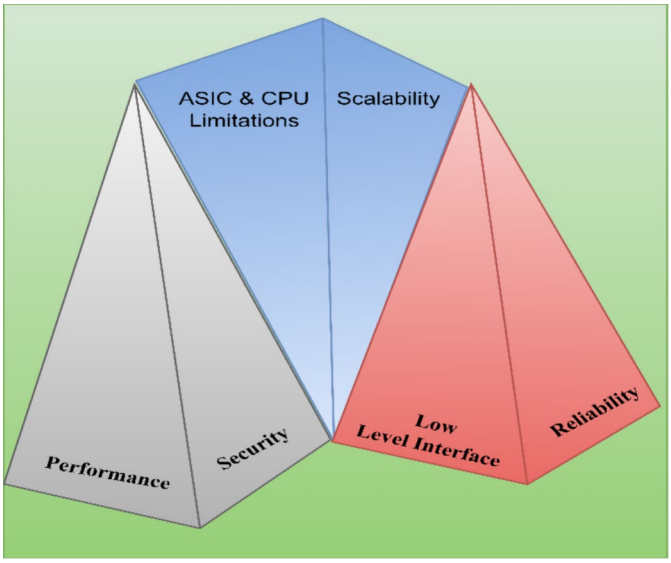

Fig. 15 SDN Challenges

SDN operational challenges such as reliability and controller placement. These factors collectively influence the network's security landscape, justifying the figure's inclusion to demonstrate that a comprehensive security strategy must consider the broader array of SDN challenges. Table 10 presents the SDN issues covered and explored in these research articles.

6.1 Security

There are various new security challenges that have emerged, including attacks on traffic flows, switches, management systems, controller recovery, failure diagnosis, and communication related to the Control plane. These challenges may allow attackers to take control of the SDN controller and take advantage of shared network services or even take over the entire network. The use of unsafe codes in new services and applications can pose security risks to network administrators and programmers that were not previously present [51]. The virtual nature of SDN may also create multiple network segments, each with its own set of risks and challenges.

Table 10 Highlight SDN Issues

S. No	Authors	SDN issues
1	M. et al. Ashton (2013)	SDN Reliability
2	Saad H. Hanji et al. (2021) [47]	SDN Scalability
3	Priyadarsini M. et al. (2017) [48]	Performance
4	Ejaz et al. (2019) [49]	Controller Placement
5	Yassive Maleh et al. (2022) [50]	SDN Security

6.2 Scalability

It is a major concern in SDN networks as the central or distributed controllers communicate with various devices' data planes. As the network size expands, an excess of network requests made by the control plane or controller can congest the controller, leading to a decline in network performance [20]. In large or dynamic networks, controllers must make quick decisions in the face of numerous and diverse events, such as failures, changes in traffic, or new incoming flows, which makes scalability a significant challenge.

6.3 Performance

The effectiveness of SDN technology is crucial for a network's reliability, security, scalability, and interoperability. Two metrics are used to evaluate the efficiency of flow-based technology: the time it takes to set up a flow and the number of flows the controller can switch without disrupting the network. However, as the decision-making overhead at the controller increases, the system may gradually slow down. This is especially true if the controller is centralized or distributed. To overcome this performance limitation, attention should be focused on factors that affect the flow-setup time and I/O performance of the controller.

7 Future trends and observation

Looking ahead, progress in AI and machine learning will likely propel the growth of SDN. These improvements will greatly improve the ability to identify intrusions and respond quickly. When these technologies are combined, they can make security measures smarter and more predictive, so problems can be stopped before they happen. Additionally, the growth of quantum computing offers both a chance to make network security protocols better and a task when it comes to cryptographic protection. With so many IoT gadgets on the market, we also need scalable security solutions, which SDN is perfectly suited to provide. As SDN becomes more connected to these new technologies, it becomes more and more important to create flexible, smart security systems. A large number of companies are interested in incorporating AI-based security products, as shown in Fig. 16.

Findings from our study show that even though network technologies have improved, legacy datasets like NSL-KDD and KDD Cup'99 are still widely used, making up 60% of all uses in testing and validating network security algorithms (Fig. 17). Their continued use is due to the large number of comparative data they provide in the literature. There is an urgent need for up-to-date datasets that represent the current landscape of cybersecurity threats, though, because network architectures have changed a lot in the last 20 years and new types of threats are always appearing. The time of "big data" and the focus on sensor node privacy make it even more important to have datasets that can help with these current problems.

Figure 18 also shows how the researchers rated the methodology, especially the performance indicators they used in the study. The most important measures are detection rate, accuracy, and recall, which shows how important it is to be very precise when

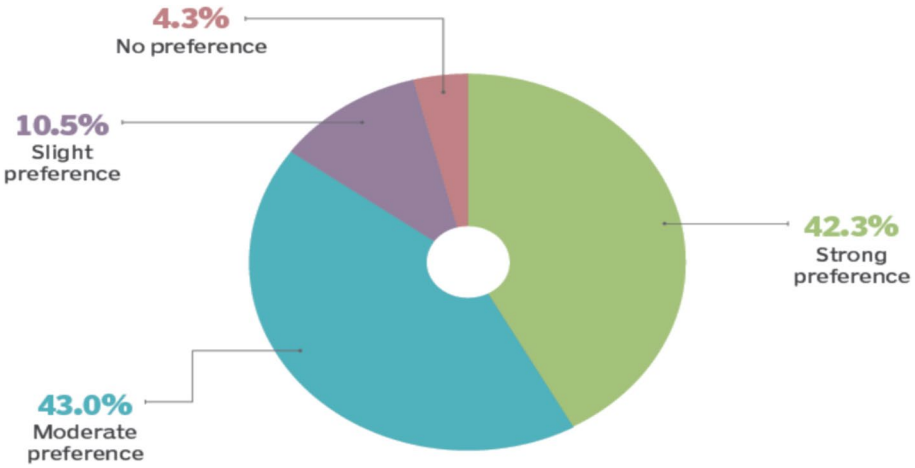

Fig. 16 Preference to AI techniques

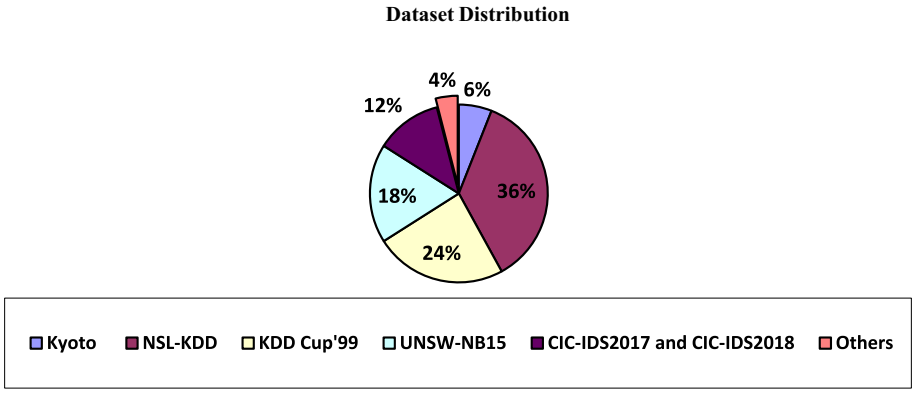

Fig. 17 Dataset Distribution

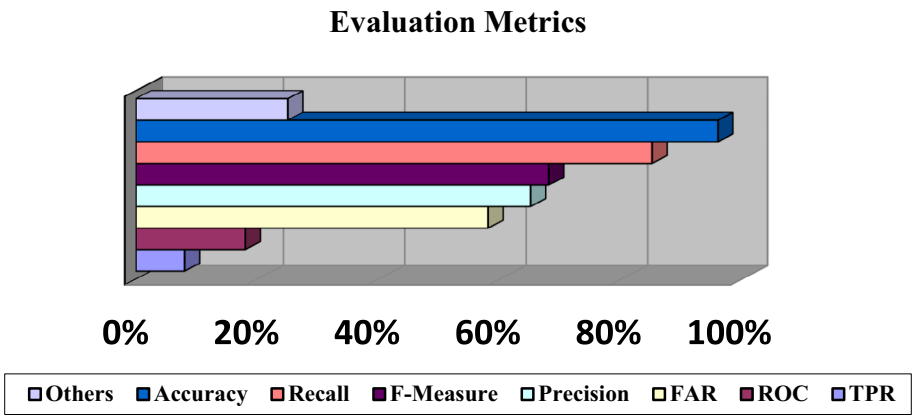

Fig. 18 Evaluation Metrics

identifying threats. It is important to think about more than just accuracy and recognition rates when designing network security systems, especially those that use machine learning and deep learning. These include recall and F-measure as well. These measures are very important for figuring out how well a threat detection system works because they make sure it can find real threats while also reducing the number of false positives. This is very important in the complicated field of network security.

8 Conclusion and future direction

In our comprehensive examination of SDN, we have traversed through the broad spectrum of its significant impact on both academic research and industry applications. Our research includes various SDN applications including data centers, cloud computing, wireless LAN, Smart Grid, SDN Forensic, and SDN-on-Chip, highlighting the widespread adoption and versatility of SDN technologies. Central to our investigation was the identification of key datasets critical for intrusion, threat, and attack detection within SDN environments. We meticulously detailed the focus of each dataset on specific attacks, the temporal context of their development, and their origins, addressing our inquiry into the most utilized datasets specific to SDN security.

Further, our paper delved into the evaluation metrics crucial for the design of IDS in SDN, spotlighting the evolving threat vectors that pose imminent challenges to network security. This discussion not only underscored the necessity for ongoing research into robust attack detection methods but also highlighted the unique advantage of SDN in providing fine-grained security measures unattainable by traditional network architectures. Despite the benefits, the centralized control mechanism of SDN underscores a critical vulnerability, necessitating relentless innovation in attack detection methodologies to safeguard against potential data breaches.

Looking forward, our finding advocate for the development of contemporary datasets that reflect the rapidly changing network infrastructures and the sophistication of emerging security threats. This ensures that IDS capabilities are both relevant and effective in the face of new challenges. Moreover, there exists an essential demand for advancing machine learning and deep learning-based threat detection models. Such models promise to significantly enhance the adaptability and predictive accuracy of SDN security frameworks, ultimately fortifying the resilience of SDN against the dynamic spectrum of cyber threats. Our research questions guided this inquiry, laying a foundation for future investigations that will continue to push the boundaries of SDN technology and security.

Dataset availability Data are available on request from the submitting author.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Dahiya S, Siwach V, Sehrawat H (2021) Review of AI Techniques in development of Network Intrusion Detection System in SDN Framework," in *2021 International Conference on Computational Performance Evaluation (ComPE)*, 168–174. https://doi.org/10.1109/ComPE53109.2021.9752430
2. Sagu A, Gill NS, Gulia P, Chatterjee JM, Priyadarshini I (2022) A Hybrid Deep Learning Model with Self-Improved Optimization Algorithm for Detection of Security Attacks in IoT Environment. Future Internet 14(10):10. https://doi.org/10.3390/fi14100301
3. Sagu A, Gill NS, Gulia P, Singh PK, Hong W-C (2023) Design of metaheuristic optimization algorithms for deep learning model for secure IoT environment. Sustain 15(3):3. https://doi.org/10.3390/su15032204
4. Banse C, Schuette J (2017) A taxonomy-based approach for security in software-defined networking," in *2017 IEEE International Conference on Communications (ICC)*, 1–6. https://doi.org/10.1109/ICC.2017.7997245.
5. Xia W, Wen Y, Foh CH, Niyato D, Xie H (2015) A Survey on Software-Defined Networking. IEEE Commun Surv Tutor 17(1):27–51. https://doi.org/10.1109/COMST.2014.2330903
6. "What is Software-Defined Networking?" Accessed: Jul. 16, 2023. [Online]. Available: https://www.ibm.com/topics/sdn
7. Kreutz D, Ramos FMV, Veríssimo PE, Rothenberg CE, Azodolmolky S, Uhlig S (2015) Software-Defined Networking: A Comprehensive Survey. Proc IEEE 103(1):14–76. https://doi.org/10.1109/JPROC.2014.2371999
8. Wickboldt JA, De Jesus WP, Isolani PH, Both CB, Rochol J, Granville LZ (2015) Software-defined networking: management requirements and challenges. IEEE Commun Mag 53(1):278–285. https://doi.org/10.1109/MCOM.2015.7010546
9. Gomez-Rodriguez JR, Sandoval-Arechiga R, Ibarra-Delgado S, Rodriguez-Abdala VI, Vazquez-Avila JL, Parra-Michel R (2021) A survey of software-defined networks-on-chip: motivations, challenges and opportunities. Micromachines 12(2):183. https://doi.org/10.3390/mi12020183
10. Securing Software Defined Networking Using Intrusion Detection System - A Review | SpringerLink. Accessed: Jul. 16, 2023. [Online]. Available: https://link.springer.com/chapter/https://doi.org/10.1007/978-981-16-8059-5_26
11. A survey on the architecture, application, and security of software defined networking: Challenges and open issues - ScienceDirect. Accessed: Jul. 16, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2542660520301219
12. (2007) First International Symposium on Empirical Software Engineering and Measurement-Title in *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, i–iii. https://doi.org/10.1109/ESEM.2007.84.
13. Systematic mapping studies in software engineering | Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering. Accessed: Jul. 16, 2023. [Online]. Available: https://dl.acm.org/doi/https://doi.org/10.5555/2227115.2227123
14. UCI KDD Archive. Accessed: Jul. 16, 2023. [Online]. Available: http://kdd.ics.uci.edu/
15. Song J, Takakura H, Okabe Y, Eto M, Inoue D, Nakao K (2011) Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation," in *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, in BADGERS '11. New York, NY, USA: Association for Computing Machinery, 29–36. https://doi.org/10.1145/1978672.1978676.
16. Tavallaee M, Bagheri E, Lu W, Ghorbani AA (2009) A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. 1–6. https://doi.org/10.1109/CISDA.2009.5356528.
17. Moustafa N, Slay J (2015) UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 1–6. https://doi.org/10.1109/MilCIS.2015.7348942.
18. Sharafaldin I, Habibi Lashkari A, Ghorbani AA (2018) Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization:," in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, Funchal, Madeira, Portugal: SCITEPRESS - Science and Technology Publications, 108–116. https://doi.org/10.5220/0006639801080116.
19. Network intrusion detection system: A systematic study of machine learning and deep learning approaches - Ahmad - 2021 - Transactions on Emerging Telecommunications Technologies - Wiley Online Library. Accessed: Jul. 16, 2023. [Online]. Available: https://onlinelibrary.wiley.com/doi/full/https://doi.org/10.1002/ett.4150

20. Software-Defined Network (SDN) Data Plane Security: Issues, Solutions, and Future Directions | SpringerLink. Accessed: Jul. 16, 2023. [Online]. Available: https://link.springer.com/chapter/https://doi.org/10.1007/978-3-030-22277-2_14
21. Rahouti M, Xiong K, Xin Y, Jagatheesaperumal SK, Ayyash M, Shaheed M (2022) SDN security review: threat taxonomy, implications, and open challenges. IEEE Access 10:45820–45854. https://doi.org/10.1109/ACCESS.2022.3168972
22. Walia GK, Kumar M, Gill SS (2024) AI-Empowered Fog/Edge Resource Management for IoT Applications: A comprehensive review, research challenges, and future perspectives. IEEE Commun Surv Tutor 26(1):619–669. https://doi.org/10.1109/COMST.2023.3338015
23. Iftikhar S et al (2023) AI-based fog and edge computing: a systematic review, taxonomy and future directions. Internet Things 21:100674. https://doi.org/10.1016/j.iot.2022.100674
24. Suman OP, Kumar M (2023) Machine Learning Based Theoretical and Experimental Analysis of DDoS Attacks in Cloud Computing, in *2023 International Conference on Device Intelligence, Computing and Communication Technologies, (DICCT)*, 526–531. https://doi.org/10.1109/DICCT56244.2023.10110201
25. Kumar M, Walia GK, Shingare H, Singh S, Gill SS (2023) AI-Based Sustainable and Intelligent Offloading Framework for IIoT in Collaborative Cloud-Fog Environments. IEEE Trans Consum Electron 1–1. https://doi.org/10.1109/TCE.2023.3320673
26. H. M. I. C. P. Ltd, "Cloud Application Security & Vulnerability Management Market." Accessed: Aug. 09, 2023. [Online]. Available: https://www.openpr.com/news/2151137/cloud-application-security-vulnerability-management-market
27. Jing H, Wang J (2022) Detection of DDoS attack within industrial IoT devices based on clustering and graph structure features. Secur Commun Netw 2022:e1401683. https://doi.org/10.1155/2022/1401683
28. Arowolo MO, Ogundokun RO, Misra S, Agboola BD, Gupta B (2023) Machine learning-based IoT system for COVID-19 epidemics. Computing 105(4):831–847. https://doi.org/10.1007/s00607-022-01057-6
29. A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks | IEEE Journals & Magazine | IEEE Xplore. Accessed: Mar. 13, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9627657
30. Catania CA, Garino CG (2012) Automatic network intrusion detection: Current techniques and open issues. Comput Electr Eng 38(5):1062–1072. https://doi.org/10.1016/j.compeleceng.2012.05.013
31. Maleh Y, Fatani IFE, Gholami KE (2022) A Systematic Review on Software Defined Networks Security: Threats and Mitigations," in *Advances in Information, Communication and Cybersecurity*, vol. 357, Y. Maleh, M. Alazab, N. Gherabi, L. Tawalbeh, and A. A. Abd El-Latif, Eds., in Lecture Notes in Networks and Systems, vol. 357. , Cham: Springer International Publishing, 591–606. https://doi.org/10.1007/978-3-030-91738-8_54
32. Albahar MA (2019) Recurrent neural network model based on a new regularization technique for real-time intrusion detection in SDN environments. Secur Commun Netw 2019:1–9. https://doi.org/10.1155/2019/8939041
33. De Assis MVO, Carvalho LF, Rodrigues JJPC, Lloret J, Proença ML Jr (2020) Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. Comput Electr Eng 86:106738. https://doi.org/10.1016/j.compeleceng.2020.106738
34. Li W, Wang Y, Jin Z, Yu K, Li J, Xiang Y (2021) Challenge-based collaborative intrusion detection in software-defined networking: an evaluation. Digit Commun Netw 7(2):257–263. https://doi.org/10.1016/j.dcan.2020.09.003
35. Abbas G, Mehmood A, Carsten M, Epiphaniou G, Lloret J (2022) Safety, security and privacy in machine learning based internet of things. J Sens Actuator Netw 11(3):3. https://doi.org/10.3390/jsan11030038
36. Garcia N, Alcaniz T, González-Vidal A, Bernabe JB, Rivera D, Skarmeta A (2021) Distributed real-time SlowDoS attacks detection over encrypted traffic using artificial intelligence. J Netw Comput Appl 173:102871. https://doi.org/10.1016/j.jnca.2020.102871
37. Dey SK, Rahman MDM (2019) Effects of machine learning approach in flow-based anomaly detection on software-defined networking. Symmetry 12(1):7. https://doi.org/10.3390/sym12010007
38. Sarker IH, Abushark YB, Alsolami F, Khan AI (2020) IntruDTree: a machine learning based cyber security intrusion detection model. Symmetry 12(5):754. https://doi.org/10.3390/sym12050754
39. Yin C, Zhu Y, Fei J, He X (2017) A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 5:21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

40. Marir N, Wang H, Feng G, Li B, Jia M (2018) Distributed abnormal behavior detection approach based on deep belief network and ensemble svm using spark. IEEE Access 6:59657–59671. https://doi.org/10.1109/ACCESS.2018.2875045
41. Shone N, Ngoc TN, Phai VD, Shi Q (2018) A deep learning approach to network intrusion detection. IEEE Trans Emerg Top Comput Intell 2(1):41–50. https://doi.org/10.1109/TETCI.2017.2772792
42. Yao H, Fu D, Zhang P, Li M, Liu Y (2019) MSML: a novel multilevel semi-supervised machine learning framework for intrusion detection system. IEEE Internet Things J 6(2):1949–1959. https://doi.org/10.1109/JIOT.2018.2873125
43. Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset | IEEE Journals & Magazine | IEEE Xplore. Accessed: Aug. 09, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/8993711
44. Birkinshaw C, Rouka E, Vassilakis VG (2019) Implementing an intrusion detection and prevention system using software-defined networking: defending against port-scanning and denial-of-service attacks. J Netw Comput Appl 136:71–85. https://doi.org/10.1016/j.jnca.2019.03.005
45. El-Shamy AM, El-Fishawy NA, Attiya G, Mohamed MAA (2021) Anomaly detection and bottleneck identification of the distributed application in cloud data center using software-defined networking. Egypt Inform J 22(4):417–432. https://doi.org/10.1016/j.eij.2021.01.001
46. Alsmadi IM, AlAzzam I, Akour M (2017) A Systematic Literature Review on Software-Defined Networking," in *Information Fusion for Cyber-Security Analytics*, I. M. Alsmadi, G. Karabatis, and A. Aleroud, Eds., in Studies in Computational Intelligence. , Cham: Springer International Publishing, 333–369. https://doi.org/10.1007/978-3-319-44257-0_14
47. Haji SH et al. (2021) Comparison of Software Defined Networking with Traditional Networking. Asian J Res Comput Sci. 1–18. https://doi.org/10.9734/ajrcos/2021/v9i230216
48. Priyadarsini M, Bera P, Bampal R (2017) Performance analysis of software defined network controller architecture—A simulation based survey," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, 1929–1935. https://doi.org/10.1109/WiSPNET.2017.8300097
49. Ejaz S, Iqbal Z, Azmat Shah P, Bukhari BH, Ali A, Aadil F (2019) Traffic Load Balancing Using Software Defined Networking (SDN) Controller as Virtualized Network Function. IEEE Access 7:46646–46658. https://doi.org/10.1109/ACCESS.2019.2909356
50. A Systematic Review on Software Defined Networks Security: Threats and Mitigations | SpringerLink. Accessed: Jul. 16, 2023. [Online]. Available: https://link.springer.com/chapter/https://doi.org/10.1007/978-3-030-91738-8_54
51. SDN Security Review: Threat Taxonomy, Implications, and Open Challenges | IEEE Journals & Magazine | IEEE Xplore. Accessed: Aug. 09, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9760465