International Journal of Information Security (2024) 23:141–185
https://doi.org/10.1007/s10207-023-00764-1

SPECIAL ISSUE PAPER

SDN as a defence mechanism: a comprehensive survey

Believe Ayodele[1] · Victor Buttigieg[1]

Accepted: 20 September 2023 / Published online: 6 October 2023

Abstract

Investing in cybersecurity is increasingly considered a significant area and aspect a business or organisation should seriously consider. Some of these security solutions are network-based and provide many levels of protection. However, traditional networks are seen to be vendor-specific and are limited, enabling minor to no network flexibility or customisation. Implementing SDN to combat cyberattacks is a workable option for resolving this traditional network constraint. Less attention has been paid to how SDN has been utilised to address security concerns, with most surveys concentrating on the security challenges the SDN paradigm faces. This study aims to provide a comprehensive overview of the state-of-the-art on how SDN has been used to combat attacks between 2017 and 2022 by highlighting the specifics of each literature, its advantages, limitations, and potential areas for further study. This work introduces a taxonomy highlighting SDN's fundamental traits and contributions as a defence mechanism (SaaDM).

Keywords Software-defined network · SDN · Cyberattacks · Cybersecurity · Defence mechanism

1 Introduction

A traditional network security appliance, such as a firewall, intrusion prevention system (IPS), intrusion detection system (IDS), and deep packet inspection (DPI), is designed to protect the network from various cyber threats. However, many of these solutions are available as specialised hardware appliances that run proprietary network operating systems (NOS) and vendor-specific protocols requiring high-level policies translated into device-specific low-level instructions by network operators. Because there is no common networking framework, network administration and control are difficult because a network specialist is required for each vendor's equipment. All of these and more make traditional network administration and programming difficult. According to the Internet Research Task Force (IRTF) RFC 7426, software defined network (SDN) offers a framework to decouple the control plane that handles the network's intelligence from the data plane, which forwards traffic based on logic received from the control plane. The open networking foundation (ONF) summarises the principle of SDN into three: (1) decoupling traffic logic and control from traffic forwarding, (2) centralised logical control, and (3) programmability of network services [1]. These SDN principles have resulted in the overall flexibility and interoperability of networking devices, resulting in an open system that offers improvements over the limitations currently observed with traditional networks [2]. Table 1 compares a traditional network to an SDN-driven network. The open system in SDN enables the development of software that may control the connection provided by a group of network resources, the flow of network traffic via them, and any potential traffic inspection and modification that may occur in the network.

SDN has emerged as a promising solution to counter various cyber threats, including scanning, spoofing, sniffing, web application attacks, and malware attacks [3–5]. It has found applications in various sectors such as smart grids [6, 7], blockchain (BC) [8, 9], IoT [10–13], health [14], and malware remediation [15–19]. SDN is becoming a fascinating area for cybersecurity because traditional networks are complex and challenging to manage due to the vertical integration of the data and control planes. SDN provides the separation of the data and control planes, which allows for centralised networking capability and the correlation of several events based on open standards such as

✉ Believe Ayodele
believesegun@outlook.com

Victor Buttigieg
victor.buttigieg@um.edu.mt

1 Communications and Computer Engineering, Faculty of Information and Communication Technology, University of Malta, Msida, Malta

Table 1 Comparison between an SDN-enabled network and a traditional network

Criteria	SDN-enabled network	Traditional network
Feasibility	Global view of the network	Local state of the network or directly connected network
Interface	Unified API (open interface)	Vendor-specified NOS and configuration (closed interface)
Features	Decoupled data plane/forward plane	The data plane and forwarding plane are vertically integrated
Programmability	The ability to configure a network efficiently, securely, and promptly	To program the network, only vendor-specific instructions can be used
Scalability	SDN-driven networks can be easily expanded and improved	It may require the purchase of new hardware
Automation	High level of system automation	Low level of automation compared to SDN network

OpenFlow (OF) [20], Protocol-Oblivious Forwarding (POF) [21], Negotiable Datapath Models (NDM) [22], Programming Protocol-Independent Packet Processor (P4) [23], and Path Computation Element Protocol (PCEP) [24], which has led to the development of better defence mechanisms [25–29]. The standardised landscape of SDN has already widened and is still evolving. Several standard organisations and consortiums have expressed interest in standardising SDN, resulting in open-source implementations becoming a strategy for adopting SDN [2, 30–32].

This work's primary objective is to provide readers with a comprehensive assessment of the state-of-the-art techniques for detecting, preventing, and mitigating attacks using the SDN paradigm from 2017 to 2022. The significant contributions of this survey are:

- A taxonomy that defines the examined works into three categories. The taxonomy is based on works that leveraged the use of SDN as a Defence Mechanism (SaaDM). Works on security issues in SDN are outside the scope of this survey.
- The taxonomy further divides the publications into several techniques and sub-techniques, which outline the method used.
- The outline of various technologies and techniques integrating with the SDN paradigm to present a defence solution.
- The location where the solution is deployed.
- The defence strategy employed.
- Information relating to the testbed used for proof of concept.
- The main contributions, advantages, and limitations of the examined works.

The rest of the paper is structured as follows: Section 2 gives an overview of the SDN architecture, and Sect. 3 introduces the taxonomy of the study. Section 4 discusses the related works. Section 5 provides details about the research methodology employed. Section 6 provides an overview of the works that have highlighted the deployment of SaaDM on the taxonomy presented in Sect. 3. Section 7 presents a comparative analysis, while Sect. 8 examines the challenges and future work. This survey's conclusion is reported in Sect. 9. The condensed overview of the survey is depicted in Fig. 1.

2 Architecture of SDN

The SDN architecture is divided into three layers: the infrastructure, control, and application layers, often known as the data, controller, and application planes, respectively. The ONF also defined two interfaces for the SDN architecture: southbound interface (SBI) and northbound interface (NBI). Figure 2 depicts the architecture of SDN by highlighting the different layers that make up the paradigm. The below points explain the planes and interfaces of the SDN architecture:

- Data plane: The data plane includes the network element, which receives instructions from the control plane through SBI. In SDN terminology, the devices at this layer are commonly referred to as network switches [33]. Devices in this plane receive instructions from the control plane through well-defined instruction sets managed through pipelines referred to as forwarding tables. By adding new forwarding rules via an abstract interface, the controller instructs the switches how to forward packets. When a packet enters a switch, the forwarding table is checked, and the packet is then forwarded appropriately. The works in [34–36] provide a survey of the data plane as regards its flexibility and programmability.
- Control plane: At the control plane is the SDN controller (identified as the "controller" in this survey). The controller, also known as the NOS, is the core component of an SDN architecture and can serve as an intermediary layer. It is responsible for maintaining a consistent view of the network state, which the control logic then uses to provide different networking services to the other

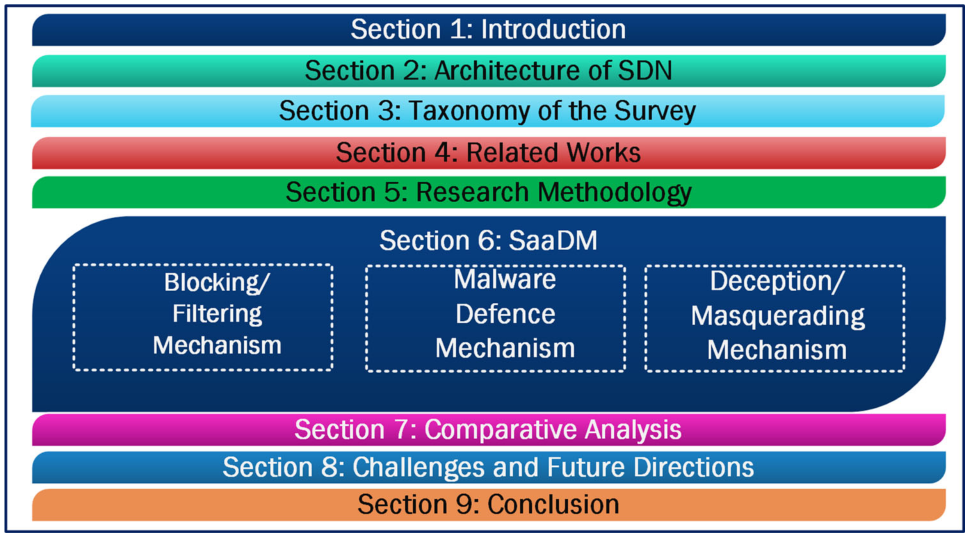

Fig. 1 Condensed overview of this survey on SaaDM

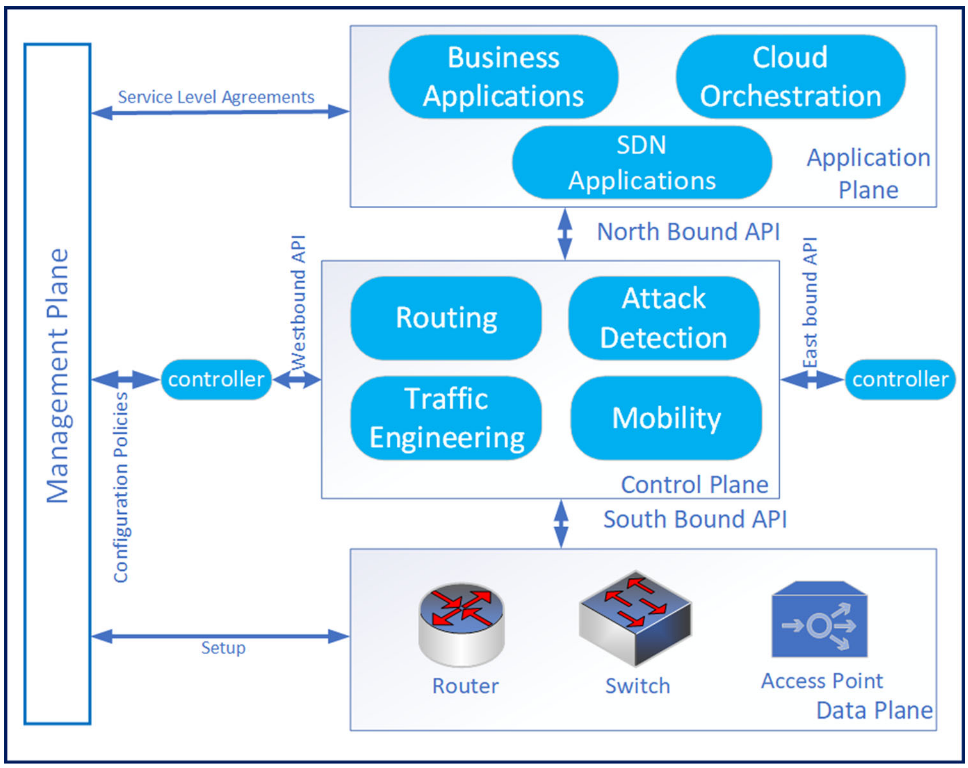

Fig. 2 The SDN architecture showing the management, data, control, and application planes

layers. The controller is software-based and can be programmed using high-level languages such as Java, Python, C, and C++. Comparative surveys and analyses of various controllers are given in [37–41]. The controller centrally manages every switch in the network in conjunction with the SDN applications installed on top of it. This design offers several advantages due to its centralised nature, including efficiently administering the network and responding to changing events. The control plane performs functions such as defining shortest path forwarding, managing devices, managing notifications, managing topologies, managing network statistics received from forwarding devices, etc.

- There are two implementations to logically and centrally manage an SDN paradigm: the implementation of a standalone and a distributed controller. A single controller is deployed to manage the network in a standalone environment. A distributed environment comprises more than one controller managing an aspect of the network, referred to as a network domain or controller domain. Several works [42–45] introduced the concept of a master–slave distributed approach where there are two levels of control, with the top level referred to as the master controller and the slave controllers sends/receives updates from the top level. The controllers can also be deployed side by side in a one-layer approach, as seen in [46–48]. The control layer connects with the infrastructure layer (data plane) using SBI and connects with the application layer using the NBI. Intra-communication between control plane components is achieved using the westbound and eastbound APIs.

- Application plane: This plane provides an interface for external applications and services to interact with the network. This layer is in charge of abstracting the low-level aspects of the network infrastructure and offering a more straightforward, programmatic interface that applications and services may use to create and operate the network. The application plane does not include applications that directly (or primarily) support the functioning of the data plane (such as routing processes within the control plane) [1].
- Management plane: The management plane is typically centralised and communicates with the data, control, and application planes to ensure that the network as a whole runs properly. Management-plane functions are often begun based on an overall network perspective and have historically been human-centric. However, most human intervention has recently been replaced by algorithms. The key responsibilities of the plane are fault detection, monitoring, and configuration management.
- NBI: The application plane communicates with the control plane via an interface, which is referred to as the NBI. The NBI defines the communication between the controller and the services and applications running over the network. Northbound APIs were created primarily so that external management systems and network applications could extract information and manipulate the underlying network and some of its behaviours. Additionally, they make the controller's functionality and the universal network abstraction data model available to network applications. They are employed to promote innovation and have efficient network orchestration [49]. These communications are managed through an API, which is usually RESTful; other APIs are discussed in [49, 50]. Network applications such as firewalls and orchestration apps can be optimised through this interface. The NBI can also integrate the controller with automation software such as Puppet [51], Ansible [52], and Chef [53], as well as an orchestration platform such as OpenStack with the intention of abstracting the internal workings of the network to aid application development that can be integrated into the network [54].
- SBI: SBI establishes a communication connection between the data plane and the control plane. This interface is in charge of relaying instructions from the controller to network devices, as well as collecting data and statistics from the devices for network administration and control. Controller instructions are sent from the SBI over different protocols such as OF, NETCONF POF, P4, and PCEP.

A number of technologies present an enabling environment or use-case for the deployment of SDN. The core enablers of SDN are network function virtualization (NFV) [55–65], 5G [66–73] and network slicing [74–80]. The following paragraphs identify the role of each of these enabling technologies and how they support and improve the SDN paradigm.

NFV is a technology that enables the deployment of network functions on virtual machines (VMs) instead of physical devices by offering a virtualised environment to run the controller and the SDN data forwarding entities (which can be thought of as network functions). It also allows for the dynamic allocation and relocation of resources. It is important to remember that while SDN-NFVs have complementary strengths, they are independent frameworks. In other words, network functions may be delivered and virtualised without SDN and vice versa. Figure 3 shows how SDN-NFV may be coupled to create solutions with the NFV architecture serving as a provider of compute and storage resources to the network functions. The works in [81–83] give a survey on how these two technologies have been integrated to achieve solutions.

Additionally, the SDN-5G convergence has a positive impact on the networking environment. SDN gives 5G networks programmability, automation, and resource optimisation [84]. It allows for flexible resource allocation, effective network slice management, and faster service offerings. However, 5G improves SDN through high-speed connection, low latency, and support for a wide range of applications. Because SDN is programmable, administrators may dynamically regulate and manage network resources in real time, assuring the best performance for various applications and services. Automation capabilities make it easier to install and manage services, lowering the need for manual intervention and increasing operational effectiveness [85]. 5G's high-speed connectivity enables rapid data transmission, benefiting SDN applications that require quick data processing and real-time analytics. The low latency of 5G enables near real-time communication between SDN controllers and network devices, facilitating rapid decision-making and instant network reconfiguration. Furthermore, 5G's support for diverse applications, such as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low latency communications (URLLC), provides SDN with a wide range of use-cases to optimise network resources and deliver tailored services [86]. Figure 4 depicts the architecture of how 5G tends to integrate with SDN, with 5G components deployed at the data plane of the SDN architecture.

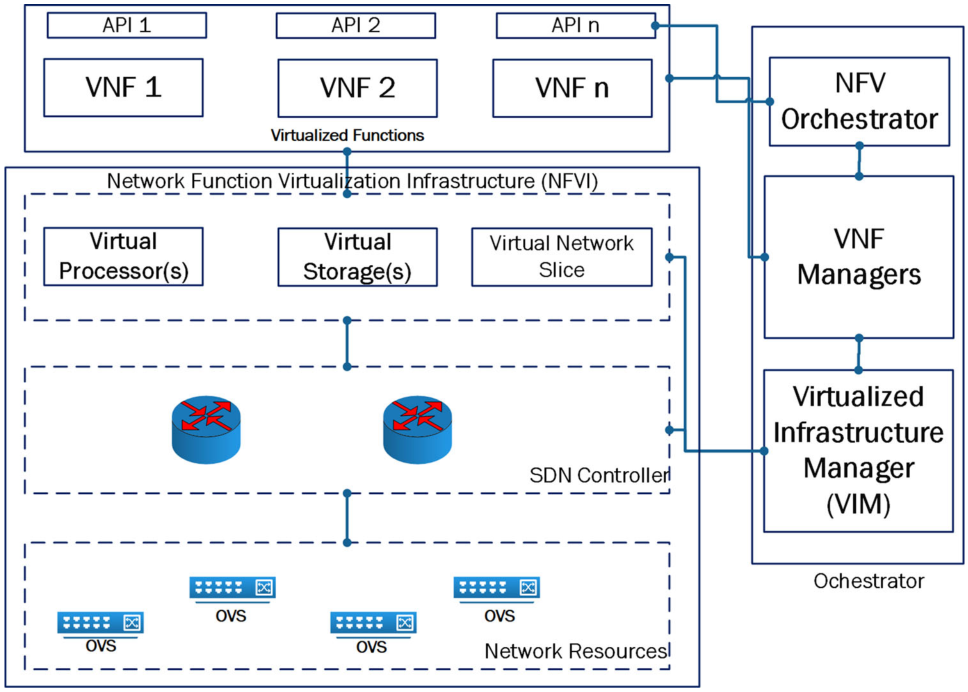

Fig. 3 An NFV-SDN architecture showing the core components

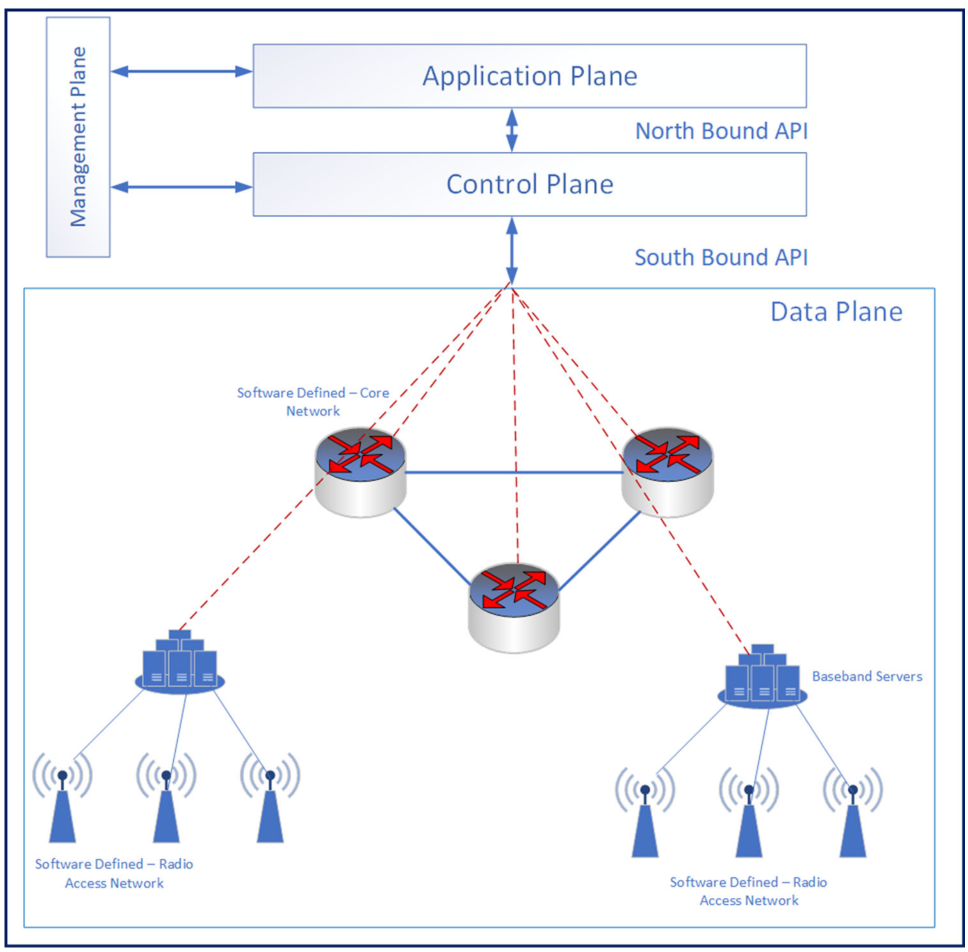

Fig. 4 An architecture of SDN-5G showing the core components

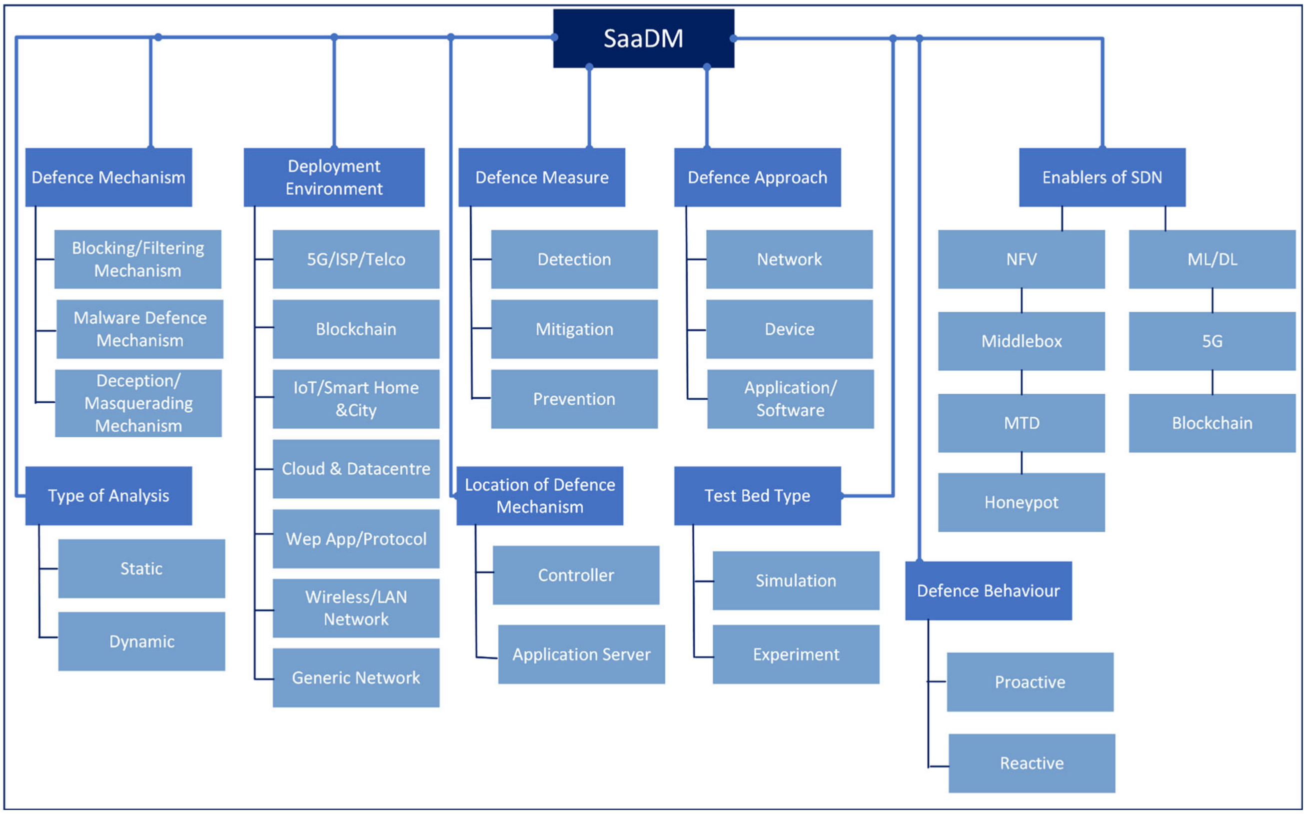

Fig. 5 The taxonomy of the study showing the various categorisation

Network slicing, a key feature of 5G, involves creating multiple virtual networks on a shared physical infrastructure. Each network slice is tailored to specific requirements, such as latency, bandwidth, and security, to accommodate diverse use cases and applications [87]. SDN plays a vital role in network slicing by providing the necessary programmability and orchestration capabilities to define, deploy, and manage these slices. Controllers dynamically allocate resources, set policies, and adjust configurations to ensure optimal performance and resource utilisation within each network slice. 5G complements SDN and network slicing by providing high-speed connectivity, low latency, and support for diverse applications. The high bandwidth and low latency capabilities of 5G networks enhance the performance and responsiveness of SDN applications and services operating within network slices.

3 Taxonomy of the study

This section presents the taxonomy used in this study. Figure 5 illustrates the specifics covered for each work, while Table 2 provides a more thorough perspective on categorising the techniques and sub-techniques used in implementing the defensive mechanism. More information on each of the categories in Fig. 5 is provided below:

- Categorisation based on defence mechanism: The survey highlights three main categories of defence techniques based on SDN and centres this survey on these categories: blocking/filtering, malware defence, and deception/masquerading mechanisms. The particular techniques and sub-techniques that have been identified are listed in Table 2.
- Categorisation based on defence behaviour: There are two identified groups: proactive and reactive behaviours. A proactive approach aims to stop or lessen an attack before it happens, whereas a reactive strategy seeks to do so after it has already happened.
- Categorisation based on defence measures: There are three categories identified: (1) detection, (2) prevention, and (3) mitigation measures. Detection measures identify an attack, prevention measures stop an attack, and mitigation measures lower the severity of an attack after it has occurred.
- Categorisation based on security approach: There are three approaches based on security that are identified: (1) network-based, (2) device-based, and (3) application/software-based. An attack on the network

Table 2 Categorisation of various defence mechanisms

Defence mechanism	Techniques	Sub-techniques	References
Blocking/filtering mechanisms	Network/connection observation techniques	Anomaly detection	[9, 13, 44, 46, 88–102]
		Threshold monitoring	[4, 99, 103–117]
		Connection/network/traffic monitoring	[5, 14, 42, 104, 106, 108, 110, 118–132]
		Packet/network/payload inspection	[111, 133–135]
		Traffic behaviour traceability	[136, 137]
		Port monitoring	[138]
	Criteria-based techniques	Traffic/packet filtering	[8, 14, 88, 139, 140]
		Traffic redirection	[141–145]
		Route creditability/data sensitivity	[146]
		Smart contract	[147–150]
		Cluster head selection	[46, 130]
	Cryptography/authentication techniques	Cryptographic authentication	[90, 91, 151–153]
		Key exchange agreement	[154–156]
		Device/network authentication	[157–161]
		MAC address hashing	[162]
	Entropy-based techniques	Entropy calculation	[9, 104, 116, 117, 137, 163–167]
	Machine learning (ML)/deep learning (DL) techniques	Hybrid models	[13, 94, 168–171]
		Single models	[27, 91, 94, 95, 98, 101, 135, 138, 164, 172–182]
	Traffic duplication techniques	Port/traffic mirroring	[11, 27, 42, 113, 132, 172, 173, 183, 184]
	Statistical techniques	Probability distribution	[110, 137]
		Hellinger distance	[165]
		Bayes filtering	[9, 148]
		Estimated weighted moving average	[116, 117]
		Adaptive correlation analysis	[185]
		Component analysis	[180, 186, 187]
		Time-series algorithm	[138]

Table 2 (continued)

Defence mechanism	Techniques	Sub-techniques	References
	Graph/geometric techniques	Graph model	[115, 150]
		Euclidean distance	[46, 149]
	Optimisation techniques	Brainstorming Optimisation	[178]
		Differential evolution	[188]
		Harmony search Optimsation	[150]
	Point-to-point/many communication techniques	One-to-one mapping of traffic	[143, 167]
		WebRTC	[189]
		publisher/subscriber communication	[190]
Malware defence mechanisms	Network/connection observation techniques	Packet/network/payload inspection	[17]
		Port monitoring	[191]
	Criteria-based techniques	Traffic redirection	[18, 192–194]
		Address listing	[195, 196]
	ML/DL techniques	ML/DL models	[96, 192, 196–199]
	Packet analysis/inspection	DNS inspection	[16]
	Payload extraction/distribution	Payload extraction/distribution	[200, 201]
Deception/masquerading mechanisms	Moving target defence (MTD)		[28, 202–212]
	Honeypot/honeynet/decoy nodes		[195, 213–219]
	Virtual topology deception		[220–225]
	MTD & honeypot		[226–228]
	Other technique		[229]

layer is stopped by a network-based method; a device-based approach stops an attack on a device or hardware, and an application-based system stops an attack on an application/software.
- Categorisation based on the location of implementation: Two locations are identified: (1) the controller and (2) a server. The latter are server-based solutions or applications installed on a server to combat an attack. At the same time, the former are modules or solutions that are installed directly on the controller, or the controller itself acts as a defence mechanism.
- Categorisation based on deployed environment: The proposed survey identifies deployment environments, domains, or environments that use the SDN paradigm to combat cyberattacks. The environments being identified include 5G, IoT, smart homes, industrial control systems (ICS), BC, cloud, data centres, local area networks (LANs), internet service providers (ISPs), smart and microgrids, and general environments, among others.
- Categorisation based on testbed deployments: This survey also identifies the type of environment deployed for proof of concept (PoC). Two categories are identified: (1) experiment-based PoC and (2) simulation-based PoC. Experiment-based settings are based on real-world environments, whereas simulation-based environments are based on software that simulates a real-world setup.
- Categorisation based on type of analysis: In the case of malware defence, this survey identifies the type of analysis deployed to evaluate the malware. The two categories identified are (1) dynamic and (2) static analysis.

4 Related works

Many studies have been carried out in recent years to foster discussion around how SDN can be deployed to combat cyberattacks. Several taxonomies for cybersecurity involving SDN have been proposed in various works or studies. These taxonomies can be broadly divided into two categories: attacks on the SDN architecture or security issues in SDN [230–232] and work leveraging SaaDM. This study focuses on the latter because the former is outside the scope of this survey. The following paragraphs go over relevant studies that use SaaDM.

Yurekten and Demirci [233] classified each threat, defence type, strategy, technique, and deployment information and created a taxonomy for SDN-based solutions for common attacks. The paper examined several attack defence mechanisms, including port scanning, spoofing attacks, sort scanning, low-level DoS attacks, malware, social engineering, sniffer attacks, and web applications.

Silva et al. [234] presented a comprehensive study of the DDoS attack mitigation techniques offered by SDN technology for protecting IoT environments. The survey showed a categorisation of mitigation measures that take the following aspects into account: DDoS mitigation approach, mitigation strategy, types of mitigated attacks, SDN architecture, and assessment methodology.

Mohammed et al. [235] reviewed the current security needs and difficulties in implementing reliable security measures for devices and environments built on IoT technologies. The security of resource-constrained IoT networks is provided by SDN and network function virtualization (NFV) technologies, which were reviewed together with the difficulties, limitations, and future research prospects related to deploying SDN- and/or NFV-based IoT security mechanisms.

Bawany et al. [236] gave a thorough overview of SDN-based DDoS attack detection methods by categorising these methods based on how they are detected and also identifying the advantages and limitations of each technique.

A survey of various SDN-based DDoS attack detection methods was presented by Beslin and Golden [237]. This work discussed how DDoS attacks are grouped into different categories, provided a list of recent DDoS attacks, and then described the other detection mechanisms that were used.

Rawat and Reddy [238] gave an in-depth examination of SDN security challenges. The study examined the risks that can be eliminated by deploying SDN, followed by a discussion of the security threats that can be eliminated by using SDN as well as mitigation measures.

The surveys presented in the previous paragraphs have assessed how SDN has been implemented to combat cyberattacks. Nonetheless, some studies provided a broad overview of the subject while ignoring specific characteristics or features of the various mechanisms, while others focused on a single attack type or deployment environment. The following criteria were developed for comparison with other related works, highlighting certain specific aspects or elements that were overlooked by previous surveys. Table 3 compares the relevant works with this survey based on the following criteria:

- Criterion #1: Did the survey consider the environment in which the deployment of the solution mechanism took place? Not all strategies will be effective in all environments.
- Criterion #2: Did the survey provide information regarding the testbed environment and type?
- Criterion #3: Did the survey review all types of attacks and threats?
- Criterion #4: Did the survey provide information regarding the approach to deployment, i.e., does the solution

Table 3 Comparison criteria of various related surveys with this work

Survey	Criteria								
	#1	#2	#3	#4	#5	#6	#7	#8	#9
[233]	×	×	✓	✓	×	×	✓	×	71
[234]	✓	×	✓	×	×	×	×	×	25
[235]	×	✓	✓	×	✓	×	✓	×	27
[236]	✓	✓	×	×	×	×	×	×	13
[237]	×	✓	×	×	×	×	×	×	8
[238]	✓	✓	×	✓	×	×	×	×	9
This work	✓	✓	✓	✓	✓	✓	✓	✓	147

mechanism combat attacks against the network, device, or application?
- Criterion #5: Did the survey provide information regarding the location of the deployment of the mechanism?
- Criterion #6: Did the survey identify the defence behaviour?
- Criterion #7: Did the survey identify the defence measure?
- Criterion #8: Did the survey identify the enabling technologies that combine with SaaDM?
- Criterion #9: Number of solution mechanisms that were reviewed.

5 Research methodology

In this section, we present the methodology used in the survey to review the current state of the art. A systematic mapping study (SMS) was conducted [239]. SMS is comparable with secondary study that aims to provide an overview of a topic as reported in primary studies. A systematic map describes research patterns across time, identifies potential research gaps and focus points, and provides a synthesis of the selected subject. We propose a number of research questions in tandem with the categorisation defined in Sect. 3 that this work will try to investigate:

- RQ1: Is SaaDM able to effectively combat cyber-attacks or IT security issues?
- RQ2: What are the main threats/attacks SaaDM can combat?
- RQ3: What are the most common defence mechanisms used?
- RQ4: What are the most common defence behaviours identified?
- RQ5: What are the most common defence approaches identified?
- RQ6: What are the deployment environments identified?

Table 4 The keywords used for database search

Sn	Keywords	
1	SDN AND cyberattacks	Software Defined Network AND cyberattacks
2	SDN AND cybersecurity	Software Defined Network AND cybersecurity
3	SDN AND IT security	Software Defined Network AND IT security
4	SDN AND network security	Software Defined Network AND network security
5	SDN AND device security	Software Defined Network AND device security
6	SDN AND application security	Software Defined Network AND application security
7	SDN AND cloud security	Software Defined Network AND cloud security

- RQ7: What type of environment is leveraged to test the defence solution?
- RQ8: What is the preferred location for implementing the defence solution?

This work selected relevant studies relating to the deployment of SaaDM. The search was conducted in three stages:

- Identifying the keywords: This stage focused on selecting the most appropriate keywords to aid in selecting the most appropriate articles/work. Software Defined Networking and the acronym – SDN were used interchangeably with other keys like "cyberattacks", "cybersecurity", "IT security", "network security", “device security”, “application security” and “cloud security”. Table 4 shows a list of these keywords. Also, the Boolean operation "AND" was used in the search to connect Software Defined Network or SDN with the other keys.
- Selection of sources or database: At this stage, we explored well-known academic databases based on the keywords

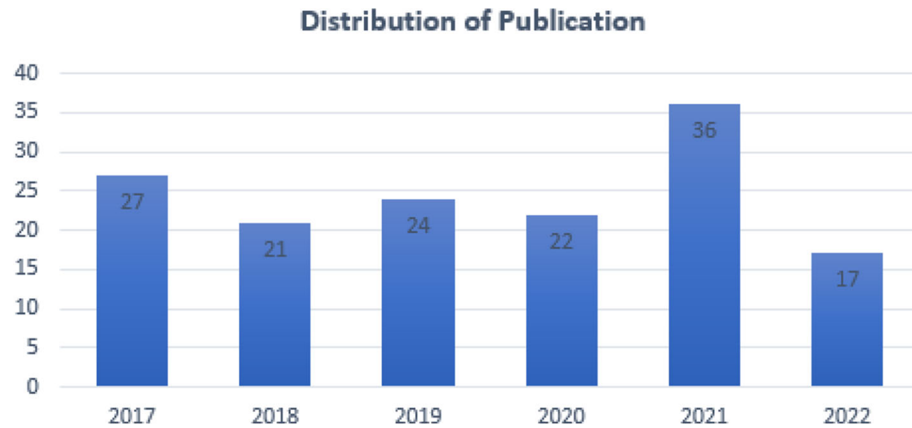

Fig. 6 Number of publications referenced in this survey grouped by publication year

defined above. These databases include IEEE Xplore [240], Google Scholar [241], and RefSeek [242].

- Inclusion and exclusion criteria: At this stage, we select the paper based on an inclusion criterion: (1) If the year of publication is between 2017 and 2022, (2) If the work leveraged the use of SaaDM. It should be noted that this survey does not cover security issues in SDN. The distribution of these works from 2017 to 2022 is seen in Fig. 6.

6 SDN as a defence mechanism

This section provides an extensive overview of the various strategies listed in Table 2 (Sect. 2) including using SDN exclusively and combining it with other technologies/techniques to combat cyberattacks. The strategies are divided into three main categories as seen in Fig. 7, which serve as the foundation for the discussion in this Section: traffic blocking/filtering, malware defence, and deception/masquerading mechanisms. Mechanisms for traffic blocking and filtering are used to manage and control network traffic in order to stop malicious or unauthorised traffic from compromising a system or environment. Many of the traffic blocking/filtering strategies addressed in Sect. 6.1 focused on the availability of the environment, but very few addressed integrity or confidentiality, which are the fundamentals of the CIA Triad, offering an indication of the controlled and regulated approach to network traffic blocking and filtering. As observed in the majority of the work reviewed in Sect. 6.2, malware defence mechanisms protect systems and networks from malicious software or programmes including viruses, worms, Trojan horses, and the most notorious, ransomware. The majority of the works in Sect. 6.2 tackled security issues that affects confidentiality, integrity, and availability, while Sect. 6.3 solely deals with the availability of the environment through deception/masquerading mechanisms. The use of deception or masquerade is done to lure, imitate, or trick. In this situation, the majority of the works covered in Sect. 6.3 uses a decoy system to lure attackers or trick them, guaranteeing a secure environment.

6.1 Blocking/filtering mechanisms

Today, various industries, organisations, research centres, institutions, military commands, and businesses install and configure various application and network security appliances that act as gateways for inbound and outbound traffic, thereby acting as a barricade between a trusted zone (sometimes referred to as an internal network) and an untrusted zone (outside network), thus reducing the attack surface for malicious attacks. Most of these security appliances monitor connection ports, allowing or denying connections based on predefined rules (also known as access control list—ACL) and must not be susceptible to penetration as defined by Cheswick et al. [243]. Firewalls are classified into various types—packet filtering, circuit gateway/proxy, and application gateway/proxy based on the exact functions they perform or mode of operations [243–246]. With the advent of SDN, the security realm has seen a rapid shift toward the design and implementation of various defences using SDN. SDN provides a standard interface between the forwarding and the control devices, easing overall network management and programming by providing a new degree of visibility which are lacking in traditional firewalls because they are inflexible, vendor-specific, and expensive as outlined in Table 1. Katwal and Sood [247] and Satasiya and Rupal [248] gave some insight into how SDN can provide better results than traditional firewalls with the ease of changing and the updating of configuration in the control plane and having the change appear on the complete networking system, whereas in a traditional network setup, if a configuration update is necessary, a network expert must manually log on to all devices. The sections that follow go over the various filtering/blocking techniques and sub-techniques indicated in Table 2. Table

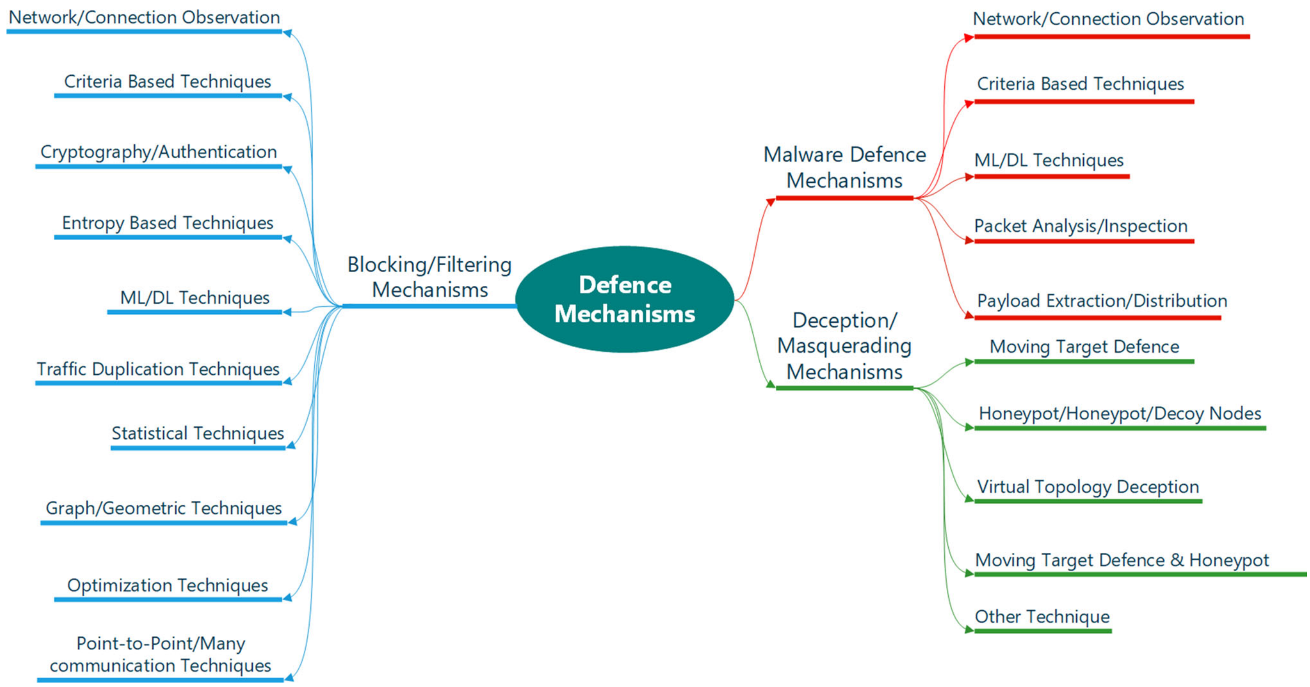

Fig. 7 A categorisation of the defence mechanisms

5 also provides a comprehensive summary of all proposed systems, including the CIA Triad [249] that the solution addressed, the enabling technology that integrated with SDN, and the attack type against which the solution was designed or tested. Table 8 summarises the main contributions and/or advantages, whereas Table 9 summarises the disadvantages and/or limitations related to a number of the literatures.

6.1.1 Network/connection observation techniques

Network/connection techniques are methods for collecting information on network traffic, activity, and performance through the analysis of network data. SDN may be used to improve network security monitor and resolve issues. Several publications offered techniques for protecting an environment based on observations of network and connection activity utilising SDN. The following paragraphs highlight network/connection observation-based sub-techniques such as anomaly detection, threshold monitoring, network/payload inspection, traffic tracing, and port monitoring.

- Anomaly Detection: SDN is used in anomaly detection to detect aberrant or unexpected activity or behaviours that deviate from normal patterns. Unexpected network traffic patterns, unauthorised entry attempts, unexpected system behaviours, and anomalous system settings might all be examples of unusual activity. The works in [14, 18, 40, 41, 43–46, 48–52, 54–56] proposed solutions based on detecting anomalous behaviour, which identifies patterns that do not conform to the network's typical behaviour.
- Threshold Monitoring: Threshold monitoring is a rule-based technique in which limits or thresholds (whether specified or based on system behaviour) are set. When the threshold is surpassed, an alert informs the controller of a possible attack or harmful behaviour. The method used to identify suspicious behaviours distinguishes anomaly detection from threshold monitoring. Unlike anomaly detection, which is dynamic, threshold monitoring is static, which means that the limit for threshold monitoring is predetermined, whereas the limit for anomaly detection is dynamic. Nevertheless, there is the potential for defining dynamic threshold monitoring, as seen in [116] and [117], where thresholds are not pre-assigned and are decided by the system's overall behaviour, while the works in [4, 99, 103–115] are based on specified/static thresholds. Threshold monitoring systems are sometimes coupled with other technologies, such as middleboxes [104, 108, 113, 117] and occasionally with anomaly detection, as shown in [104], to detect an attack.
- Connection/Network/Traffic Monitoring: Connection monitoring tracks connections between two devices in a network using IP addresses, ports, protocols, etc., in real time. These sub-techniques involve analysing traffic volumes and patterns for potential security threats. The solutions proposed in [5, 14, 104, 106, 108, 110, 118–120, 122, 123, 125–129] are based on monitoring the traffic between a source and destination, such as collecting

Table 5 Specifics of works based on blocking/filtering techniques

References/framework	Attack type	Measures	Approach	Defence behaviour	Location of deployment	SDN testbed	Deployment environment	Enabling technology	CIA triad tackled
[4]	DoS	DP	N	R	CS	Si	Web protocol	NS	A
[146]	Generic	DP	D*	P	C	NS	IoT	VPN technology	CoI
[147]	DDoS	DMP	D*	R	C	NS	IoT	Blockchain	CoIA
[150]	Generic	DP	N	R	C	Si	Datacentre	Blockchain	CoI
[151]	Generic	D	N	R	C	Si	5G	NS	CoI
[152]	Generic	DP	N	R	C	E	5G	Blockchain, NFV	CoIA
[155]	Eavesdropping, MiTM, IP spoofing, Replay attacks	DM	ND*	P	C	Si	5G	NS	A
[5]	DDoS	DM	N	R	C	Si	Generic	NS	A
[156]	Generic	DM	ND*	R	C	E	5G	NS	AI
[159]	ARP spoofing, port-stealing, DHCPv4 spoofing	DP	N	P	C	Si	LAN	NS	A
[161]	Generic	DP	N	RP	C	Si	LAN	NS	AI
[163]	Dos, DDoS	DM	N	R	C	Si	IoT	NS	A
[164]	DDoS	DM	N	R	C	Si	Web protocol	ML	A
[165]	DDoS	DM	A	R	C	Si	Web protocol	NS	A
[166]	DDoS	DM	N	R	C	Si	Web	NS	A
[167]	DNS Amplification	DM	N	R	C	Si	Blockchain	NS	A
[168]	Bot, DDoS, Brute force	D	N	R	C	NS	IoT	DL	A
[170]	Generic	D	N	R	C	NS	Smart Homes/Cities	DL	A
[171]	Phishing	DM	N	R	CS	Si	Web Protocols	ML	Co
[172]	DDoS	DM	N	R	C	Si	IoT	DL	A
[174]	MiTM, DDoS, Side channel, malicious code	DM	N	R	C	E	IoT	Fuzzy Neural Network	A
[177]	Generic	D	N	RP	C	Si	Generic	ML	A
[178]	Generic	D	N	R	C	Si	Generic	ML	A
[179]	Phishing	DP	N	R	C	Si	LAN	ML	Co
[180]	DDoS	D	N	R	CS	Si	Web Protocols	ML,DL	A

Table 5 (continued)

References/framework	Attack type	Measures	Approach	Defence behaviour	Location of deployment	SDN testbed	Deployment environment	Enabling technology	CIA triad tackled
[181]	Generic	D	N	R	CS	NS	Blockchain	ML	Co
[183]	Generic	DP	N	R	CS	NS	LAN	Middlebox	A
[184]	DDoS	DM	N	R	CS	Si	Generic	Middlebox	A
[186]	DDoS	DM	N	R	C	Si	Datacentre	ML	A
[14]	DNS Amplification	DM	N	R	C	Si	Blockchain	ML	A
[188]	Generic	DM	N	R	C	Si	Smart Grid	Middlebox	CoIA
[189]	Rogue APs	DP	N	R	C	Si	LAN	WebRTC	A
[190]	Generic	D	N	R	C	Si	Smart Grid	NS	A
[11]	DDoS	DP	ND*	RP	C	Si	LAN	ML	A
[13]	DDoS	D	N	R	C	NS	IoT	DL	A
[14]	DoS/DDoS	DP	N	R	C	Si	LAN	NS	A
[27]	DDoS	DP	N	R	C	E	LAN	ML	A
[88]	DDoS	DM	N	R	C	NS	ISP	NS	A
[89]	DDoS	DM	N	R	CS	NS	ISP	Middle box	A
[91]	Generic	DM	ND*	R	C	Si	5G	ML, Cryptography, NFV	CoIA
[93]	Generic	DM	N	R	C	Si	IoT	ML	A
[94]	DDoS, Brute Force, Port Scan	D	N	R	C	NS	IoT	ML	A
[46]	Generic	DM	ND*	R	C	Si	IoT	Blockchain, NFV	CoIA
[97]	DDoS	DM	ND*	R	C	Si	IoT	Blockchain, Machine Learning	CoIA
[98]	Generic	DM	ND*	R	C	Si	IoT	ML	A
[44]	Dos, DDoS	DM	ND*	R	C	Si	IoT	ML	A
[100]	ARP spoofing, Blacklisted MAC Addresses	DP	N	R	S	E	Generic	Middle box	A
[101]	Generic	DM	ND*	R	S	Si	Generic	ML, Middle box	A
[102]	Generic	DP	N	R	C	Si	Microgrid	NS	AI
[103]	DDoS	DM	ND*	R	C	Si	5G	Fog Computing	A

Table 5 (continued)

References/framework	Attack type	Measures	Approach	Defence behaviour	Location of deployment	SDN testbed	Deployment environment	Enabling technology	CIA triad tackled
[104]	Generic	DP	N	R	C	NS	5G	ML	A
[105]	Generic	DP	N	R	C	E	IoT	Blockchain, Edge Computing	Co
[106]	DDoS	DM	N	R	C	Si	IoT	NS	A
[107]	DDoS	DM	N	R	C	E	IoT	NS	A
[108]	DDoS	DM	N	R	C	Si	IoT	Middle Box	A
[110]	DDoS	D	N	R	C	NS	Cloud Computing	NS	A
[113]	DDoS	DM	N	R	C	E	Web protocol	Middle Box	A
[114]	DDoS	DM	N	R	C	Si	Web protocol	NS	A
[118]	DDoS	DM	N	R	C	Si	5G	NS	A
[119]	DoS, DHCP related threats, Eavesdropping	DM	N	R	C	E	5G	NS	A
[120]	DDoS	DP	N	R	C	E	5G	NS	A
[122]	DDoS, IP spoofing attack, flow table overloading attack	DM	N	R	C	Si	5G	ML, NFV	AI
[123]	Botnet	D	N	R	C	E	5G	NFV, Middle Box	A
[124]	Generic	D	N	R	C	E	5G	NFV, Middle Box	A
[125]	ARP Poisoning	DP	N	R	C	Si	LAN	NS	A
[128]	Generic	DP	N	R	C	Si	Blockchain	NS	Co
[130]	DDoS	DM	N	RP	C	Si	Smart Grid	Blockchain	CoIA
[42]	Generic	DM	N	R	CS	Si	Generic	Middle Box	A
[131]	DoS	DM	N	R	C	Si	Generic	sFlow	A
[134]	DDoS	DP	N	R	C	E	Generic	NS	A
[135]	Phishing	DM	NA	R	C	E	Web Protocol	NS	Co
[136]	Generic	DM	N	R	C	NS	IoT	NS	CoI
[140]	DDoS	D	N	R	C	Si	Generic	NS	A
[143]	DNS amplification	DP	N	R	C	Si	LAN	NFV	A

Table 5 (continued)

References/framework	Attack type	Measures	Approach	Defence behaviour	Location of deployment	SDN testbed	Deployment environment	Enabling technology	CIA triad tackled
AutAvailChain [153]	Generic	DP	N	P	C	Si	IoT	Blockchain	CoIA
Bloc-Sec [90]	Generic	DM	ND*	R	C	Si	5G/IoT	Blockchain, Neural Networks	Co
BotSifter [176]	Botnet, SPAM, port scanning, DDoS, click fraud	DM	N	R	C	NS	Datacentre	ML	A
CAAMP [141]	DDoS	DM	N	R	C	Si	Datacentre	NFV	A
CAuth [160]	IP spoofing	DM	N	R	S	Si	LAN	NS	A
Cochain-SC [148]	DDoS	DM	N	R	C	Si	Blockchain	ML, Blockchain	CoA
DistB-Condo [149]	Generic	DM	N	R	S	Si	Smart Homes/Cities	Blockchain, NFV	CoI
DNSxD [111]	DNS Exfiltration	DM	N	R	C	Si	LAN	NS	A
DSM-SVM [187]	DDoS	DM	N	R	C	Si	Generic	ML	A
DTARS [92]	DDoS/Botnet	DM	N	R	C	Si	5G/IoT	NFV, Neural Network	A
FlexProtect [142]	DDoS	DM	N	R	C	Si	Datacentre	NFV	A
FlowTrApp [109]	DDoS	DM	N	R	C	Si	Datacentre	sFlow	A
FORT [138]	DDoS	D	N	R	C	Si	Generic	ML	A
HostWatcher [144]	DDoS	DM	N	R	C	Si	Datacentre	NS	A
HYBRID-CNN [169]	Generic	D	N	R	C	NS	Smart grids/micro grids	DL	A
IDSIoT-SDL [95]	DoS, DDoS, Brute force, Heartbled, Botnet	D	N	R	C	Si	IoT	DL	A
LFADefender [126]	DDoS	DM	N	R	C	Si	LAN	NS	A
NFGA [162]	ARP Spoofing	DM	N	R	S	Si	LAN	NS	A
PCSS [154]	Generic	DP	D*	R	C	E	Smart Homes/Cities	NS	CoI
PortEdge [175]	DDoS, DoS, Fuzzing, Port scanning	DM	N	R	C	NS	IoT	DL	A
RADAR [185]	DDoS	DM	N	R	C	Si	Generic	NS	A
R-IDPS [173]	DDoS	DM	N	R	C	Si	IoT	ML	A

Table 5 (continued)

References/framework	Attack type	Measures	Approach	Defence behaviour	Location of deployment	SDN testbed	Deployment environment	Enabling technology	CIA triad tackled
SDIoT-DDoS-DA [96]	DDoS	DM	N	R	C	E	IoT	ML	A
SDNDefender [137]	DDoS	DP	N	R	C	Si	Generic	NS	A
SDN-SF [99]	Generic	D	ND*	R	C	NS	IoT	NS	I
S-DPS [117]	DDoS	DM	N	P	CS	Si	Smart Grid	Middle Box	A
SEAL [116]	DDoS	DM	NA	R	C	Si	Smart Home	NS	A
SEAL [129]	DDoS	DM	NA	R	S	Si	Smart City	NS	A
SHDA [112]	DDoS	DM	N	R	C	Si	Web protocol	NS	A
SHSec [115]	Generic	DM	N	R	C	NS	Smart Home	NS	A
SIS [158]	Malware Injection, DDoS, Spoofing/masquerading, MiTM	DM	D*	P	C	Si	NS	NS	CoIA
SLICOTS [127]	DDoS	DP	N	R	C	Si	LAN	NS	A
TENNISON [132]	DDoS, Scanning	DM	N	R	S	Si	Generic	NS	A
[182]	DDoS	DM	N	R	C	Si	ISP/Telco	ML	A
[145]	DDoS	DP	N	P	C	Si	Generic	NS	A

Measures: *D* detect, *M* mitigate, *P* prevent | Approach: *N* network, *D** device, *A* application | Defence behaviour: *R* reactive, *P* proactive | CIA triad: *A* availability, *I* integrity, *Co* confidentiality | Testbed: *Si* simulation, *E* experiment, *NS* not specified | Location of deployment: *C* controller, *S* server device status, traffic data, and connection handshakes that can aid in the detection of an attack. These methods rely on forwarding devices to collect statistics and transmit them to the controller for analysis. Most monitoring-based solutions are reactive and block an attack at the network rather than the device or application level. However, in the instances of [14] and [130], these strategies work reactively and proactively. As seen in [42, 108, 123–125, 131, 132], the connection/traffic monitoring approach relies on middleboxes such as IDS, IPS, and DPI, while some are deployed with protocols such as sFlow as seen in [42, 126, 131]. A combination of a middlebox and sFlow was proposed by [132].

- Packet/Network Inspection: Packet/network inspection is based on analysing headers and contents to determine the origin, destination, protocol, and any other important information. Packet/network inspection is a rule-based technique that matches a traffic's header/content to a pre-defined rule. Although packet/network inspection and anomaly detection appear to be similar, packet inspection is performed at the packet level, whereas anomaly detection is performed on the entire network. While both methods can detect suspicious behaviour, packet inspection provides more detailed information about individual packets. The works in [90, 111, 134] offered solutions based on packet/network inspection, which involves inspecting the header and the data part of the packet/network that is transmitted by the inspection engine/module. The approach by [111] is based on inspecting DNS-based data exfiltration using the SDN architecture by creating two flow tables: the first table has two flow rules to route DNS requests to the controller, and the other rule redirects traffic to the second flow table to handle non-DNS traffic. Abdulqadder et al. [90] proposed a lightweight security architecture that ensures that all IoT devices must register their credentials before access to the network is granted. The controller handles the traffic through packet header and packet content inspection using spiking dual fizzy neural networks. In another paper, [134] proposed a distributed firewall based on SDN that performs stateless inspection by creating OF rules that are applied on switches to prevent lateral movement by an attacker. In a different approach by [135], the system operates in two modes: "store and forward" and "forward and inspect," with the former simply storing information, inspecting, and classifying it for analysis, and the latter forwarding packets to destinations while keeping a temporary copy for inspection.
- Traffic Behaviour Traceability: Individual traffic and its associated metadata can be tracked across the network using traffic traceability, which aids in detecting potential security threats. This method is similar to the connection/network monitoring discussed above; however, connection monitoring focuses on the connections and communications between network nodes and provides high-level detail, whereas traffic traceability provides a more detailed analysis of individual traffic. Wang et al. [136] proposed an ID-based SDN secure network architecture for IoT Big data based on improving user identity recognition for network traffic. The solution works by verifying the source address of the devices, ensuring that the source IP of the terminals cannot be forged by introducing behaviour traceability based on RFC 5210 [250] and RFC 7039 [251] and in a similar study to [136, 137] proposed SDNDefender which is a method that uses a dynamic entry match algorithm to prevent IP spoofing and TCP SYN flooding by ensuring that the source address of an incoming packet is verifiable. The algorithm matches a packet using path-based routing rules and decides on the next course of action by utilising SDN's centralised administration.
- Port Monitoring: Port monitoring entails identifying and analysing network traffic transmitted across certain network ports. Each network port is connected to a certain protocol or service and is used to identify specific network services or applications operating on a device. Jia et al. [138] introduced FORT. FORT is a DDoS detection approach that distributes a rule-based detection algorithm at edge switches and chooses whether to start the detection scheme by monitoring the port and collecting the port status regularly. A support vector machine (SVM) method is implemented to determine whether a DDoS attack happens, and a time-series method is used to generate port data adaptively.

6.1.2 Criteria-based techniques

Criteria-based techniques enable the creation of alerts when particular conditions/criteria are satisfied. The matching requirements might be based on matching signature patterns, traffic analysis, or behavioural characteristics. This section identifies a variety of sub-techniques that are based on meeting specific requirements and are either managed or deployed utilising the SDN paradigm.

- Traffic/Packet Filtering: Traffic/packet filtering is a criteria-based technique that provides network security by filtering network traffic based on many criteria, such as rate-limiting and reverse lookup. Filtering could be applied to an inbound/outbound port, service, or server. Ezekiel et al. [88] and Vempati et al. [140] proposed a rate-limiting approach to stop or minimise malicious traffic by utilising the controller. Vempati et al. [140] proposed closed-loop feedback that adjusts system characteristics through disturbance rejection by limiting the impact of an attack by ensuring the network remains consistent with some specified service-level agreement. At the same time, [88], in a similar approach, applies rate limiting when it is impossible to identify the attack traffic from genuine traffic during a suspicious attack by an ISP and further employs traffic redirection to a scrubber for deeper analysis of the traffic. A firewall (ChainGaurd) was proposed by Steichen et al. [8] for blockchain nodes to filter unauthorised communications and add security to the BC nodes by analysing the source of traffic to distinguish between legitimate and illegitimate traffic through routine peer information retrievals from the guarded BC nodes. The firewall uses a list to filter traffic (a whitelist, blacklist, or grey list). Rezaei and Hashemi [14] proposed a system that performs the function of monitoring and distributing firewall rules in form of ACLs in an OF context, thereby facilitating effective filtering of traffic and ensuring dynamic management of the network. The two approaches for ACL distributions employed are proactively or reactively. A proactive approach broadcasts the controller's new flow table entries to all OF switches, whereas a reactive approach requires the OF switches to contact the controller each time to get a new rule. Badotra and Singh [139] proposed a firewall that operates by filtering packets at levels 4 and 7 of the OSI [252] model and also employs a list-based approach (blacklist and whitelist) similar to [8] to prevent traffic originating from known malicious hosts. According to [139], the approach to traffic filtering is that all traffic is allowed, all traffic originating from a known bad site is prohibited (Blacklist method), and all traffic is prohibited, and only traffic from recognised, reputable sites is allowed (Whitelist method).

- Traffic Redirection: Traffic redirection ensures the rerouting of network traffic from one destination to another based on specific criteria when a potential security threat is discovered. The works in [141–145] proposed solutions based on traffic redirection for further action to be carried out. Beigi-Mohammadi et al. [141] proposed a solution to install a copy of the application's topology ("shark tank") in a private environment isolated from the rest of the network when suspicious traffic is detected. In a similar approach, [142] proposed using a virtual network function (VNF) to initiate a service-specific defence mechanism to analyse malicious traffic close to the source of the attack. Kim et al. [143] proposed using the controller to prevent DNS amplification attacks by redirecting/forwarding DNS query records to an external database server when the controller discovers that the local switch is out of storage to store DNS query records. Yuan et al. [144] proposed the deploying of caching to transfer traffic to a data centre host at a slower pace. All traffic approaching a host is redirected to the cache first, thereby managing traffic flow and preventing DDoS-related attacks. Also, by delivering an HTTP redirection to the client, [145] ensures that all HTTP are inspected. This message instructs the client to use the true Web application's IP address. As a result, assuming the botnet nodes do not implement the whole HTTP protocol, this redirection will be ignored.
- Route Creditability/Data Sensitivity: Gheisari et al. [146] proposed a novel solution to privacy preserving IoT device packet management by employing the controller to split data based on context. Based on the information received by the controller, devices divide data into two sections based on data sensitivity. When the sensitivity is set to high, the controller routes traffic in the most secure way. When the device's sensitivity is low, the controller will instruct the device to split the data into two parts and send them via separate secure pathways.
- Smart Contract: BC has also found its way into cyberattacks by combining it with the SDN paradigm. A smart contract is a digital contract that is encoded on the BC platform and automatically executes the terms of the contract when certain conditions are met. Smart Contracts ensures confidentiality since it is immutable and tamperproof. A number of works have proposed the use of BC with SDN to handle security issues, as seen in [147–150]. El Houda et al. [148] proposed a strategy for mitigating DDoS attacks at two levels: intra- and inter-domain. The proposed method employs the concept of smart contracts to work with several SDN-based domains in threat mitigation. Furthermore, [147] and [149] propose solutions by allowing smart contracts to leverage flow rules to enforce security policies and monitor suspicious traffic. Each segment of the IoT network is designed to have a controller which in turn is embedded with smart contracts to enforce privacy and maintain trust within the IoT network. Pourvahab and Ekbatanifard [150] proposed a forensic solution that employed smart contracts to assist users in tracking their data in a cloud environment. The controller's responsibility is deploying flow rules depending on the status of the network and collecting all data from the cloud service provider.
- Cluster Head Selection: Islam et al. [46] and Xiong et al. [130] proposed solutions relating to electing a cluster head (controller) in an environment that has multiple controllers deployed. Islam et al. [46] proposed an energy-aware distributed and decentralised BC-SDN architecture for IoT. IoT devices can request access to the controller, which then registers the device and assigns it an IP address, preventing it from being registered in another cluster. Xiong et al. [130] developed a distributed architecture for a smart grid in which an SDN cluster head is selected and used as a blockchain node in a distinct approach. The cluster head (controller) confirms the request for domain identification for a node based on database information.

6.1.3 Cryptography/authentication techniques

This section describes how SaaDM protects information and ensures data integrity and confidentiality. SDN has merged with various technologies, most notably BC, to provide secrecy through encryption and authenticity through a flexible and programmable architecture. Cryptography/authentication techniques leveraging the SDN paradigm are discussed below.

- Cryptographic/Key Exchange Agreement: This technique establishes a shared secret key between two parties during communication. The secret key provides encryption and decryption of messages between the parties, thereby ensuring confidentiality and integrity. The works in [90, 151–156] proposed solutions based on cryptographic approaches for an IoT and 5G environment. Based on three factors, [90] provides a lightweight security architecture for IoT networks. The first factor depends on the user ID, password, and a random prime number. The second factor is the retina, stego image, and finger vein, while the third factor is a physically unclonable function. These elements are shown to users in the order listed. The works in [151] and [150] proposed a system that provides end-to-end security based on elliptic curve cryptography, while the use of a symmetric key, hash function, and hash value deployed in an SDN-enabled 5G environment was proposed by [152, 153] and [91], respectively. The use of key exchange and keys for authentication was presented by [154–156] by providing mutual user, controller, and smart device authentication and preventing an attacker from learning and altering data during transmission.
- Device/Network Authentication: The works in [158–161] propose techniques that ensure network and device authentication before granting access to other network resources or the environment. Karmakar et al. [158] proposed a security architecture that controls network access to verified IoT devices while applying fine-grained limitations to protect IoT network infrastructure flows. Authorisation is carried out using a dynamic, policy-driven approach. Rietz et al. [159] proposed a comparable authentication approach to [158], in which the controller implements port authentication and unlocks after successful authentication. To secure DNS servers, [160] devised a method to detect IP spoofing. While verifying a legitimate inquiry, the process automatically blocks any fake query. The technique detects the faked DNS query packet using the modified packet as an authentication mechanism. Nife et al. [161] proposed a solution that uses SDN's centralised and programmable capabilities to secure a network and its components. To verify the user's identity, 802.1x authentication relies on the Extensible Authentication Protocol (EAP), similar to that proposed in [159].
- Mac Address Hashing: Cox et al. [162] presented an ARP network flow guard that can detect and mitigate ARP spoofing. The module operates by obfuscating the MAC address of a host with an appropriate IP and port association. In order to build a table of *MAC:IP:portfixed:state* associations for each LAN device, the system keeps track of DHCP offers, requests, and acknowledgements.

6.1.4 Entropy calculation techniques

Several works employed the entropy calculation to measure the randomness or unpredictability of network traffic by leveraging the SDN environment in deploying a flexible architecture. Entropy calculation works by analysing the frequency of distribution and calculating the probability of each value to detect the occurrence of an attack. The works in [9, 104, 116, 117, 137, 163–167] proposed solutions that leveraged SDN and applied entropy calculation to mitigating attacks. The works in [9, 104, 117, 163] quantified the unpredictability of network flow over a given period by computing the entropy value. All the works relied on the use of source IP, with [104, 117, 163] relying on additional metrics such as the source port, destination address, and port.

The works in [116] and [137] implemented a lightweight entropy-based method for detecting attacks by effectively determining the unpredictability of network traffic flows by relying on the destination IP address only, in an opposite solution to [9]. Galeano-Brajones et al. [163] proposed the use of entropy calculation during the network monitoring stage to obtain information about the network, feeding an entropy calculation algorithm-based detection method. The information analysed includes source/destination IP, source/destination port similar to [104, 117, 163], and an additional metric—protocol information. Sumantra and Gandhi [166] proposed using Shannon's entropy to reduce DDoS attack utilising the source IP, destination IP, and an additional metric—the total number of requests issued to measure the uncertainty or unpredictability of an occurrence. El Houda et al. [167] proposed an entropy calculation scheme to detect illegitimate flows and contain them in the space of the genuine requester.

The works in [164] and [165] proposed using entropy calculations and additional techniques to calculate the unpredictability of network traffic. Mohammadi et al. [165] employed entropy calculation and further employed Hellinger distance (HD) to determine the source of anomalous behaviour, while [164] proposed a detection procedure that is split into two steps. First, the IP entropy is determined to see whether a DDoS attack has been generated. The system sets the flag to 1 if the IP entropy detection result is a DDoS attack. When the flag is set to 1, the traffic collection module extracts features from the flow table entries and message packets.

6.1.5 Machine learning/deep learning-based techniques

Machine learning (ML) and deep learning (DL) have gained much traction in mitigating cyberattacks when combined with an SDN architecture or paradigm. Hybrid and single models are the main models used to combat an attack. A single machine learning model is a single algorithm or approach to learn patterns from data and generate predictions. In contrast, hybrid machine learning models incorporate different algorithms or approaches to boost performance and accuracy, which may include combining many models. In complicated issues where a single method may only capture some of the significant patterns in the data, hybrid models can be very beneficial.

The works in [13, 168–171] proposed hybrid models to mitigate an attack. Javeed et al. [13] proposed a hybrid DL algorithm based on the Cuda-Deep Neural Network Gated Recurrent Unit (CuDNNGRU) and Long Short-Term Memory (CuDNNLSTM) algorithms. The model result is compared using Cuda-Deep Neural Network Gated Recurrent Unit (CuDNN-GRU) and Cuda-Bidirectional Long Short-Term Memory (CuBLSTM). In another work, Javeed et al. [168] proposed two hybrid models—Cu-DNNGRU and Cu-BLSTM—and compared these with the model in [13]. In another work, [170] proposed the DNNLSTM model to mitigate attacks in an IoT environment, and in another position, [169] proposed a model based on a hybrid CNN to detect an attack in a smart grid by classifying abnormal flow. Miao and Wu [171] presented an ensemble learning technique based on stacking to detect phishing URLs by stacking three models.

The works in [27, 95, 98, 101, 135, 138, 164, 172–180, 182] proposed single models to combat attacks. These authors deployed multiple models and compared the results to determine the most effective mitigation of an attack. Several independent models were based on SVM [27, 101, 138, 164, 173, 175, 180], artificial neural network (ANN) [98, 101, 176, 180], convolutional neural network (CNN) [135, 172, 175, 180], long short-term memory (LSTM) [95, 172], random forest (RF) [101, 180], decision tree (DT) [98, 177], gated recurrent unit (GRU) [180], recurrent neural network (RNN) [52], K-nearest neighbour algorithm (KNN) [180].

Ullah et al. [94] proposed five models to combat attacks in an IoT environment, including three single and two hybrid models. The independent models are CNN, LSTM, and deep neural network (DNN), while the hybrid models are LSTM-GRU and LSTM-CNN. Varadharjan et al. [91] and Gaba et al. [181] proposed ML models; however, the authors needed to indicate which model was employed.

6.1.6 Traffic duplication techniques

The capacity of the controller to have a global view of the network and automatically generate rules to reroute malicious traffic led to the exploration of traffic duplication methods as a strategy to counter security concerns. The traffic duplication technique involves replicating traffic to an extra device or environment. It may be performed in several ways, including network taps and port mirroring.

A number of the proposed systems that will be explored in this section implement traffic or port mirroring. This method includes copying or duplicating traffic to a detection module/system, which then delivers more intelligence to the controller. Traffic mirroring to a middlebox was proposed as a solution by [11, 183] and [184] for further analysis of the traffic. The works in [27] and [173] propose ML-based detection modules that leverage SVM through mirrored traffic to identify and mitigate attacks. The results in [42] and [132] proposed a combination of one or more middlebox solutions and/or monitoring solutions, such as sFlow, to mitigate attacks through mirrored traffic to these middlebox solutions, which in turn provide intelligence to the controller to create corresponding traffic rules. In a different method, [113] and [172] presented solutions to mirror incoming traffic to a network monitor and a middleware component to analyse the traffic further and offer further intelligence to the controller.

6.1.7 Statistical techniques

The works that used statistical methods to help with attack detection or pinpoint an attack's origin or source are discussed in this section. A few of these approaches have been combined with additional techniques already covered above.

- Probability distribution: Probability distribution (PD) has been used to analyse the frequency and intensity of attacks by calculating the possibility of an event occurring at random based on specific parameters. The authors of [110, 165] suggested a PD-based method for calculating the flow rule limit by employing distinct methods. While [165] used PD to assess the similarity between the various flows to differentiate between legitimate and malicious HTTP traffic, [110] applied PD to determine the flow rule limit count during a non-attack phase and deemed a flow malicious once it exceeded a predefined threshold. Yu et al. [137] proposed the use of the univariate Gaussian distribution, which is a variant of PD, to discover TCP SYN flood attacks by analysing the transmission rate of SYN packets.
- Bayesian filtering: Bayesian filtering (BF) offers an effective way to monitor network traffic, given that it provides a real-time response estimation for suspicious behaviour. The works in [9] and [148] leveraged BF (together with entropy calculation) as one of the schemes to mitigate

attacks by classifying the likelihood of illegitimacy and notifying the controller of the necessary action.

- Estimated Moving Weight Average: The estimated moving weight average (EMWA) provides a method to compute the average value for a set of data points (traffic flow) where the more recent data are given greater weight than the older data, ensuring that a dynamic threshold is applied based on overall system behaviour to detect irregular behaviour by examining the trends in traffic behaviour. The works in [116] and [117] proposed the use of EMWA in generating a dynamic threshold to determine dynamic threshold values per feature for the following traffic window for network traffic and alert the controller to create a corresponding rule to allow or deny network traffic.
- Adaptive Correlation Analysis: Adaptive correlation analysis (ACA) measures the relationship between two or more variables and analyses the parameters based on the observed data by dividing the data into segments and analysing each segment, which, in this case, ensures that traffic is thoroughly examined from source to identify patterns of activity that may indicate a security threat or other abnormal activity. Zheng et al. [185] proposed leveraging ACA to detect and mitigate attacks with DDoS using an SDN switch, where the controller instructs each switch which flow to investigate rather than analysing all flows that do not require pre-processing.
- Component Analysis: The works in [180, 186] and [187] employed the use of component analysis to identify patterns and relationships in network traffic by reducing the flow characteristics to detect anomalies or patterns that may indicate security threats. The solutions leverage CA for the preprocessing of data packets to reduce the dimension of the flow properties to ease the classification of traffic.
- Time Series: Jia et al. [138] proposed a DDoS detection method called FORT, which uses a time-series algorithm to indicate port statistics. Time-series analyses the traffic, provides a trend to forecast the future value of traffic behaviour, and provides intelligence to the controller in determining whether an anomalous port belongs to the attack source or the attack target. The controller can then initiate the detection of the switch to which the port belongs.

6.1.8 Graphic/geometric-based techniques

Porvahab and Ekbatanifard [150] and Sharma et al. [115] proposed distinct solutions based on graph theory/geometry. The technique offered by [150] facilitates evidence analysis in a cloud context by generating a logical graph of evidence collected from BC and certified by a digital signature. The work in [115] leverages the use of a flow graph builder that generates an incremental graph model based on the information obtained from a traffic parser, which is then delivered to the controller after going through other processes to allow or block traffic. The controller aids in the collection of the evidence. Islam et al. [46] and Rahman et al. [149] proposed using the Euclidean distance technique to measure the distance between nodes in a distributed IoT architecture, aiming to mitigate cyber-attacks by clustering IoT nodes with the controller as the cluster head.

6.1.9 Optimisation techniques

The works in [178, 188] and [150] offered various optimisation methodologies in their respective works to improve the security posture by utilising the SDN paradigm. The method proposed in [178] uses the brainstorming optimisation technique to optimise the extreme learning machine (ELM) weight to increase the detection capabilities of the solution. This technique simulates the behaviour of humans throughout the brainstorming process. Ghosh et al. [188] also proposed using differential evolution (DE) to improve the measurement of data collected from several substations in an intelligent grid, assess the state of the grid, and generate an alarm for the controller when an anomaly is detected. Porvahab and Ekbatanifard [150] proposed the harmony search optimisation technique to produce secret keys that serve as part of the authentication requirement in addition to a user's ID, password, and secret code in a cloud.

6.1.10 Point-to-point/many communication techniques

The works in [143] and [167] proposed a strict one-to-one mapping between a request and a response. Kim et al. [143] used this approach to prevent orphan DNS responses by looking for an existing request that matches a specific response and managing the entire network using the SDN paradigm. Similarly, [167] presented a stateful mapping (one-to-one mapping) mechanism between DNS requests and responses, with the controller telling the OF switches to monitor the request/response.

Ferreira and Saotome [190] proposed an SDN-based cybersecurity architecture for smart grids. The architecture employs the GOOSE protocol [253], a publisher/subscriber protocol used when exchanging frames between intelligent electronic devices (IEDs) on a client machine. Following packet capture, categorisation will be utilised to establish the signature patterns included in each package that will transit through the controller, resulting in signature standardisation.

Cox et al. [189] proposed utilising SDN and WebRTC to detect and block rogue access points from connecting to a network. The proposed work creates a table of MAC to IP address mappings and examines the corresponding switch port, DHCP, and ARP packets. Static MAC-IP address-port number mappings from a user-supplied file are also supported by the system. For security concerns, entries to the table can only be made via a legal DHCP offer, with the entry set to initialise.

6.2 Malware defence mechanisms

Malicious software, including Trojan horses, computer viruses, and Internet worms, substantially compromises the security of networked systems. The diversity and abundance of malware significantly dim the effectiveness of conventional signature-based detection. Malware propagation poses a severe challenge to contemporary information technology because certain malware employs difficult-to-identify obfuscation techniques, including encryption, polymorphism, and stealth behaviour, resulting in anti-malware software occasionally failing to detect or remove this malware completely. However, an analysis of malware is necessary to comprehend its behaviour and substance. Malware analysis falls under the categories of static and dynamic analysis. While the latter involves the execution or running of the malware program in an isolated environment to analyse its behaviour, the former analyses or examines the malware without actually running or executing it. To prevent the proliferation of malware, a variety of strategies have been used, including those that are behaviourally based [254, 255], DL/ML-based [256–258], heuristic-based [259–261], and signature-based [262–264]. However, some of these techniques do not work well enough to stop these attacks on their own. When a zero-day attack or other unknown attack is found, signature-based defences, which are thought to be particularly successful against known attacks, fall short. The disadvantage of the behavioural-based method is that some binaries may not execute correctly in a protected environment; as a result, some malware samples may be mistakenly labelled as benign. Despite using rules and ML approaches, the heuristic approach is still viewed as having limitations because it cannot detect sophisticated malware. The ML-based technique, which has also offered high resistance against malware attacks, has a drawback because it cannot fend off evasion attacks. Due to its overall flexibility, interoperability, programmability, and global view of the network, SDN has been used to combat malware attacks in networks. The abilities of SDN could further enhance the already proposed solutions based on ML/DL, behavioural analysis, signature-based approaches, and heuristics by providing a global view of the entire network and thereby collecting adequate statistics or flows that can help mitigate/prevent an attack. The next subsections cover the numerous malware security approaches and sub-techniques listed in Table 2, which are based on the SDN paradigm. Table 6 also offers a detailed overview of all proposed systems, including the CIA-Triad addressed by the solution, the enabling technology that connects with SDN, and the malware type against which the solution was proposed or tested. Table 8 summarises the main advantages/contributions, and Table 9 summarises the drawbacks of the approaches grouped together.

6.2.1 Network/connection observation techniques

Network and connection monitoring are important strategies for identifying and mitigating attacks by malware. Malware is frequently designed to create network connections in order to interact with command-and-control (C&C) servers, exfiltrate data, or perform other destructive actions. Malicious activity can be detected and blocked by monitoring network traffic and connections. In this section, we identify two sub-techniques: packet/payload inspection and port monitoring. Rouka et al. [17] proposed a detection and mitigation technique based on packet/payload inspection against Ransomware using ExPetr as a case study. The authors proposed a three-module approach that was implemented on a POX-based controller and consisted of port blocking, server message block (SMB) payload inspection, and HTTP payload inspection modules. Each of the modules implements a unique form of defence. The modules may be enabled alone or in combination, depending on the threat models used and the security policies in place.

In another work, [191] proposed a work based on port monitoring using SDN for mitigating the Sodinokibi ransomware and also integrated the use of IDS. A static and dynamic study of the malware's behaviour was done to analyse the malware effectively. When sending data packets, the detection system checks the IP address and port. Using the IDS's database signature, suspicious files and traffic are found.

6.2.2 Criteria-based techniques

This subsection discusses two criteria-based sub-techniques, as seen in Table 2, namely traffic redirection and address listing. The works in [18, 192–194] proposed solutions based on traffic redirection. Alotaibi and Vassilakis [18] proposed a five-module system (DPI, ARP scanning, packet header inspection, honeypot, and SMB packet modules) to defend the self-replicating BadRabbit ransomware. The modules mainly monitored the traffic on communication ports 80, 445, and 139 for HTTP and SMB protocol-related communications. Once the controller has checked the network and determined that the traffic is malicious, the study recommends the use of honeypots to reroute destination traffic on ports 445, 139, and 80 to various honeypots. Additionally, BadRabbit can be located using ARP scanning. If the packet is an ARP, the source IP address is checked and compared to 0.0.0.0.

Table 6 Specifics of works based on malware defence techniques

References/framework	Malware type	Measures	Approaches	Defence behaviour	Location of deployment	SDN testbed	Deployment environment	Enabling technology	Type of analysis	CIA Triad tackled
[16]	WannaCry	DM	N	R	C	E	Generic	NS	Static & Dynamic	CoIA
[17]	ExPetr	DM	N	R	C	E	Generic	NS	Static & Dynamic	CoIA
[18]	BadRabbit, WannaCry, NotPetya	DM	N	R	C	E	Generic	NS	Static & Dynamic	CoIA
IoTSDN-RAN [96]	CrytoWall	DM	N	R	C	Si	IoT	ML	NS	CoIA
[191]	Sodinokibi Ransomware	DMP	N	PR	CS	E	Cloud	IDS	Static & Dynamic	CoIA
[192]	Generic approach	DM	N	R	C	E	IoT	NFV, ML, honeypot	Dynamic	CoIA
WORM-HUNTER [193]	Sasser Worm	D	N	R	CS	E	Generic	Virtualization	NS	CoIA
MARS [194]	Generic approach	D	N	R	CS	E	Generic	NS	Static & Dynamic	CoIA
[196]	Ransomware	D	N	R	C	Si	Generic	ML	Dynamic	CoIA
FedDICE [197]	WannaCry, Petya, BadRabbit, PowerGhost	DM	N	R	C	NS	IoT	ML	NS	CoIA
[198]	ModPack, Matsnu, Rammit, Suppobox, Tinba, Bamital, Gameover	D	N	PR	C	E	Generic	ML	NS	CoIA
[199]	Phishing	D	N	R	C	Si	ISP	DL	NS	CoIA
[200]	CryptoWall and Locky	D	N	R	C	E	Web Protocol	NS	NS	CoIA
[201]	Rig Exploit Kit, Cerber, Crypt XXX, Qakbot malware	DP	N	R	C	NS	Generic	NS	Static & Dynamic	CoIA

Measures: *D* detect, *M* mitigate, *P* prevent | Approach: *N* network, *D** device, *A* application | Defence behaviour: *R* reactive, *P* proactive | CIA triad: *A* availability, *I* integrity, *Co* confidentiality | Testbed: *Si* simulation, *E* experiment, *NS* not specified | Location of deployment: *C* controller, *S* server

A closed-loop defence system based on SDN was proposed by [193] using a carefully designed dynamic flow table. The proposed approach has the ability to build several types of honeynet systems with different network structures, protecting the network against worms. The method offered a way to conceal the traffic that was being redirected to a honeypot from attackers. In order to prevent an attack from spreading to other honeynet systems, the flow table manages all traffic and ensures that traffic from victim honeypots is isolated in the honeynet system to which they belong. In another work by [194], the proposed method integrates the network layer into the malware analysis process. Three fundamental components make up the architecture: the sandbox, which controls the execution of binary samples; the controller, which oversees and directs the environment used for malware analysis; and the resource pool, which is a collection of hardware and online services that can be used to build or modify an analytical environment. In addition to the aforementioned components, the architecture makes use of software modules that are constructed on top of the controller. Such modules successfully communicate with the malware analysis process by controlling the system settings.

Zolotukhin and Hamalainen [192] proposed an NFV- and SDN-based defensive system for IoT networks. The fundamental element of the defence system is a reinforced ML agent that assesses the threat of a possible attack and decides on the best course of action to reduce it. The ML algorithm recognises unusual behavioural patterns, assesses the threats of prospective intrusions, and neutralises the threat by sending more flows to the controller. These flows include modifying link capabilities, banning malicious connections, or rerouting some network traffic to security middleboxes.

The works in [196] and [195] proposed solutions based on address listing. Chang et al. [196] proposed a weighted K-nearest-neighbour (KNN)-based technique for ransomware prediction. The proposed system includes an SDN-integrated dynamic isolation mechanism for the detection and prediction of packet traffic containing ransomware. The study developed two tiers of security against suspected IP traffic based on SDN architecture. At the first layer, blacklist isolation is used to compare IPs on the Ransomware tracker, which is an implemented module. All programmes and websites that use network behaviour tracking and content tracing as a form of extortion are tracked by a non-profit group called Ransomware tracker. Every five minutes, the website updates its blacklist of malicious domain names and IP addresses. Also, [195] proposed a solution based on a number of mitigation strategies such as SMB scan detection, ARP scan detection, DNS blacklisting, IP and MAC address blacklisting and traffic re-routing. The strategy based on IP and MAC address blacklisting works when the MAC address and the physical location of the device are unknown, and a blocking rule for the specified IP address is installed on every switch. For DNS backlisting, the controller checks whether the packet is a DNS through the header fields and compares it with a backlist to decide whether to drop/allow the traffic.

6.2.3 ML/DL-based techniques

The solutions proposed by [96, 196], and [192] are based on distinct ML models at various levels of detection/mitigation. The work proposed by [96] detected ransomware using Naive Bayes and PCA at various phases. Chang et al. [196] deployed the KNN algorithm in the form of nearest neighbours for malicious traffic dynamic isolation to detect and predict traffic containing ransomware, and [192] proposed a reinforced ML agent that assesses the threats of a potential attack and decides on the best course of action to mitigate them.

In another work, [197] created the FedDICE architecture, which combines SDN with ML to reduce the frequency of attacks on a hospital's infrastructure. A federated distributed architecture that integrates federated learning among hospitals that are spread out geographically and exchange information to guarantee learning that respects individual privacy. Four well-known ransomware attacks—WannaCry, Petya, BadRabbit, and PowerGhost—were used to test the proposed method's accuracy. Ahmed et al. [198], in another work, proposed the use of SDN and ML to create an accurate, affordable, and scalable system for detecting compromised hosts connecting with external C&C servers. The method depends more on the behavioural traffic profile than the packet content and dynamically chooses network flows for real-time diagnosis by trained models. An SDN-based monitoring system was created to automatically mirror TCP/UDP flows relevant to domain generation algorithm queries so that the trained models may diagnose the data using various ML models.

Wazirali et al. [199] proposed a solution to detect phishing URLs using SDN, clustering and feature selection techniques, and the CNN algorithm. Recursive Feature Elimination (RFE) combined with the SVM algorithm form the basis of the feature selection technique. The control layer receives the URL phishing detection process from the user's hardware, trains it continuously on new data, and then sends the results to the switches.

6.2.4 Packet analysis/inspection techniques

The practice of analysing network traffic at a granular level to discover and diagnose network faults, fix security concerns, and optimise network performance is known as packet analysis. It entails intercepting network packets as they pass across a network and analysing the contents of those packets to obtain meaningful network information. DNS inspection is a type of packet inspection. The work proposed by Akbanov et al. [16] was based on inspecting DNS packets/traffic. The method enables the observation of DNS requests made by the WannaCry worm component across internal and external networks using port 445 of the SMBv1 protocol during the infection and replication phases. The inspection of DNS traffic using dynamic blacklisting, which scans explicitly the network traffic for the presence of malicious domain names or IP addresses used during WannaCry's contact with the C&C server, is the basis of the proposed work.

6.2.5 Payload extraction/distribution techniques

The payload is extracted in order to be able to retrieve actual executable code or data from a malware sample for examination. The payload is the portion of the malware that executes the harmful purpose or activity on the infected system. Cabaj et al. [200] proposed an SDN-based detection method for extracting payloads. On observing two crypto ransomware families' network communication, namely CryptoWall and Locky, it was determined that an analysis of the HTTP message sequences and their respective content sizes is sufficient to detect such threats. The preprocessing is done on incoming TCP segments that include HTTP traffic and reassembles outbound messages. The size of the data supplied to the server and the host IP or domain address are taken out of the HTTP header when the entire request is put back together. The destination HTTP server, which can be a ransomware C&C or a proxy server, is then defined using the retrieved host IP or domain name.

In another work, [201] proposed a system to detect and stop the distribution of Exploit Kit (EK). A number of payloads can be attached to a single new thread using the transport distribution system capability, which is available to exploit kit operators. In actuality, it is an essential part of payload distribution, and with the apparent rise in ransomware attacks utilising EKs, it is necessary to have a detection mechanism against this type of malware. The run-time malware detection system uses ransomware communication patterns, significant network indications of EKs, and ransomware family signatures when it is implemented in a network. The intention is to hide the malicious nature of these payloads from users because the filenames closely resemble those of system executables; therefore, it mixes these components with content from the profile.

6.3 Deception/masquerading mechanism

According to Wang and Lu [265], cyber deception is the deliberate use of misleading techniques to entice attackers to take (or stop from taking) specific actions that aid computer security defences. Wang and Lu [265] also provided a conceptual framework for cyber deception, outlining a two-step process in which the defence first acquires as much intelligence as possible about the adversary before developing a real deception plan based on what is discovered about the adversary. Cyber deception was divided into two categories by Lu et al. [266]: information simulation and dissimulation. The latter involves hiding information and includes techniques like masking, repackaging, and dazzling. In the former, decoying, invention, and mimicry tactics are employed to generate and deploy false information to confuse and divert attackers. The proposed works in this section critically examine the use of SDN in combating attacks by masking, mimicking, and fooling the attacker. In the next paragraph, various works based on deception or masquerading are divided into different techniques, as shown in Table 2. Table 7 provides the specifics of each of the works, while Tables 8 and 9 provide more details about the solutions and their main advantages and limitations, respectively.

6.3.1 MTD-based techniques

The works in [28, 202–211, 224] proposed various solutions based on moving target defence (MTD). MTD is a proactive cybersecurity technique that aims to enhance the complexity and unpredictability of a network environment in order to make it more difficult for attackers to detect and exploit vulnerabilities. The main idea underlying MTD is to make the attack surface more dynamic and unpredictable, making it more difficult for attackers to start an attack effectively. Zhao et al. [204] proposed the finger hopping (FPH) approach, which is based on the SDN paradigm, to combat fingerprinting attacks. FPH exerts the principle of MTD to deceive attackers who are fingerprinting an environment. The fingerprinting activity is detected by the IDS's detection module using a signature database that can be created by gathering probe signatures from tools like Nmap. If it is found that a communication is fingerprinted, its outward transmission will be rerouted and changed to hop its fingerprints. A flexible network configuration is necessary to achieve traffic rerouting without disrupting communication. In an improved approach by [209], the solution factors in the limitation of the signature database used in [204] by passing the traffic through a second system for confirmation.

The work proposed by [209] is based on obfuscating the attack surface through host mutation, port obfuscation, and decoy server obfuscation. Hyder and Ismail [203] proposed an architecture that employed the idea of shadow servers to counter reconnaissance attacks directed at servers operating in an SDN environment, which constitute the first stage of cyberattacks. Once probing is detected, the traffic is sent to shadow servers, which are selected based on round robin. The selected shadow server then responds to the attacker's probing traffic. The shadow web server's IP address will be changed to match the original web server's IP address as a response to the probing traffic. In a similar work to [203, 206]

Table 7 Specifics of works based on deception/masquerading defence techniques

References/framework	Attack	Measure	Approach	Defence behaviour	Location of deployment	SDN testbed	Deployment environment	Enabling technology	CIA triad
SDHHive [195]	Ransomware	DP	N	P	C	Si	Generic	Honeypot	CoIA
[219]	Generic	D	N	R	C	Si	Generic	Honeypot	A
[227]	Reconnaissance, data exfiltration attacks	DP	N	P	CS	NS	IoT	Honeypot	A
[226]	Generic	DP	N	P	C	Si	Generic	Honeypot	A
[228]	DDoS	DP	N	P	C	Si	IoT	Honeypot	A
HaT [213]	Reconnaissance	DP	N	P	C	Si	Generic	NS	A
[214]	Generic	DP	N	P	C	NS	IoT	Honeypot	A
[267]	Fingerprinting, scanning	DP	N	P	C	E	Generic	Honeypot	CoIA
[218]	Reconnaissance	DP	N	P	C	Si	Generic	Honeypot	A
[221]	Generic	DM	N	R	C	Si	UAV	Honeypot	A
[215]	Generic	DM	N	P	C	Si	IoT	Honeypot	A
[208]	DDoS	DPM	N	PR	C	Si	IoT	MTD	A
MTDCD [224]	Generic	DP	N	P	CS	Si	Generic	Honeypot	A
[205, 206]	Scanning	DP	N	P	CS	Si	Generic	Middlebox (IPS)	A
ACyDS [222, 223]	Reconnaissance	DP	N	P	C	NS	Generic	honeypot	A
[204]	Fingerprinting	DP	N	P	CS	Si	Generic	Middlebox (IDS)	A
CHAOS [209]	MITM, port scanning	DP	N	P	CS	E	Generic	NS	A
[207]	DoS	DPM	N	PR	C	Si	Smart Grid	NS	A
[229]	Generic	DP	N	P	C	Si	Generic	honeypot	A
BOTTLENET [220]	DDoS	DP	N	P	C	Si	Generic	NS	A
[211]	DDoS	DP	N	P	A	NS	VANET	BC	A
[212]	Generic	DP	N	P	C	Si	IoT	Honeypot	A

Measure: *D* detect, *M* mitigate, *P* prevent | Approach: *N* network, *D** device, *A* application | Defence behaviour: *R* reactive, *P* proactive | CIA Triad: *A* availability, *I* integrity, *Co* confidentiality | Testbed: *Si* simulation, *E* experiment, *NS* not specified | Location of deployment: *C* controller, *S* server

Table 8 A summary of the main contributions and advantages of the literatures discussed in Sects. 6.1, 6.2, and 6.3

Advantages/main contributions	References
Dynamic threshold to determine an attack	[4, 27, 99, 116, 117, 165]
System performance is effectively managed during attack defence to avoid system overhead or performance issues	[5, 92, 95, 108, 111, 113, 131, 137, 144, 163, 163, 164, 164, 166, 166, 174, 178, 184–188]
Real-time detection of attacks	[8, 42, 92, 118, 120, 140, 171, 177]
A single model, a hybrid ML model for traffic classification for SDN decision making, improves accuracy and performance	[13, 93, 94, 98, 104, 170]
Clustering/HA of the controller to distribute load and avoid SPoF	[44, 46, 97, 102, 106, 129, 130, 147, 149, 152, 153, 160, 188, 197]
The system works both proactively and reactively	[11, 14, 126, 167, 177, 191, 207, 208]
Hashing of network traffic/details to prevent attacks like MiTM and Eavesdropping	[90, 91]
Deployment of solutions at the fog/edge layer to shield cloud network	[92, 97, 176]
Ability of the system to detect the source of an attack	[96, 103, 110, 128, 137, 165]
Offers a two-layer or multi-layer security approach or authentication	[88, 90, 107, 116, 151, 155, 156, 161, 171, 196, 200, 205, 206, 213]
Ability of the system to combat both internal and external attacks	[148, 161, 183]
Offers a real-time/online training of ML model	[101, 173]
Offers a multi-attack detection mechanism/framework	[17, 18, 100, 119, 175]
Implements a queuing or buffer system to handle traffic flow	[115, 122, 142]
Offers static and dynamic analysis of malware	[16, 17, 191]
Capable of hiding system-related details from attackers, which may mislead the attacker	[192, 193, 213, 214, 220, 222–224, 227–229]

Table 9 A summary of the disadvantages and limitations of the literatures discussed in Sects. 6.1, 6.2, and 6.3

Disadvantages/Limitation	References
Static threshold to determine an attack	[107, 109–112, 131, 135, 137, 148–150, 164, 166, 213]
The solution is limited to a specific attack type	[5, 118, 120]
Possibility of bandwidth/system overhead for sFlow or other protocols deployed	[8, 131, 132, 141, 146]
Possibility of overhead/performance issues is a result of traffic mirroring	[11, 27, 108, 113, 123, 173, 178, 183, 198, 219]
The solution only detects an attack. Another solution is required to mitigate an ongoing attack	[13, 95, 96, 110, 124, 151]
Possibility that the controller could experience overload or SPoF	[46, 93–96, 98, 102, 103, 105, 117, 126, 143, 144, 154, 158, 165, 176, 177, 185, 186, 194, 214]
In the case of IoT, the solution exposes the cloud network to attack	[94]
The solution is only able to detect an attack at one OSI layer	[109, 129, 134, 158]
Prone to human error because of manual configuration required	[99, 106, 106, 115, 125]
The solution is unable to detect unknown attacks	[160, 201]
The solution does not propose a malware analysis, be it static or dynamic analysis	[193]
The solution could be exposed if the service/file name deployed during malware propagation is changed or updated by an attacker	[18]
No experiment or PoC to validate the proposed work	[147, 211]
The approach is not cost-effective or time-consuming in implementing	[142, 169, 175]
The solution does not offer online training for the dataset	[90, 113, 155]

proposed an MTD-based technique that implements address randomisation for end hosts while performing transparent address changes of packets for the IPS, while the operation of the IPS continues to monitor the devices' actual IP addresses. The controller modifies the packet headers as they pass via switches using virtual IP (vIP) addresses, and the IPS observes based on real IP (rIP), which is constant. Chiba et al. [205] proposed a further improvement to this approach by randomising host mutations using the rIPs and vIPs.

Galadima et al. [208] and [211], in a similar approach to [203, 205, 206], proposed solutions based on an MTD approach that involves a continuous modification of the attack surface through the use of a reactive and proactive MTD shuffling mechanism that proactively shuffles the network addresses of IoT devices and edge computing servers. The proposed approach established network obfuscation using a combination of both time- and event-based movements. The proposed system implements two stages of mechanisms: the proactive stage, which entails masking the IP during map generation, and the reactive stage, which lessens the impact of the attack. A host may have many randomised virtualised addresses since the network address shuffle is based on multiplexing.

The proposed solution by [210] ensures that large-scale attacks from the ISP side are mitigated before reaching the customer's network by applying constant system profile changes and randomisation. In a similar work, [202] proposed "Whack-a-Mole", an SDN-driven cloud resource management strategy through network obfuscation that can aid cloud service providers. "Whack-a-Mole" operates on two levels: first, it uses a revolutionary virtual machine spawning methodology to spawn multiple copies of crucial services onto fresh instances of cloud resources, and second, it uses address space randomisation to assign IP addresses to the replicas.

The approach put forth by [28] is similar to that of [210] in that it uses NFV and MTD to avoid attacks in an ISP environment. To trick attackers, the solution uses MTD to alter traffic pathways on the fly. Similarly, [28, 207] proposed MTD in a smart grid communication environment for adjusting the direction of data flow by changing the path. When an attack is detected, the path is modified immediately; if the attack is not detected, the path is continually altered by the controller.

6.3.2 Honeypot/honeynet/decoy node-based techniques

The works in [195, 213–219] proposed solutions based on honeypot/honeynet or decoy nodes deployment. Security techniques and technologies, including honeypots, honeynets, and decoy nodes, are deployed to detect, deflect, or stop attempts at unauthorised usage of a system. A honeypot system, often referred to as a decoy system, is a network-attached device that may entice or trick attackers. It may seem to be a fundamental element of the surrounding environment. A network's honeypots can be combined into honeynets. Li et al. [219] proposed a honeynet built on SDN. At the start and end of the operation of the entire system, the attack migration system screens all traffic and sends a copy to the honey network for further analysis. Karate et al. [195] proposed a solution that deployed an IDS that can recognise SMB and ARP scans and a honeypot that supports API integration. The controller analyses suspicious traffic packets and routes them to the honeypot for additional analysis.

Xing et al. [213] proposed a deceptive method for preventing network reconnaissance using SDN. The idea behind this strategy is to hide active, valuable hosts in order to persuade adversaries that these hosts do not exist or are frequently unavailable and to trap adversaries in decoy nodes in order to give the impression that valuable hosts have been located while learning their attack vectors. Lin [214] proposed an in-network system that intelligently spoofs network traffic by introducing false information into regular traffic in order to direct it to suspicious nodes. Traffic is routed to the controller, which effectively isolates the suspicious nodes from other communication nodes. The controller impersonates imaginary nodes, or "phantom nodes", in network communications that are used to interact with suspicious nodes.

Anjum et al. [218] proposed HoneyRoles, a solution that employs honeypot connections to create haystack-like networks around client hosts that play important organisational roles and make an adversary residing in one or more compromised packet forwarding devices uncertain of the identities of crucial client hosts. Tan et al. [215], in a similar work, proposed a topology with the idea of protecting essential drones on a UAV network using the concept of honeypot drones, fooling attackers into thinking that the honeypot drone is a key drone. In this setup, drones can be deployed as relays or servers.

The approach proposed by Kyung et al. [216] leveraged the use of honeypots and SDN to build an environment to prevent the internal propagation of malware. The solution assumed the shape of a reverse proxy to take better control over inbound/outbound traffic while obtaining network settings from the controller. Bernieri et al. [217] proposed an architecture—MimePot that offers a model-based approach in contrast to traditional honeypots; it replicates physical processes to deceive attackers targeting an industrial plant.

6.3.3 Virtual topology-based techniques

The solutions proposed by [220–225] are based on deploying virtual topologies to confuse an attacker. The solutions offered by [223] and [224], also known as ACyDS, are to give each server in a corporate network its own constantly changing virtual "network view". Each host has a unique view of the "virtual network," which is made up of enterprise servers, peer hosts, a virtual network architecture, and a mix of actual hosts and honeypots. ACyDS provides virtual network views, and the proposed works provide two-way IP address translation that is undetectable to connected endpoints. Gao et al. [224] and Achleitner et al. [225] proposed an MTD-enhanced cyber deception defensive system based on SDN. The system leverages a virtual network topology to confuse the target network and information obtained by adversaries about the system. The solution by [224] leverages the IP randomisation technique, while [225] leverages a honeypot server.

The solution proposed by Shimanaka et al. [221] directs malicious traffic from the compromised operational network (O-Net) to an identically configured deception network (D-Net). The proposed work leverages SDN to implement the transfer from the O-Net to the D-Net. Two OF switches link each O-Net subnet to its associated D-Net subnet. Each endpoint on the D-Net has the same IP address as the equivalent endpoint on the O-Net in order to prevent alerting the adversary that the attack has been moved from the O-Net to the D-Net. Kim et al. [220] proposed a framework known as BOTTLENET that is capable of hiding network bottlenecks using SDN-based topology deception. Topology deception focuses on obstructing the identification of bottlenecks by giving adversaries false trace responses as they conduct topological probing of the target networks. A virtual network generator creates virtual networks (based on virtual topology) and virtual switches and links with concrete network configurations, then deploys them to deployment hosts.

6.3.4 MTD and honeypot combined techniques

These authors [226–228] proposed solutions that leverage both MTD and honeypot deployment to combat attacks. Bellalis et al. [226] proposed a virtual network architecture that leverages the controller to dynamically create and manage flow rules to direct and control network traffic. The solution includes a packet handler for handling network packets and simulating virtual network resources, a virtual network generator for describing the virtual network components and their connectivity; and a honeypot server for monitoring. Luo et al. [228] proposed an SDN-based honeypot to mimic IoT devices in order to lure attackers and malware. MTD helps to frequently change the IP of the IoT devices or servers, removing predictability and making the discovery of hosts in the environment harder, while the SDN-based honeypots imitate the IoT devices to capture and monitor the activities of the attackers. Ge et al. [227] proposed a system that combines MTD with cyber deception (i.e., a decoy system) as a defence method to achieve intrusion prevention (i.e., network topology shuffling). The proposed technique's effectiveness and efficiency are based on an analytically graphical security model in an SDN-based IoT network. The defence model includes a decoy system as a deception with the purpose of luring attackers into the system and analysing the attack behaviour.

6.3.5 Other techniques

Islam and Al-Shaer [229] proposed an active cyber deception framework (ADF) that provides some rich APIs for developing cyber deception applications. The ADF provides deception as a service through a centralised controller that enables thorough diagnosis of observations and prompts deceptive action response.

7 Comparative analysis

SDN deployment, whether as a stand-alone solution or in tandem with other technologies, provides a high degree of flexibility and allows for the development of a wide range of security strategies. As such, a comparative analysis was conducted to answer the research questions defined in Sect. 5, which is discussed below.

- The existing solutions reviewed have a number of advantages but also have a number of drawbacks, as shown in Tables 8 and 9, respectively. However, an analysis of these studies reveals that SaaDM is capable of combating cyberattacks because the solution addresses at least one of the three CIA triad components. Tables 5, 6, and 7 identify and list the CIA triad components for each solution, which provides the answer to RQ1.
- Attack Measures: The attack measure assesses the various works uses of security measures. The evaluation is based on data that demonstrate that, for blocking/filtering mechanisms, 59% of the work proposed both detection and mitigation techniques; 23% were based on detection and prevention; 17% were based on detection alone; and 1% were based on all three measures. For malware defensive mechanisms, 43% of works falling under this category were based on detection and mitigation measures; another 43% were based on detection only, 7% were based on detection and prevention, and 7% were based on all three. When it comes to deception mechanisms, 75% of the work is based on detection and prevention, 10% on detection and mitigation, 5% on detection alone, and 10% on all three. Figure 8a depicts the combined distribution of the three categories of defence mechanisms. Seventy-three works account for 50% of detection and mitigation, 42 works account for 30% of detection and prevention, 25 works account for 18% of detection, and four works account for 3% of detection, mitigation, and prevention. The analysis above provides answers to RQ3, showing that

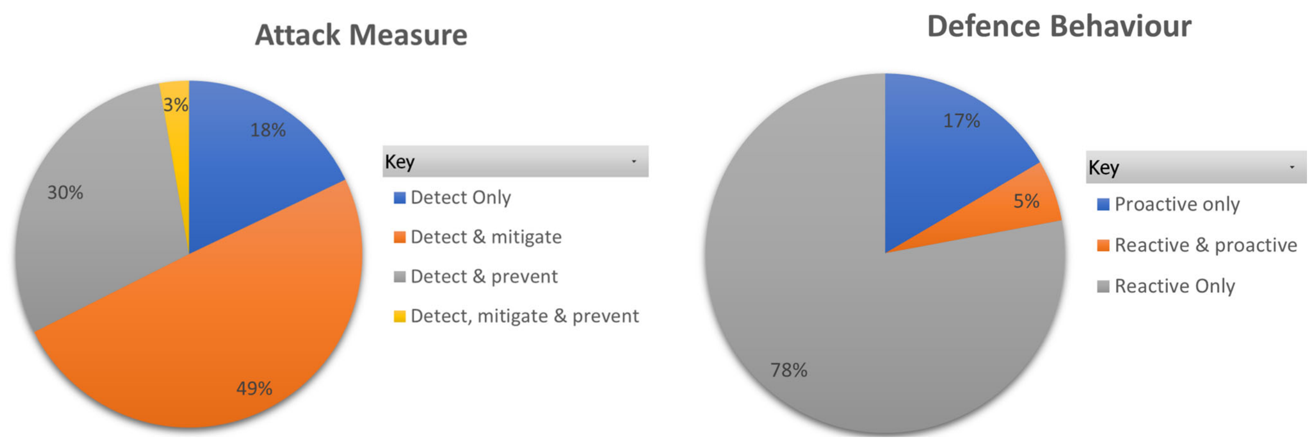

Fig. 8 **a** Attack measures type distribution. **b** Defence measures distribution

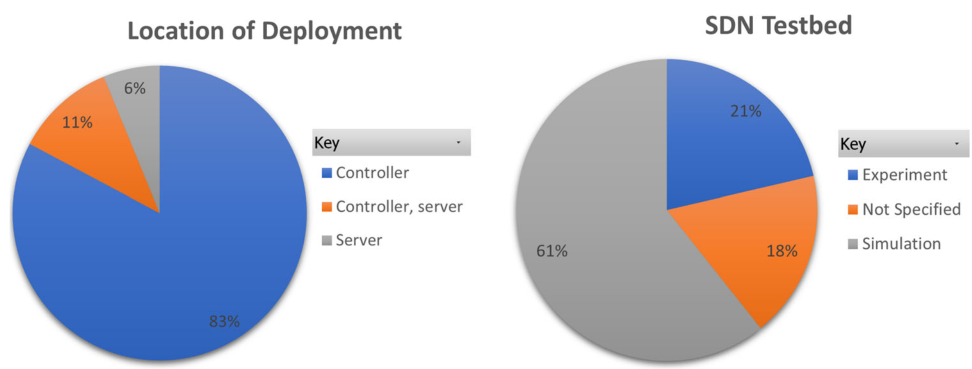

Fig. 9 **a** The distribution showing the location of deployment. **b** SDN Testbed distribution

53% and 43%, respectively, provide detection and mitigation measures against attacks for blocking/filtering and malware defensive mechanisms, while 75% of deception mechanism measures offer both detection and prevention measures.

- Defence Behaviour: One important finding from all the literature reviewed is that, according to the categories mentioned above, blocking/filtering mechanisms based on defence behaviour are typically reactive, whereas proposed works based on malware and deception mechanisms are typically proactive. Figure 8b shows that 78% of the work is reactive, while 17% is proactive. The remaining 5% are both proactive and reactive. The analysis provides answers to RQ4, which shows that most defence solutions are reactive, accounting for the majority.
- Location of Deployment: For an answer to RQ8, we examine the preferred deployment location of the defence solutions. Figure 9a indicates that the majority (83%) of the discussed solutions deployed the defence mechanism directly on the controller. The remaining 11% of the solutions were both implemented on a server and had additional defensive modules placed on the controller, compared to 6% of the solutions that were delivered on a server or loaded as an application.
- Deployment Environment: To answer RQ6, this survey also considered the deployment environment because not all security solutions are suited to all environments. The study demonstrates that the majority of solutions put forth were generic in nature, indicating that the solution is not limited to any particular environment. Solutions that were primarily deployed to address security issues in an IoT setting were then shown to be close behind. Figure 10 depicts how these solutions are distributed as per the deployment environment.
- Type of SDN Testbeds: To answer RQ7, we analyse the SDN testbeds and environments used, 30% of the solutions were based on real-world experimentation, while 61% relied on simulations, particularly the Mininet application, and 18% gave no information about the SDN testbed that was employed. The distribution of these works is shown in Fig. 9b.
- Defence Approach: According to the analysis, as seen in Fig. 11, 85% of the defence mechanisms are network-based solutions, while 9% are combined device- and network-based solutions. The analysis also shows that 2% of the defence mechanisms are deployed to combat attacks against the network and application. Three per cent of the proposed solutions are device-based. The results

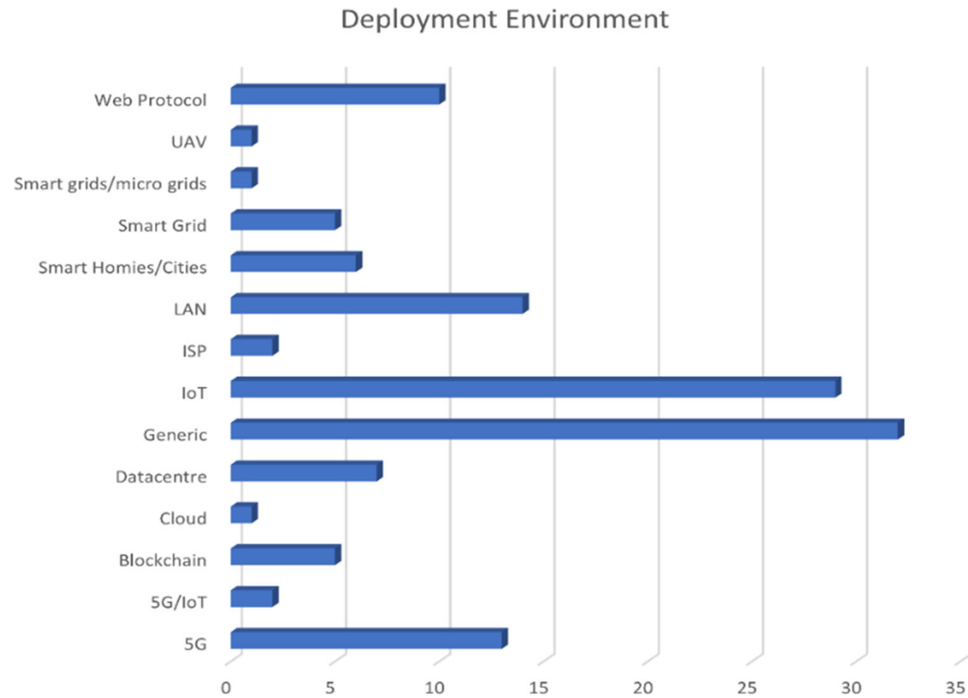

Fig. 10 The distribution of deployment environment

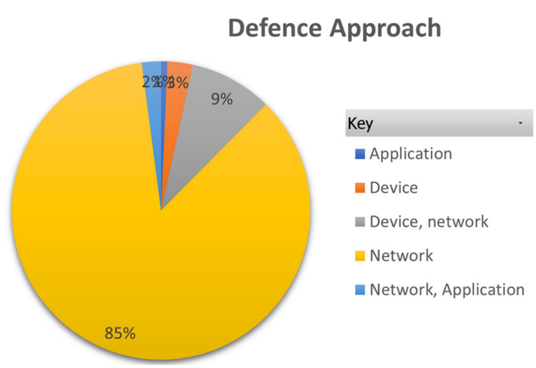

Fig. 11 Defence approach distribution

demonstrate that SDN as a defence mechanism is primarily employed at the network level, highlighting SDN's capacity to have a global view of the network because the majority of entry points for an attack are often through the network. This analysis provides the answer to RQ5, which identified that most of the defence solutions are network-based.

- Enabling Technology: A variety of technologies were paired with SDN to provide a defence mechanism, as seen in Fig. 12. Most of these works (32%) adopted an SDN as the sole defence measure, highlighting SDN's capabilities to combat cyberattacks. Other technologies such as ML, middle box, honeypot, NFV, BC, and DL accounted for 20, 9, 9, 8, 7, and 6% of the total proposed solutions combined with SDN, respectively. In contrast, the use of other technologies such as sFlow, neural networks, and virtualization accounted for around 1–2% of each of the total proposed solutions.
- Attack/Malware Types: To answer RQ2, we analysed the solutions reviewed, and our study revealed 197 attack/malware types. Most of the solutions were generic, which means they could combat any attack with little or no setup changes. The generic solutions accounted for 22% of the total. DDoS protection solutions received the highest proportion, which stands at 30%. DoS and port scan accounted for 5 and 3%, respectively. In contrast, IP spoofing, phishing, reconnaissance, MiTM, ARP spoofing, brute force, DNS amplification, and WannaCry accounted for 2% each, with the remainder of the attack/malware types accounting for 1% each. Figure 13 depicts a portion of the distribution, while Tables 5, 6, and 7 list the attack/malware types.

8 Challenges and future direction

There have been several challenges with the solutions implementation and some major problems that need additional study to make them more effective. The following are a few of the issues the survey identified:

- Controller Overload: Avoid overloading the controller with traffic at all costs. Most of the works evaluated proposed using just one controller, which, in this case, can

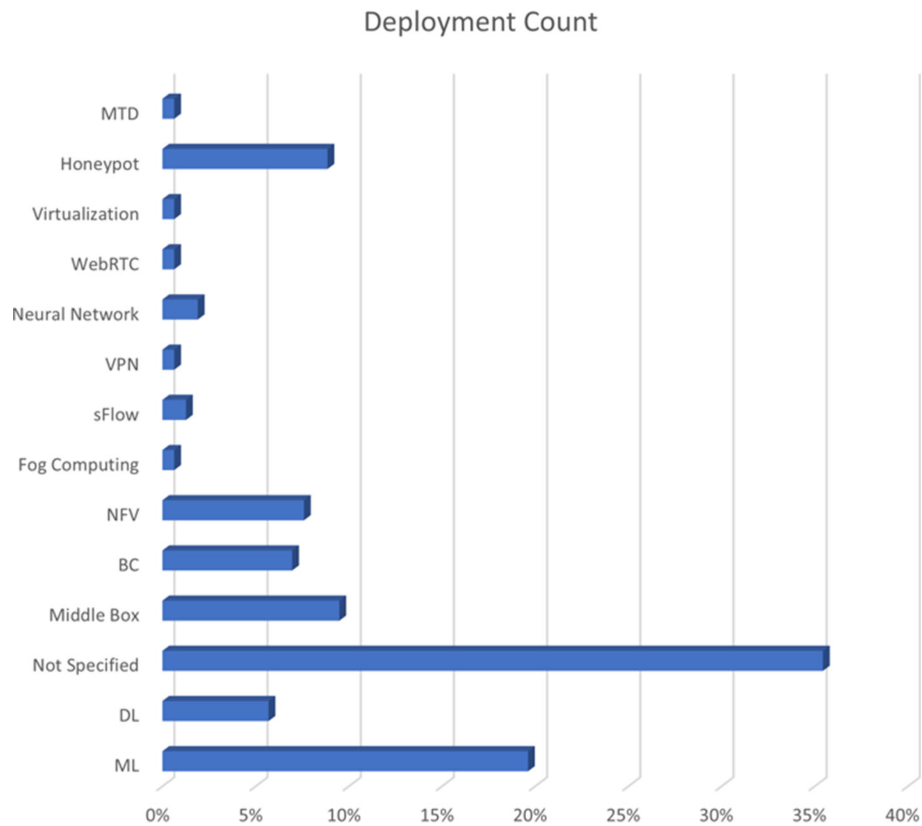

Fig. 12 The chart showing the distribution of all enabling technology

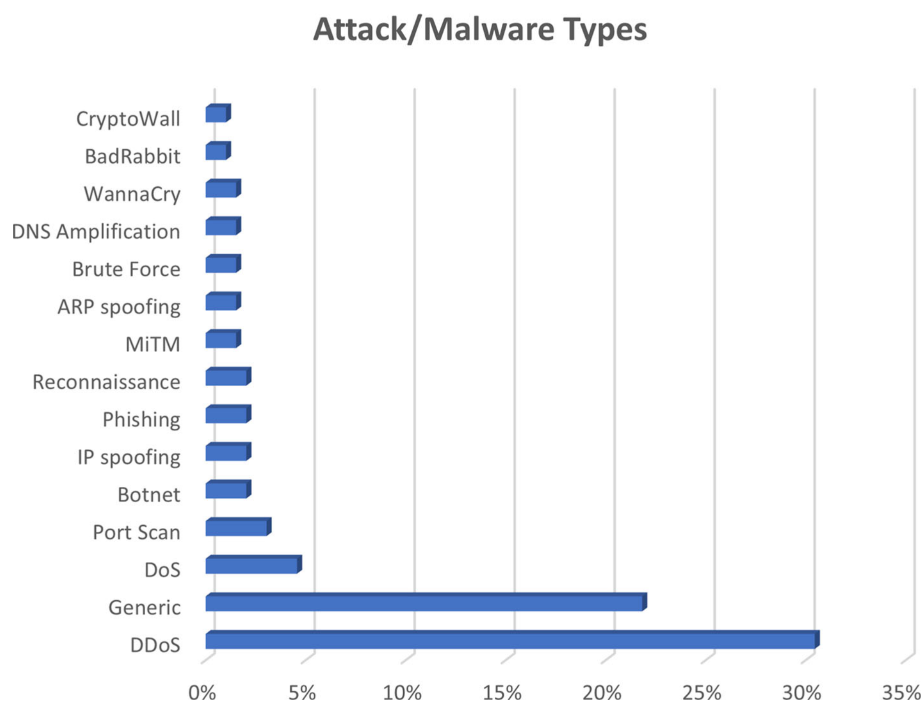

Fig. 13 The chart showing a portion of the distribution of the attack/malware types

result in performance issues in an enterprise or large setting. Given that flows can be proactive (the controller inserts rules into the switches in real-time) or reactive (the switches ask the controller for instructions when a new flow arrives), it is crucial to carefully consider the best controls for a given environment to save controller resources. Additionally, HA should be considered as a standard for deployment in an SDN environment to distribute the load in a more resilient architecture effectively. In this situation, a cluster or an architecture that enables high availability would be appropriate. Another benefit of utilising this method to handle issues affecting single points of failure is that it may assure the network's overall availability.

- Hiding the Existence of Deception/Decoy Systems: Many proactive measures were centred on deploying honeypots or MTD. However, most of the works surveyed did not clarify or demonstrate how this environment is shielded from attackers. Hiding this information would be seen as a vital precaution to ensure the credibility of such systems because there are possibilities these days that attackers can discover the existence of honeypot software, which might cause a total adjustment of the attack approach [268–270].
- Dataset for ML/DL: There is a need for a repository where up-to-date datasets for ML/DL can be obtained and used for training purposes to ensure the overall accuracy of such models, as it was observed that the majority of the datasets used for training different ML/DL models were based on old data collected. As attackers tend to improve daily in their approach, this makes it difficult to evaluate the credibility of such models in detecting recent attack types, as different attacks will exhibit different network behaviours. Additionally, there were no publicly accessible datasets for SDN-NFV setup to train ML/DL to combat cyberattacks when this study was put together.
- Performance Issue with Middleboxes: Middleboxes have grown in importance, but they can be difficult to manage and occasionally interfere with network traffic flows at different layers. This interference can have a negative impact on end-to-end performance and real-time detection and prevention of attacks. Another limitation of having middleboxes is that most of these devices are based on predefined patterns or signatures that compare observed traffic against a signature database, which means that they may be unable to detect unknown attacks. As a result, there is a need to address or improve the capabilities of middleboxes with SDN in combating attacks.

This study focused on several SaaDM-leveraged approaches. Based on this, we offer the following possible future possibilities, concentrating on how SaaDM or SDN generally may be used to serve future technologies. The emergence of 6G technology can benefit from SDN by handling some of the core KPIs of 6G [271], such as ensuring trustworthiness and end-to-end assurance in conjunction with BC, similar to the SDN-5G integration [90, 152] by enabling a secure network slicing utilising directed acyclic graph-BC (which addresses perceived weaknesses in traditional BC) and context-aware authentication handover [272]. SDN can give 6G better sensing for AI/ML applications decision-aware reconfiguration of network resources since the SDN paradigm ensures a global network view, thereby enabling improved monitoring and management of security concerns. SDN further enables the development of security policies that may be applied broadly throughout the network. This may lessen the likelihood of data breaches and unauthorised access. Furthermore, 6G networks can have a secure framework thanks to portable, secure SDN frameworks. For 6G networks, the lightweight secure SDN framework initiative suggests a compact secure SDN architecture that would allow network management to alter the network's behaviour in response to security concerns [273, 274]. Secure Networking with Software-Defined Reconfigurable Intelligent Surfaces, incorporating Intelligent Reflective Surfaces (IRSs) into an SDN architecture, may also be utilised to address security issues in 6G networks. Secure and resource-conserving wireless communication can be made possible by integrating IRSs into an SDN architecture, which can provide communication environment intelligence and programmability [275].

Converging Information Technology (IT) and Operational Technology (OT) are required to connect corporate and industrial information flows to integrate Industrial IoT and Industry 4.0 intelligence. As a result, OT networks are frequently linked to IT networks, which leaves them open to attacks [276, 277]. An OT network security compromise can result in physical damage, production interruptions, and monetary losses. Segmenting essential systems from unrelated network components helps reduce these risks. It is a network security method that lets businesses set up secure areas within their networks to guard critical information and software. With micro-segmentation, the network is divided into smaller chunks, isolated, and subject to stringent device communication rules. This strategy limits the harm that may be caused in the case of a breach by impeding attacker's capacity to move laterally within the network. By offering flexible and agile network control and automation, SDN may enhance security in micro-segmentation by allowing administrators to quickly manage security policies and deny access based on a granular level. Lowering the network's attack surface and enhancing network traffic visibility this can make it simpler to identify and address security problems. SDN may also route traffic through an inspection point, giving security experts the ability to design micro-segmentation and separate application workloads from one another, enhancing security in both data centre and cloud deployments.

The SDN paradigm and intent-based networking (IBN) can function together. IBN increases operational efficiency while reducing mistakes and risks. Although both technologies have been the subject of numerous studies [278, 279], there is still work to be done in this area. IBN can benefit from SDN by offering a simple programmable architecture for IBN controllers to manage. A flexible and dynamic network environment that can adapt to changing network requirements and enable IBN controllers to execute network policies rapidly and easily may also be provided by the deployment of SDN. SDN may offer real-time analysis of network data and useful insights into network behaviour and performance in terms of security, which can be utilised by the IBN controller to make crucial decisions. In addition to our earlier discussion of the integration of SDN into the 6G architecture, another area that merits more research is the combination of SDN, IBN, and 6G [278, 280], which could result in a highly programmable and intelligent network that can easily and quickly adapt to changing network requirements.

The future of data processing and analytics is expected to occur at the network's edge, exploiting the decentralisation trend enabled by emerging IoT and edge computing capabilities [281, 282]. This is referred to as the next-generation Internet of Things (NGIoT). One of the NGIoT roadmaps is future-proof security and privacy, which entails developing IoT to ensure privacy, increase traceability, and trust beyond regulatory compliance. This can be achieved with SDN by ensuring transparent monitoring of IoT devices and networks and allowing real-time visibility into network activity. A way where SDN may be used to maintain privacy in IoT is through the use of BC-based technology that can be enhanced. As a result, SDN-BC provides a decentralised platform for data exchange, in which data ownership is protected and access to data is rigorously restricted. As part of the NGIoT roadmap, SDN may be set up to offer a multi-purpose general infrastructure that supports machine–human interaction by providing interfaces to link with business applications or orchestration environments via the NBI.

9 Conclusion

This survey established a taxonomy that divided SDN as a security mechanism into three major categories: (1) blocking and filtering, (2) malware defence, and (3) deception/masquerading mechanisms, as well as other classifications such as defensive measures, defence/security approach, location of defence mechanism implementation, deployment environment of the defence mechanism, kind of malware analysis employed, and the various testbeds used for PoC. This work also highlighted the main benefits and drawbacks of each. The study further provided a comparative analysis of the various works and an outline of the challenges and future work needed in this area. We believe that our study will be helpful for scholars in this sector and security experts interested in investigating the implementation of SaaDM.

Authors' contributions BA helped in the conceptualisation, methodology, literature review and writing of the original draft, VB helped in the conceptualisation, review, and supervision. All authors read and approved the final manuscript.

Declarations

Conflict of interest The authors declare that they have no competing interests as defined by Springer or other interests that might be perceived to influence the results and/or discussion reported in this paper.

References

1. Hood, D.: SDN Architecture issue 1.1. ONF TR-521, p. 59 (2016)
2. Haleplidis, E., Pentikousis, K., Denazis, S., Salim, J.H., Meyer, D., Koufopavlou, O.: Software-defined networking (SDN): layers and architecture terminology. Internet Engineering Task Force, Request for Comments RFC 7426 (2015). https://doi.org/10.17487/RFC7426
3. Cabaj, K., Mazurczyk, W.: Using software-defined networking for ransomware mitigation: the case of cryptowall. IEEE Netw. **30**(6), 14–20 (2016). https://doi.org/10.1109/MNET.2016.1600110NM
4. Wang, Y.C., Ye, R.X.: Credibility-based countermeasure against slow HTTP DoS attacks by using SDN. In: 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), pp. 890–895 (2021). https://doi.org/10.1109/CCWC51732.2021.9375911
5. Al-Mashadani, A.K.A., Ilyas, M.: Distributed denial of service attack alleviated and detected by using mininet and software defined network. Webology **19**(1), 4129–4144 (2022). https://doi.org/10.14704/web/v19i1/web19272
6. Ghosh, U., Chatterjee, P., Shetty, S., Ghosh, U., Chatterjee, P., Shetty, S.: Securing SDN-enabled smart power grids: SDN-enabled smart grid security. https://www.igi-global.com/gateway/chapter/www.igi-global.com/gateway/chapter/204668. Accessed 12 May 12 2022
7. Petroulakis, N.E., Fysarakis, K., Askoxylakis, I., Spanoudakis, G.: Reactive security for SDN/NFV-enabled industrial networks leveraging service function chaining. Trans. Emerg. Telecommun. Technol. **29**(7), e3269 (2018). https://doi.org/10.1002/ett.3269
8. Steichen, M., Hommes, S., State, R.: ChainGuard—A firewall for blockchain applications using SDN with OpenFlow. In: 2017 Principles, Systems and Applications of IP Telecommunications (IPTComm), pp. 1–8 (2017). https://doi.org/10.1109/IPTCOMM.2017.8169748

9. Houda, Z.A.E., Hafid, A., Khoukhi, L.: BrainChain—A machine learning approach for protecting blockchain applications using SDN. In ICC 2020—2020 IEEE International Conference on Communications (ICC), pp. 1–6 (2020). https://doi.org/10.1109/ICC40277.2020.9148808
10. Sahoo, K.S., Sahoo, B., Panda, A.: A secured SDN framework for IoT. In: 2015 International Conference on Man and Machine Interfacing (MAMI), pp. 1–4 (2015). https://doi.org/10.1109/MAMI.2015.7456584
11. Hamza, A., Gharakheili, H.H., Sivaraman, V.: Combining MUD policies with SDN for IoT intrusion detection. In: Proceedings of the 2018 Workshop on IoT Security and Privacy, in IoT S&P'18. Association for Computing Machinery, New York, pp. 1–7 (2018). https://doi.org/10.1145/3229565.3229571
12. Hamza, A., Gharakheili, H.H., Benson, T.A., Sivaraman, V.: Detecting volumetric attacks on loT devices via SDN-based monitoring of MUD activity. In: Proceedings of the 2019 ACM Symposium on SDN Research, in SOSR '19. Association for Computing Machinery, New York, pp. 36–48 (2019). https://doi.org/10.1145/3314148.3314352
13. Javeed, D., Gao, T., Khan, M.T.: SDN-enabled hybrid DL-driven framework for the detection of emerging cyber threats in IoT. Electronics **10**(8), 918 (2021). https://doi.org/10.3390/electronics10080918
14. Rezaei, G., Hashemi, M.R.: An SDN-based firewall for networks with varying security requirements. In 2021 26th International Computer Conference, Computer Society of Iran (CSICC), pp. 1–7 (2021). https://doi.org/10.1109/CSICC52343.2021.9420571
15. Cusack, G., Michel, O., Keller, E.: Machine learning-based detection of ransomware using SDN. In: Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, pp. 1–6 (2018). ACM. https://doi.org/10.1145/3180465.3180467
16. Akbanov, M., Vassilakis, V.G., Logothetis, M.D.: Ransomware detection and mitigation using software-defined networking: the case of WannaCry. Comput. Electr. Eng. **76**, 111–121 (2019). https://doi.org/10.1016/j.compeleceng.2019.03.012
17. Rouka, E., Birkinshaw, C., Vassilakis, V.G.: SDN-based malware detection and mitigation: the Case of ExPetr ransomware. In: 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), pp. 150–155 (2020). https://doi.org/10.1109/ICIoT48696.2020.9089514
18. Alotaibi, F.M., Vassilakis, V.G.: SDN-based detection of self-propagating ransomware: the case of badrabbit. IEEE Access **9**, 28039–28058 (2021). https://doi.org/10.1109/ACCESS.2021.3058897
19. Ropke, C.: SDN malware: problems of current protection systems and potential countermeasures, p. 12 (2016)
20. Open Networking Foundation. OpenFlow Switch Specification (2015). https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf. Accessed 5 Dec 2022
21. Song, H.: Protocol-oblivious forwarding: unleash the power of SDN through a future-proof forwarding plane. In: Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking, in HotSDN '13. Association for Computing Machinery, New York, pp. 127–132 (2013). https://doi.org/10.1145/2491185.2491190
22. Open Networking Foundation. NDM Negotiation OpenFlow Extension (2016). http://opennetworking.wpengine.com/wp-content/uploads/2014/11/TR-536_NDM_Negotiation_OpenFlow_Extension.pdf. Accessed 5 Nov 2022
23. Bosshart, P., et al.: P4: programming protocol-independent packet processors. ACM SIGCOMM Comput. Commun. Rev. **44**(3), 87–95 (2014). https://doi.org/10.1145/2656877.2656890
24. Farrel, A.: An architecture for use of PCE and PCEP in a network with central control (2017). https://tools.ietf.org/id/draft-ietf-teas-pce-central-control-05.html. Accessed 12 Jun 2022
25. Javid, T., Riaz, T., Rasheed, A.: A layer2 firewall for software defined network. In: 2014 Conference on Information Assurance and Cyber Security (CIACS), pp. 39–42 (2014). https://doi.org/10.1109/ciacs.2014.6861329
26. Afek, Y., Bremler-Barr, A., Shafir, L.: Network anti-spoofing with SDN data plane. In: IEEE INFOCOM 2017—IEEE Conference on Computer Communications, pp. 1–9 (2017). https://doi.org/10.1109/INFOCOM.2017.8057008
27. Chen, C.C., Chen, Y.R., Lu, W.C., Tsai, S.C., Yang, M.C.: Detecting amplification attacks with Software Defined Networking. In: 2017 IEEE Conference on Dependable and Secure Computing, pp. 195–201 (2017). https://doi.org/10.1109/DESEC.2017.8073807
28. Aydeger, A., Saputro, N., Akkaya, K.: A moving target defense and network forensics framework for ISP networks using SDN and NFV. Future Gener. Comput. Syst. **94**, 496–509 (2019). https://doi.org/10.1016/j.future.2018.11.045
29. Birkinshaw, C., Rouka, E., Vassilakis, V.G.: 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 (2019). https://doi.org/10.1016/j.jnca.2019.03.005
30. ITU-T. Resolution 77—Standardization work in the ITU Telecommunication Standardization Sector for software-defined networking. ITU (2012). https://www.itu.int:443/en/publications/ITU-T/Pages/publications.aspx. Accessed 17 Jun 2022
31. Boucadair, M., Trossen, D., Farrel, A.: Considerations for the use of SDN in Semantic Routing Networks. Internet Engineering Task Force, Internet Draft (2022). Available: https://datatracker.ietf.org/doc/draft-boucadair-irtf-sdn-and-semantic-routing. Accessed 17 Jun 2022
32. ONF. Software-Defined Networking (SDN) Definition. Open Networking Foundation (2022). https://opennetworking.org/sdn-definition/. Accessed 17 Jun 2022
33. Foukas, X., Marina, M.K., Kontovasilis, K.: Software defined networking concepts. In: Liyanage, M., Gurtov, A., Ylianttila, M. (eds.) Software Defined Mobile Networks (SDMN), pp. 21–44. John Wiley & Sons Ltd, Chichester (2015)
34. Kaljic, E., Maric, A., Njemcevic, P., Hadzialic, M.: A survey on data plane flexibility and programmability in software-defined networking. IEEE Access **7**, 200 (2019). https://doi.org/10.1109/access.2019.2910140
35. Zhang, X., Cui, L., Wei, K., Tso, F.P., Ji, Y., Jia, W.: A survey on stateful data plane in software defined networks. Comput. Netw. **184**, 107597 (2021). https://doi.org/10.1016/j.comnet.2020.107597
36. Hauser, F., et al.: A survey on data plane programming with P4: fundamentals, advances, and applied research. J. Netw. Comput. Appl. **212**, 103561 (2023). https://doi.org/10.1016/j.jnca.2022.103561
37. Zhu, L., et al.: SDN controllers: a comprehensive analysis and performance evaluation study. ACM Comput. Surv. **53**(6), 1–40 (2021). https://doi.org/10.1145/3421764
38. Salman, O., Elhajj, I.H., Kayssi, A., Chehab, A.: SDN controllers: a comparative study. In: 2016 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1–6 (2016). https://doi.org/10.1109/MELCON.2016.7495430
39. Oktian, Y.E., Lee, S., Lee, H., Lam, J.: Distributed SDN controller system: a survey on design choice. Comput. Netw. **121**, 100–111 (2017). https://doi.org/10.1016/j.comnet.2017.04.038
40. Paliwal, M., Shrimankar, D., Tembhurne, O.: Controllers in SDN: a review report. IEEE Access **6**, 36256–36270 (2018). https://doi.org/10.1109/ACCESS.2018.2846236

41. Mamushiane, L., Lysko, A., Dlamini, S.: A comparative evaluation of the performance of popular SDN controllers. In: 2018 Wireless Days (WD), pp. 54–59 (2018). https://doi.org/10.1109/WD.2018.8361694
42. Veena, S., Manju, R.: Detection and mitigation of security attacks using real time SDN analytics. In: 2017 International conference of Electronics, Communication and Aerospace Technology ICECA, pp. 87–93 (2017). https://doi.org/10.1109/iceca.2017.8212770
43. Bhunia, S.S., Gurusamy, M.: Dynamic attack detection and mitigation in IoT using SDN. In: 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), pp. 1–6 (2017). https://doi.org/10.1109/ATNAC.2017.8215418
44. Binu, P.K., Mohan, D., Haridas, E.S.: An SDN-based prototype for dynamic detection and mitigation of DoS attacks in IoT. In: 2021 Third International Conference on Inventive Research in Computing Applications ICIRCA (2021). https://doi.org/10.1109/icirca51532.2021.9544755
45. Darabseh, A., Freris, N.M.: A software-defined architecture for control of IoT cyberphysical systems. Clust. Comput. **22**(4), 1107–1122 (2019). https://doi.org/10.1007/s10586-018-02889-8
46. Islam, Md.J., et al.: Blockchain-SDN-based energy-aware and distributed secure architecture for IoT in smart cities. IEEE Internet Things J. **9**(5), 3850–3864 (2022). https://doi.org/10.1109/JIOT.2021.3100797
47. Manocha, P.S., Kumar, R.: Improved spider monkey optimization-based multi-objective software-defined networking routing with block chain technology for Internet of Things security. Concurr. Comput. Pract. Exp. **34**(11), e6861 (2022). https://doi.org/10.1002/cpe.6861
48. Kalkan, K., Zeadally, S.: Securing internet of things with software defined networking. IEEE Commun. Mag. **56**(9), 186–192 (2018). https://doi.org/10.1109/MCOM.2017.1700714
49. Tijare, P., Vasudevan, D.: The Northbound APIs of Software Defined Networks (2016). 10.5281/zenodo.160891
50. Du, S.G., Lee, J.W., Kim, K.: Proposal of GRPC as a new northbound API for application layer communication efficiency in SDN. In: Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, IMCOM '18. New York, NY, USA: Association for Computing Machinery, pp. 1–6 (2018). https://doi.org/10.1145/3164541.3164563
51. Puppet. Puppet—Powerful infrastructure automation and delivery (2023). https://puppet.com/. Accessed 16 Sept 2022
52. Ansible. Ansible is Simple IT Automation (2023). https://www.ansible.com. Accessed 16 Sept 2022
53. Chef. Chef. Chef Software (2023). https://www.chef.io/. Accessed 16 Sept 2022
54. Lessing, M.: What are SDN Northbound APIs (and SDN REST APIs)?. SDxCentral (2019). https://www.sdxcentral.com/networking/sdn/definitions/what-the-definition-of-software-defined-networking-sdn/north-bound-interfaces-api/. Accessed 16 Sept 2022
55. Costa-Requena, J., et al.: SDN and NFV integration in generalized mobile network architecture. In: 2015 European Conference on Networks and Communications (EuCNC), pp. 154–158 (2015). https://doi.org/10.1109/EuCNC.2015.7194059
56. Bouras, C., Kollia, A., Papazois, A.: SDN & NFV in 5G: advancements and challenges. In: 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), pp. 107–111 (2017). https://doi.org/10.1109/ICIN.2017.7899398
57. Alam, I., et al.: A survey of network virtualization techniques for internet of things using SDN and NFV. ACM Comput. Surv. **53**, 35–40 (2020). https://doi.org/10.1145/3379444
58. Jain, A., Sadagopan, N.S., Lohani, S.K., Vutukuru, M.: A comparison of SDN and NFV for re-designing the LTE Packet Core. In: 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 74–80 (2016). https://doi.org/10.1109/NFV-SDN.2016.7919479
59. Ojo, M., Adami, D., Giordano, S.: A SDN-IoT architecture with NFV implementation. In: 2016 IEEE Globecom Workshops (GC Wkshps), pp. 1–6 (2016). https://doi.org/10.1109/GLOCOMW.2016.7848825
60. Kim, T., Koo, T., Paik, E.: SDN and NFV benchmarking for performance and reliability. In: 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 600–603 (2015). https://doi.org/10.1109/APNOMS.2015.7275403
61. Wang, Q., Shou, G., Liu, Y., Hu, Y., Guo, Z., Chang, W.: Implementation of multipath network virtualization with SDN and NFV. IEEE Access **6**, 32460–32470 (2018). https://doi.org/10.1109/ACCESS.2018.2842058
62. Liyanage, M., Ahmad, I., Ylianttila, M., Gurtov, A., Abro, A.B., de Oca, E.M.: Leveraging LTE security with SDN and NFV. In: 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS), pp. 220–225 (2015). https://doi.org/10.1109/ICIINFS.2015.7399014
63. Hoffmann, M., et al.: SDN and NFV as enabler for the distributed network cloud. Mob. Netw. Appl. **23**(3), 521–528 (2018). https://doi.org/10.1007/s11036-017-0905-y
64. Liu, G., Wood, T.: Cloud-scale application performance monitoring with SDN and NFV. In: 2015 IEEE International Conference on Cloud Engineering, pp. 440–445 (2015). https://doi.org/10.1109/IC2E.2015.45
65. Bernardo, D.V., Chua, B.B.: Introduction and analysis of SDN and NFV security architecture (SN-SECA). In: 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, pp. 796–801 (2015). https://doi.org/10.1109/AINA.2015.270
66. Yousaf, F.Z., Bredel, M., Schaller, S., Schneider, F.: NFV and SDN—key technology enablers for 5G networks. IEEE J. Sel. Areas Commun. **35**(11), 2468–2478 (2017). https://doi.org/10.1109/JSAC.2017.2760418
67. Hasneen, J., Sadique, K.M.: A survey on 5G architecture and security scopes in SDN and NFV. In: Iyer, B., Ghosh, D., Balas, V.E. (eds.) Applied Information Processing Systems, Advances in Intelligent Systems and Computing, pp. 447–460. Springer, Singapore (2022)
68. Cho, H.-H., Lai, C.-F., Shih, T.K., Chao, H.-C.: Integration of SDR and SDN for 5G. IEEE Access **2**, 1196–1204 (2014). https://doi.org/10.1109/ACCESS.2014.2357435
69. Trivisonno, R., Guerzoni, R., Vaishnavi, I., Soldani, D.: SDN-based 5G mobile networks: architecture, functions, procedures and backward compatibility. Trans. Emerg. Telecommun. Technol. **26**(1), 82–92 (2015). https://doi.org/10.1002/ett.2915
70. Sayadi, B., et al.: SDN for 5G mobile networks: NORMA perspective. In: Noguet, D., Moessner, K., Palicot, J. (eds.) Cognitive Radio Oriented Wireless Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 741–753. Springer International Publishing, Cham (2016)
71. Friha, O., Ferrag, M.A., Shu, L., Nafa, M.: A robust security framework based on blockchain and SDN for fog computing enabled agricultural internet of things. In: 2020 International Conference on Internet of Things and Intelligent Applications (ITIA), pp. 1–5 (2020). https://doi.org/10.1109/ITIA50152.2020.9312286
72. Zaidi, Z., Friderikos, V., Yousaf, Z., Fletcher, S., Dohler, M., Aghvami, H.: Will SDN be part of 5G? IEEE Commun. Surv. Tutor. **20**(4), 3220–3258 (2018). https://doi.org/10.1109/COMST.2018.2836315

73. Ksentini, A., Bagaa, M., Taleb, T.: On using SDN in 5G: the controller placement problem. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2016). https://doi.org/10.1109/GLOCOM.2016.7842066
74. Barakabitze, A.A., Ahmad, A., Mijumbi, R., Hines, A.: 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. Comput. Netw. **167**, 106984 (2020). https://doi.org/10.1016/j.comnet.2019.106984
75. Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J.J., Lorca, J., Folgueira, J.: Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges. IEEE Commun. Mag. **55**(5), 80–87 (2017). https://doi.org/10.1109/MCOM.2017.1600935
76. Zhou, X., Li, R., Chen, T., Zhang, H.: Network slicing as a service: enabling enterprises' own software-defined cellular networks. IEEE Commun. Mag. **54**(7), 146–153 (2016). https://doi.org/10.1109/MCOM.2016.7509393
77. Scano, D., Valcarenghi, L., Kondepu, K., Castoldi, P., Giorgetti, A.: Network slicing in SDN networks. In: 2020 22nd International Conference on Transparent Optical Networks (ICTON), pp. 1–4 (2020). https://doi.org/10.1109/ICTON51198.2020.9203184
78. Le, L.V., Lin, B.S.P., Tung, L.P., Sinh, D.: SDN/NFV, Machine Learning, and Big Data Driven Network Slicing for 5G. In: 2018 IEEE 5G World Forum (5GWF), pp. 20–25 (2018). https://doi.org/10.1109/5GWF.2018.8516953
79. Chartsias, P.K., et al.: SDN/NFV-based end to end network slicing for 5G multi-tenant networks. In: 2017 European Conference on Networks and Communications (EuCNC), pp. 1–5 (2017). https://doi.org/10.1109/EuCNC.2017.7980670
80. Costa-Requena, J., Poutanen, A., Vural, S., Kamel, G., Clark, C., Roy, S.K.: SDN-based UPF for mobile backhaul network slicing. In: 2018 European Conference on Networks and Communications (EuCNC), pp. 48–53 (2018). https://doi.org/10.1109/EuCNC.2018.8442795
81. Demirci, S., Demirci, M., Sagiroglu, S.: Virtual security functions and their placement in software defined networks: a survey. Gazi Univ. J. Sci. **32**(3), 833–851 (2019). https://doi.org/10.35378/gujs.422000
82. Demirci, S., Sagiroglu, S.: Optimal placement of virtual network functions in software defined networks: a survey. J. Netw. Comput. Appl. **147**, 102424 (2019). https://doi.org/10.1016/j.jnca.2019.102424
83. Vineetha. Dynamic service function chaining of network functions using SDN (2016). Available: https://www.semanticscholar.org/paper/Dynamic-Service-Function-Chaining-of-Network-Using-Vineetha/bb6f3f8951a23743ec387712bf177d8e0632f05c. Accessed 2 May 2023
84. Coronado, E., et al.: Zero touch management: a survey of network automation solutions for 5G and 6G networks. IEEE Commun. Surv. Tutor. **24**(4), 2535–2578 (2022). https://doi.org/10.1109/COMST.2022.3212586
85. Slamnik-Krijestorac, N., Kremo, H., Ruffini, M., Marquez-Barja, J.M.: Sharing distributed and heterogeneous resources toward end-to-end 5G networks: a comprehensive survey and a taxonomy. IEEE Commun. Surv. Tutor. **22**(3), 1592–1628 (2020). https://doi.org/10.1109/COMST.2020.3003818
86. Ullah, Y., Roslee, M.B., Mitani, S.M., Khan, S.A., Jusoh, M.H.: a survey on handover and mobility management in 5G HetNets: current state, challenges, and future directions. Sensors **23**(11), 5081 (2023). https://doi.org/10.3390/s23115081
87. Basilier, H., Lemark, J., Centonza, A., Asberg, T.: Applied network slicing scenarios in 5G. Ericsson Technol. Rev. **2021**(2), 2–11 (2021). https://doi.org/10.23919/ETR.2021.9904667
88. Ezekiel, S., Divakaran, D.M., Gurusamy, M.: Dynamic attack mitigation using SDN. In: 27th International Telecommunication Networks and Applications Conference (ITNAC), pp. 1–6 (2017). https://doi.org/10.1109/atnac.2017.8215430
89. Sahay, R., Blanc, G., Zhang, Z., Debar, H.: Towards autonomic DDoS mitigation using Software Defined Networking. In: SENT 2015: NDSS Workshop on Security of Emerging Networking Technologies (2017). Internet Society, San Diego. https://doi.org/10.14722/sent.2015.23004
90. Abdulqadder, I.H., Zhou, S., Zou, D., Aziz, I.T., Akber, S.M.A.: Bloc-sec: blockchain-based lightweight security architecture for 5G/B5G enabled SDN/NFV cloud of IoT. In: 2020 IEEE 20th International Conference on Communication Technology (ICCT), pp. 499–507 (2020). https://doi.org/10.1109/ICCT50939.2020.9295823
91. Varadharajan, V., Tupakula, U., Karmakar, K.K.: Techniques for securing 5G network services from attacks. In: 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 273–280 (2021). https://doi.org/10.1109/TrustCom53373.2021.00052
92. Krishnan, P., Duttagupta, S., Achuthan, K.: SDN/NFV security framework for fog-to-things computing infrastructure. Softw. Pract. Exp. **50**(5), 757–800 (2020). https://doi.org/10.1002/spe.2761
93. Thorat, P., Dubey, N.K., Khetan, K., Challa, R.: SDN-based predictive alarm manager for security attacks detection at the IoT gateways. In: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), pp. 1–2 (2021). https://doi.org/10.1109/CCNC49032.2021.9369623
94. Ullah, I., Raza, B., Ali, S., Abbasi, I.A., Baseer, S., Irshad, A.: Software defined network enabled fog-to-things hybrid deep learning driven cyber threat detection system. Secur. Commun. Netw. **2021**, 1–15 (2021). https://doi.org/10.1155/2021/6136670
95. Wani, A., Khaliq, R.: SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). CAAI Trans. Intell. Technol. **6**, 281–290 (2021). https://doi.org/10.1049/cit2.12003
96. Wani, A., Revathi, S.: DDoS detection and alleviation in IoT using SDN (SDIoT-DDoS-DA). J. Inst. Eng. India Ser. B **101**(2), 117–128 (2020). https://doi.org/10.1007/s40031-020-00442-z
97. Guha Roy, D., Srirama, S.N.: A blockchain-based cyber attack detection scheme for decentralized internet of things using software-defined network. Softw. Pract. Exp. **51**, 1540–1556 (2021). https://doi.org/10.1002/spe.2972
98. Shafi, Q., Basit, A., Qaisar, S., Koay, A., Welch, I.: Fog-assisted SDN controlled framework for enduring anomaly detection in an IoT network. IEEE Access **6**, 73713–73723 (2018). https://doi.org/10.1109/ACCESS.2018.2884293
99. Wang, S., Gomez, K.M., Sithamparanathan, K., Zanna, P.: Software defined network security framework for IoT based smart home and city applications. In: 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1–8 (2019). https://doi.org/10.1109/ICSPCS47537.2019.9008703
100. Girdler, T., Vassilakis, V.G.: Implementing an intrusion detection and prevention system using Software-Defined Networking: defending against ARP spoofing attacks and Blacklisted MAC Addresses. Comput. Electr. Eng. **90**, 106990 (2021). https://doi.org/10.1016/j.compeleceng.2021.106990
101. Pérez-Díaz, J.A., Valdovinos, I.A., Choo, K.-K.R., Zhu, D.: A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access **8**, 155859–155872 (2020). https://doi.org/10.1109/ACCESS.2020.3019330
102. Jin, D., et al.: Toward a cyber resilient and secure microgrid using software-defined networking. IEEE Trans. Smart Grid **8**(5), 2494–2504 (2017). https://doi.org/10.1109/TSG.2017.2703911

103. Hussein, A., Elhajj, I.H., Chehab, A., Kayssi, A.: SDN VANETs in 5G: an architecture for resilient security services. In: 2017 Fourth International Conference on Software Defined Systems (SDS), pp. 67–74 (2017). https://doi.org/10.1109/SDS.2017.7939143
104. Li, J., Zhao, Z., Li, R.: Machine learning-based IDS for software-defined 5G network. IET Netw. **7**(2), 53–60 (2018). https://doi.org/10.1049/iet-net.2017.0212
105. Medhane, D.V., Sangaiah, A.K., Hossain, M.S., Muhammad, G., Wang, J.: Blockchain-enabled distributed security framework for next-generation IoT: an edge cloud and software-defined network-integrated approach. IEEE Internet Things J. **7**(7), 6143–6149 (2020). https://doi.org/10.1109/JIOT.2020.2977196
106. Yin, D., Zhang, L., Yang, K.: A DDoS attack detection and mitigation with software-defined internet of things framework. IEEE Access **6**, 24694–24705 (2018). https://doi.org/10.1109/ACCESS.2018.2831284
107. Bhayo, J., Jafaq, R., Ahmed, A., Hameed, S., Shah, S.A.: A time-efficient approach toward DDoS attack detection in IoT network using SDN. IEEE Internet Things J. **9**(5), 3612–3630 (2022). https://doi.org/10.1109/JIOT.2021.3098029
108. Manso, P., Moura, J., Serrão, C.: SDN-based intrusion detection system for early detection and mitigation of DDoS attacks. Information (2019). https://doi.org/10.3390/info10030106
109. Buragohain, C., Medhi, N.: FlowTrApp: an SDN based architecture for DDoS attack detection and mitigation in data centers. In: 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), pp. 519–524 (2017). https://doi.org/10.1109/SPIN.2016.7566750
110. Bhushan, K., Gupta, B.B.: Detecting DDoS attack using software defined network (SDN) in cloud computing environment. In: 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), pp. 872–877 (2018). https://doi.org/10.1109/SPIN.2018.8474062
111. Steadman, J., Scott-Hayward, S.: DNSxD: detecting data exfiltration over DNS. In: 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–6 (2018). IEEE, Verona. https://doi.org/10.1109/NFV-SDN.2018.8725640
112. Hong, K., Kim, Y., Choi, H., Park, J.: SDN-assisted slow HTTP DDoS attack defense method. IEEE Commun. Lett. **22**(4), 688–691 (2018). https://doi.org/10.1109/LCOMM.2017.2766636
113. Lukaseder, T., Maile, L., Erb, B., Kargl, F.: SDN-assisted network-based mitigation of slow DDoS attacks. In: Beyah, R., Chang, B., Li, Y., Zhu, S. (eds.) Security and Privacy in Communication Networks, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 102–121. Springer International Publishing, Cham (2018)
114. Sanjeetha, R., Ajay Shastry, K.N., Chetan, H.R., Kanavalli, A.: Mitigating HTTP GET FLOOD DDoS attack using an SDN controller. In: 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), pp. 6–10 (2020). https://doi.org/10.1109/rteict49044.2020.9315608
115. Sharma, P.K., Park, J.H., Jeong, Y.-S., Park, J.H.: SHSec: SDN based secure smart home network architecture for internet of things. Mob. Netw. Appl. **24**(3), 913–924 (2019). https://doi.org/10.1007/s11036-018-1147-3
116. Bawany, N.Z., Shamsi, J.A.: SEAL: SDN based secure and agile framework for protecting smart city applications from DDoS attacks. J. Netw. Comput. Appl. **145**, 102381 (2019). https://doi.org/10.1016/j.jnca.2019.06.001
117. Mahmood, H., Mahmood, D., Shaheen, Q., Akhtar, R., Changda, W.: S-DPS: an SDN-based DDoS protection system for smart grids. Secur. Commun. Netw. **2021**, e6629098 (2021). https://doi.org/10.1155/2021/6629098
118. Forland, M.K., Kralevska, K., Garau, M., Gligoroski, D.: Preventing DDoS with SDN in 5G. In: 2019 IEEE Globecom Workshops (GC Wkshps), pp. 1–7 (2019). https://doi.org/10.1109/GCWkshps45667.2019.9024497
119. Cabaj, K., Gregorczyk, M., Mazurczyk, W., Nowakowski, P., Żórawski, P.: Network threats mitigation using software-defined networking for the 5G internet of radio light system. Secur. Commun. Netw. **2019**, e4930908 (2019). https://doi.org/10.1155/2019/4930908
120. Cabaj, K., Gregorczyk, M., Mazurczyk, W., Nowakowski, P., Żórawski, P.: SDN-based mitigation of scanning attacks for the 5G internet of radio light system. In: Proceedings of the 13th International Conference on Availability, Reliability and Security, pp. 1–10 (2018). ACM, Hamburg. https://doi.org/10.1145/3230833.3233248
121. Yazdinejad, A., Parizi, R.M., Dehghantanha, A., Choo, K.-K.R.: Blockchain-enabled authentication handover with efficient privacy protection in SDN-based 5G networks. IEEE Trans. Netw. Sci. Eng. **8**(2), 1120–1132 (2021). https://doi.org/10.1109/TNSE.2019.2937481
122. Abdulqadder, I.H., Zou, D., Aziz, I.T., Yuan, B.: Enhanced attack aware security provisioning scheme in SDN/NFV enabled over 5G network. In: 2018 27th International Conference on Computer Communication and Networks (ICCCN), pp. 1–9 (2018). https://doi.org/10.1109/ICCCN.2018.8487339
123. Huertas Celdrán, A., Gil Pérez, M., García Clemente, F.J., Martínez Pérez, G.: Towards the autonomous provision of self-protection capabilities in 5G networks. J. Ambient Intell. Humaniz. Comput. **10**, 4707–4720 (2019). https://doi.org/10.1007/s12652-018-0848-6
124. Khettab, Y., Bagaa, M., Dutra, D.L.C., Taleb, T., Toumi, N.: Virtual security as a service for 5G verticals. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–6 (2018). https://doi.org/10.1109/WCNC.2018.8377298
125. Sasan, Z., Salehi, M.: SDN-based defending against ARP poisoning attack. J. Adv. Comput. Res. **8**(2), 95–102 (2017)
126. Wang, J., Wen, R., Li, J., Yan, F., Zhao, B., Yu, F.: Detecting and mitigating target link-flooding attacks using SDN. IEEE Trans. Dependable Secure Comput. **16**(6), 944–956 (2019). https://doi.org/10.1109/TDSC.2018.2822275
127. Mohammadi, R., Javidan, R., Conti, M.: SLICOTS: an SDN-based lightweight countermeasure for TCP SYN flooding attacks. IEEE Trans. Netw. Serv. Manag. **14**(2), 487–497 (2017). https://doi.org/10.1109/TNSM.2017.2701549
128. Wallace, V., Scott-Hayward, S.: Can SDN deanonymize Bitcoin users?. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), pp. 1–7 (2020). https://doi.org/10.1109/ICC40277.2020.9148936
129. Bawany, N.Z., Shamsi, J.A.: Application layer DDoS attack defense framework for smart city using SDN (2016). Available: https://www.researchgate.net/profile/Natalie-Walker-15/publication/302960855_Proceedings_of_the_Third_International_Conference_on_Computer_Science_Computer_Engineering_and_Social_Media_CSCESM2016_Thessaloniki_Greece_2016/links/5739d1c808ae9ace840db301/Proceedings-of-the-Third-International-Conference-on-Computer-Science-Computer-Engineering-and-Social-Media-CSCESM2016-Thessaloniki-Greece-2016.pdf#page=3
130. Xiong, A., et al.: A distributed security SDN cluster architecture for smart grid based on blockchain technology. Secur. Commun. Netw. **2021**, e9495093 (2021). https://doi.org/10.1155/2021/9495093
131. Navid, W., Bhutta, M.N.M.: Detection and mitigation of Denial of Service (DoS) attacks using performance aware Software Defined

Networking (SDN). In: 2017 International Conference on Information and Communication Technologies (ICICT), pp. 47–57 (2017). https://doi.org/10.1109/ICICT.2017.8320164
132. Fawcett, L., Scott-Hayward, S., Broadbent, M., Wright, A., Race, N.: Tennison: a distributed SDN framework for scalable network security. IEEE J. Sel. Areas Commun. **36**(12), 2805–2818 (2018). https://doi.org/10.1109/JSAC.2018.2871313
133. Flauzac, O., Robledo, E.G., Gonzalez, C., Mauhourat, F., Nolot, F.: SDN Architecture to prevent attacks with OpenFlow. In: 2020 8th International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6 (2020). https://doi.org/10.1109/WINCOM50532.2020.9272445
134. Chowdhary, A., et al.: SDFW: SDN-based stateful distributed firewall. arXiv. https://doi.org/10.48550/arXiv.1811.00634
135. Biju, J.M., Prakash, A.J.: Phishdect & mitigator: SDN based phishing attack detection (2019)
136. Wang, X., Xu, K., Chen, W., Li, Q., Shen, M., Wu, B.: ID-based SDN for the internet of things. IEEE Netw. **34**(4), 76–83 (2020). https://doi.org/10.1109/MNET.011.1900380
137. Yu, T., Rui, L., Qiu, X.: SDNDefender: a comprehensive DDoS defense mechanism using hybrid approaches over software defined networking. Secur. Commun. Netw. (2021). https://doi.org/10.1155/2021/5097267
138. Jia, K., Liu, C., Liu, Q., Wang, J., Liu, J., Liu, F.: A lightweight DDoS detection scheme under SDN context. Cybersecurity **5**(1), 27 (2022). https://doi.org/10.1186/s42400-022-00128-7
139. Badotra, S., Singh, J.: Creating firewall in transport layer and application layer using software defined networking. In: Saini, H.S., Sayal, R., Govardhan, A., Buyya, R. (es.) Innovations in Computer Science and Engineering, Lecture Notes in Networks and Systems, vol. 32, pp. 95–103 (2019). Springer Singapore, Singapore. https://doi.org/10.1007/978-981-10-8201-6_11
140. Vempati, J., Dantu, R., Badruddoja, S., Thompson, M.: Adaptive and predictive SDN control during DDoS attacks. In: 2020 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 1–6 (2020). https://doi.org/10.1109/isi49825.2020.9280542
141. Beigi-Mohammadi, N., Barna, C., Shtern, M., Khazaei, H., Litoiu, M.: CAAMP: completely automated DDoS attack mitigation platform in hybrid clouds. In: 2016 12th International Conference on Network and Service Management (CNSM), pp. 136–143 (2017). https://doi.org/10.1109/CNSM.2016.7818409
142. Chen, M.H., Ciou, J.Y., Chung, I.H., Chou, C.F.: FlexProtect: a SDN-based DDoS attack protection architecture for multi-tenant data centers. In: Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region, HPC Asia 2018, pp. 202–209 (2018). Association for Computing Machinery, New York. https://doi.org/10.1145/3149457.3149476
143. Kim, S., Lee, S., Cho, G., Ahmed, M.E., Jeong, J., Kim, H.: Preventing DNS amplification attacks using the history of DNS queries with SDN. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security—ESORICS 2017, Lecture Notes in Computer Science, pp. 135–152 (2017). Springer International Publishing, Cham. https://doi.org/10.1007/978-3-319-66399-9_8
144. Yuan, B., Zou, D., Jin, H., Yu, S., Yang, L.T.: HostWatcher: Protecting hosts in cloud data centers through software-defined networking. Future Gener. Comput. Syst. **105**, 964–972 (2020). https://doi.org/10.1016/j.future.2017.04.023
145. Gonçalves, D.S.M., Couto, R.S., Rubinstein, M.G.: A protection system against HTTP flood attacks using software defined networking. J. Netw. Syst. Manag. **31**(1), 16 (2022). https://doi.org/10.1007/s10922-022-09704-1
146. Gheisari, M., Wang, G., Khan, W.Z., Fernández-Campusano, C.: A context-aware privacy-preserving method for IoT-based smart city using Software Defined Networking. Comput. Secur. **87**, 101470 (2019). https://doi.org/10.1016/j.cose.2019.02.006
147. Al-Sakran, H., Alharbi, Y., Serguievskaia, I.: Framework architecture for securing IoT using blockchain, smart contract and software defined network technologies. In: 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), pp. 1–6 (2019). https://doi.org/10.1109/ICTCS.2019.8923080
148. Abou El Houda, Z., Hafid, A.S., Khoukhi, L.: Cochain-SC: an intra- and inter-domain Ddos mitigation scheme based on blockchain using SDN and smart contract. IEEE Access **7**, 98893–98907 (2019). https://doi.org/10.1109/ACCESS.2019.2930715
149. Rahman, A., et al.: DistB-Condo: distributed blockchain-based IoT-SDN model for smart condominium. IEEE Access **8**, 209594–209609 (2020). https://doi.org/10.1109/ACCESS.2020.3039113
150. Pourvahab, M., Ekbatanifard, G.: Digital forensics architecture for evidence collection and provenance preservation in IaaS cloud environment using SDN and blockchain technology. IEEE Access **7**, 153349–153364 (2019)
151. Garg, S., Kaur, K., Kaddoum, G., Ahmed, S.H., Jayakody, D.N.K.: SDN-based secure and privacy-preserving scheme for vehicular networks: a 5G perspective. IEEE Trans. Veh. Technol. **68**(9), 8421–8434 (2019). https://doi.org/10.1109/TVT.2019.2917776
152. Hakiri, A., Dezfouli, B.: Towards a blockchain-SDN architecture for secure and trustworthy 5G massive IoT networks. In: Proceedings of the 2021 ACM International Workshop on Software Defined Networks & Network Function Virtualization Security, SDN-NFV Sec'21. Association for Computing Machinery, New York, pp. 11–18 (2021). https://doi.org/10.1145/3445968.3452090
153. Camilo, G.F., Rebello, G.A.F., de Souza, L.A.C., B. Duarte, O.C.M.: AutAvailChain: automatic and secure data availability through blockchain. In: GLOBECOM 2020—2020 IEEE Global Communications Conference, pp. 1–6 (2020). https://doi.org/10.1109/GLOBECOM42002.2020.9322396
154. Iqbal, W., Abbas, H., Rauf, B., Bangash, Y.A., Amjad, M.F., Hemani, A.: PCSS: privacy preserving communication scheme for SDN enabled smart homes. IEEE Sens. J. **22**(18), 17677–17690 (2022). https://doi.org/10.1109/JSEN.2021.3087779
155. Yao, J., Han, Z., Sohail, M., Wang, L.: A robust security architecture for SDN-based 5G networks. Future Internet (2019). https://doi.org/10.3390/fi11040085
156. Sutrala, A.K., Obaidat, M.S., Saha, S., Das, A.K., Alazab, M., Park, Y.: Authenticated key agreement scheme with user anonymity and untraceability for 5G-enabled softwarized industrial Cyber-physical systems. IEEE Trans. Intell. Transp. Syst. **23**(3), 2316–2330 (2022). https://doi.org/10.1109/TITS.2021.3056704
157. Debroy, S., Calyam, P., Nguyen, M., Stage, A., Georgiev, V.: Frequency-minimal moving target defense using software-defined networking. In: 2016 International Conference on Computing, Networking and Communications (ICNC), pp. 1–6 (2016). https://doi.org/10.1109/ICCNC.2016.7440635
158. Karmakar, K.K., Varadharajan, V., Nepal, S., Tupakula, U.: SDN-enabled secure IoT architecture. IEEE Internet Things J. **8**(8), 6549–6564 (2021). https://doi.org/10.1109/JIOT.2020.3043740
159. Rietz, R., Cwalinski, R., König, H., Brinner, A.: An SDN-based approach to ward Off LAN attacks. J. Comput. Netw. Commun. **2018**, e4127487 (2018). https://doi.org/10.1155/2018/4127487
160. Sahri, N., Okamura, K.: Protecting DNS services from IP spoofing: SDN collaborative authentication approach. In: Proceedings of the 11th International Conference on Future Internet Technologies, CFI '16. Association for Computing Machinery, New York, pp. 83–89 (2017). https://doi.org/10.1145/2935663.2935666
161. Nife, F., Kotulski, Z., Reyad, O.: New SDN-oriented distributed network security system. Appl. Math. Inf. Sci. **12**, 673–683 (2018). https://doi.org/10.18576/amis/120401

162. Cox, J.H., Clark, R.J., Owen, H.L.: Leveraging SDN for ARP security. In: SoutheastCon 2016, pp. 1–8 (2017). https://doi.org/10.1109/SECON.2016.7506644
163. Galeano-Brajones, J., Carmona-Murillo, J., Valenzuela-Valdes, J.F., Luna-Valero, F.: Detection and mitigation of DoS and DDoS attacks in IoT-based stateful SDN: an experimental approach. Sensors **20**(3), 816 (2020). https://doi.org/10.3390/s20030816
164. Yang, L., Zhao, H.: DDoS attack identification and defense using SDN based on machine learning method. In: 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), pp. 174–178 (2018). https://doi.org/10.1109/I-SPAN.2018.00036
165. Mohammadi, R., Lal, C., Conti, M., Sharma, L.: Software defined network-based HTTP flooding attack defender. Comput. Electr. Eng. **101**, 108019 (2022). https://doi.org/10.1016/j.compeleceng.2022.108019
166. Sumantra, I., Indira Gandhi, S.: DDoS attack detection and mitigation in software defined networks. In: 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), pp. 1–5 (2020). https://doi.org/10.1109/ICSCAN49426.2020.9262408
167. El Houda, Z.A., Khoukhi, L., Hafid, A.: ChainSecure—A scalable and proactive solution for protecting blockchain applications using SDN. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2018). https://doi.org/10.1109/GLOCOM.2018.8647279
168. Javeed, D., Gao, T., Khan, M.T., Ahmad, I., Ahmad, I.: A hybrid deep learning-driven SDN enabled mechanism for secure communication in internet of things (IoT). Sensors **21**, 4884 (2021). https://doi.org/10.3390/s21144884
169. Ding, P., Li, J., Wang, L., Wen, M., Guan, Y.: HYBRID-CNN: an efficient scheme for abnormal flow detection in the SDN-based smart grid. Secur. Commun. Netw. **2020**, e8850550 (2020). https://doi.org/10.1155/2020/8850550
170. Razib, M.A., Javeed, D., Khan, M.T., Alkanhel, R., Muthanna, M.S.A.: Cyber threats detection in smart environments using SDN-enabled DNN-LSTM hybrid framework. IEEE Access **10**, 53015–53026 (2022). https://doi.org/10.1109/ACCESS.2022.3172304
171. Miao, M., Wu, B., A Flexible Phishing Detection Approach Based on Software-Defined Networking Using Ensemble Learning Method. In: Proceedings of the 2020 4th International Conference on High Performance Compilation, Computing and Communications, HP3C 2020. Association for Computing Machinery, New York, pp. 70–73 (2020). https://doi.org/10.1145/3407947.3407952
172. Tawfik, M., Al-Zidi, N.M., Alsellami, B., Al-Hejri, A.M., Nimbhore, S.: Internet of things-based middleware against cyber-attacks on smart homes using software-Defined networking and deep learning. In: 2021 2nd International Conference on Computational Methods in Science & Technology (ICCMST), pp. 7–13 (2021). https://doi.org/10.1109/ICCMST54943.2021.00014
173. Mazhar, N., Salleh, R., Zeeshan, M., Hameed, M.M., Khan, N.: R-IDPS: real time SDN based IDPS system for IoT security. In: 2021 IEEE 18th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET), pp. 71–76 (2021). https://doi.org/10.1109/HONET53078.2021.9615449
174. Farhin, F., Sultana, I., Islam, N., Kaiser, M.S., Rahman, M.S., Mahmud, M.: Attack Detection in internet of things using software defined network and fuzzy neural network. In: 2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), pp. 1–6 (2020). https://doi.org/10.1109/ICIEVicIVPR48672.2020.9306666
175. Demirpolat, A., Sarica, A.K., Angin, P.: ProtÉdge: a few-shot ensemble learning approach to software-defined networking-assisted edge security. Trans. Emerg. Telecommun. Technol. **32**(6), e4138 (2021). https://doi.org/10.1002/ett.4138
176. Zha, Z., Wang, A., Guo, Y., Montgomery, D., Chen, S.: BotSifter: an SDN-based online bot detection framework in data centers. In: 2019 IEEE Conference on Communications and Network Security (CNS), pp. 142–150 (2019). https://doi.org/10.1109/CNS.2019.8802854
177. Satheesh, N., et al.: Flow-based anomaly intrusion detection using machine learning model with software defined networking for OpenFlow network. Microprocess. Microsyst. **79**, 103285 (2020). https://doi.org/10.1016/j.micpro.2020.103285
178. Xiao, Y., Liu, J., Zhang, L.: Cyber-physical system intrusion detection model based on software-defined network. In: 2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS), pp. 170–173 (2021). https://doi.org/10.1109/ICSESS52187.2021.9522345
179. Masoud, M., Jaradat, Y., Ahmad, A.Q.: On tackling social engineering web phishing attacks utilizing software defined networks (SDN) approach. In: 2016 2nd International Conference on Open Source Software Computing (OSSCOM), pp. 1–6 (2017). https://doi.org/10.1109/OSSCOM.2016.7863679
180. Yungaicela-Naula, N.M., Vargas-Rosales, C., Perez-Diaz, J.A.: SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning. IEEE Access **9**, 108495–108512 (2021). https://doi.org/10.1109/ACCESS.2021.3101650
181. Gaba, S., Budhiraja, I., Makkar, A., Garg, D.: Machine learning for detecting security attacks on blockchain using software defined networking. In: 2022 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 260–264 (2022). https://doi.org/10.1109/ICCWorkshops53468.2022.9814656
182. Alwabisi, S., Ouni, R., Saleem, K.: Using machine learning and software-defined networking to detect and mitigate DDoS attacks in fiber-optic networks. Electronics (2022). https://doi.org/10.3390/electronics11234065
183. Kao, Y.C., Liu, J.C., Wang, Y.H., Chu, Y.H., Tsai, S.C., Lin, Y.B.: Automatic blocking mechanism for information security with SDN. J. Internet Serv. Inf. Secur. **9**, 60–73 (2019)
184. Baiju, B.V.: Ddos attack detection using SDN techniques. Turk. J. Comput. Math. Educ. TURCOMAT **12**, 326–335 (2021). https://doi.org/10.17762/turcomat.v12i10.4174
185. Zheng, J., Li, Q., Gu, G., Cao, J., Yau, D.K.Y., Wu, J.: Realtime DDoS defense using COTS SDN switches via adaptive correlation analysis. IEEE Trans. Inf. Forensics Secur. **13**(7), 1838–1853 (2018). https://doi.org/10.1109/TIFS.2018.2805600
186. Gupta, B.B., Chaturvedi, C.: Software defined networking (SDN) based secure integrated framework against distributed denial of service (DDoS) attack in cloud environment. In: 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 1310–1315 (2019). https://doi.org/10.1109/ICCES45898.2019.9002596
187. Revathi, M., Ramalingam, V.V., Amutha, B.: A machine learning based detection and mitigation of the DDOS attack by using SDN controller framework. Wirel. Pers. Commun. (2021). https://doi.org/10.1007/s11277-021-09071-1
188. Ghosh, U., Chatterjee, P., Shetty, S.: A security framework for SDN-enabled smart power grids. In: 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), pp. 113–118 (2017). https://doi.org/10.1109/ICDCSW.2017.20
189. Cox, J.H., Clark, R.J., Owen, H.L.: Leveraging SDN and WebRTC for rogue access point security. IEEE Trans. Netw. Serv. Manag. **14**(3), 756–770 (2017). https://doi.org/10.1109/tnsm.2017.2710623

190. Ferreira, F.A., Saotome, O.: Cyber Security Architecture in Smart Grids Using Software Defined Networks, p. 3 (2017)
191. Umar, R., Riadi, I., Kusuma, R.: Mitigating sodinokibi ransomware attack on cloud network using software-defined networking (SDN). Int. J. Saf. Secur. Eng. **11**, 239–246 (2021). https://doi.org/10.18280/ijsse.110304
192. Zolotukhin, M., Hämäläinen, T.: On artificial intelligent malware tolerant networking for IoT. In: 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), pp. 1–6 (2018). https://doi.org/10.1109/NFV-SDN.2018.8725767
193. Hu, Y., Zheng, K., Wang, X., Yang, Y.: WORM-HUNTER: a worm guard system using software-defined networking. KSII Trans. Internet Inf. Syst. TIIS **11**(1), 484–510 (2017). https://doi.org/10.3837/tiis.2017.01.026
194. Ceron, J.M., Margi, C.B., Granville, L.Z.: MARS: from traffic containment to network reconfiguration in malware-analysis systems. Comput. Netw. **129**, 261–272 (2017). https://doi.org/10.1016/j.comnet.2017.10.003
195. Karakate, M., Esaki, H., Ochiai, H.: SDNHive: a proof-of-concept SDN and honeypot system for defending against internal threats. In: 2021 the 11th International Conference on Communication and Network Security, ICCNS 2021, pp. 9–20 (2021). Association for Computing Machinery, New York. https://doi.org/10.1145/3507509.3507511
196. Chang, H.Y., Lin, T.L., Hsu, T.F., Shen, Y.S., Li, G.R.: Implementation of ransomware prediction system based on weighted-KNN and real-time isolation architecture on SDN Networks. In: 2019 IEEE International Conference on Consumer Electronics—Taiwan (ICCE-TW), pp. 1–2 (2019). https://doi.org/10.1109/ICCE-TW46550.2019.8991771
197. Thapa, C., Karmakar, K.K., Celdran, A.H., Camtepe, S., Varadharajan, V., Nepal, S.: FedDICE: a ransomware spread detection in a distributed integrated clinical environment using federated learning and SDN based mitigation. In: Yuan, X., Bao, W., Yi, X., Tran, N.H. (eds.) Quality, Reliability, Security and Robustness in Heterogeneous Systems, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 3–24. Springer International Publishing, Cham (2021)
198. Ahmed, J., Gharakheili, H.H., Russell, C., Sivaraman, V.: Automatic detection of DGA-enabled malware using SDN and traffic behavioral modeling. IEEE Trans. Netw. Sci. Eng. **9**(4), 2922–2939 (2022). https://doi.org/10.1109/TNSE.2022.3173591
199. Wazirali, R., Ahmad, R., Abu-Ein, A.A.-K.: Sustaining accurate detection of phishing URLs using SDN and feature selection approaches. Comput. Netw. **201**, 108591 (2021). https://doi.org/10.1016/j.comnet.2021.108591
200. Cabaj, K., Gregorczyk, M., Mazurczyk, W.: Software-defined networking-based crypto ransomware detection using HTTP traffic characteristics. Comput. Electr. Eng. **66**, 353–368 (2018). https://doi.org/10.1016/j.compeleceng.2017.10.012
201. Raunak, P., Krishnan, P.: Network detection of ransomware delivered by exploit kit. ARPN J. Eng. Appl. Sci. **12**, 3885–3889 (2017)
202. Nguyen, M., Pal, A., Debroy, S.: Whack-a-Mole: Software-defined Networking driven Multi-level DDoS defense for Cloud environments. In: 2018 IEEE 43rd Conference on Local Computer Networks (LCN), pp. 493–501 (2018). https://doi.org/10.1109/LCN.2018.8638054
203. Hyder, M.F., Ismail, M.A.: INMTD: intent-based moving target defense framework using software defined networks. Eng. Technol. Appl. Sci. Res. **10**(1), 5142–5147 (2020). https://doi.org/10.48084/etasr.3266
204. Zhao, Z., Liu, F., Gong, D.: An SDN-based fingerprint hopping method to prevent fingerprinting attacks. Secur. Commun. Netw. **2017**, e1560594 (2017). https://doi.org/10.1155/2017/1560594
205. Chiba, S., Guillen, L., Izumi, S., Abe, T., Suganuma, T.: An SDN-based moving target defense as a countermeasure to prevent network scans. IEICE Trans. Commun. (2022). https://doi.org/10.1587/transcom.2021TMP0020
206. Chiba, S., Guillen, L., Izumi, S., Abe, T., Suganuma, T.: Design of a network scan defense method by combining an SDN-based MTD and IPS. In: 2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS), pp. 273–278 (2021). https://doi.org/10.23919/APNOMS52696.2021.9562686
207. Abdelkhalek, M., Hyder, B., Govindarasu, M., Rieger, C.G.: Moving target defense routing for SDN-enabled smart grid. In: 2022 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 215–220 (2022). https://doi.org/10.1109/CSR54599.2022.9850341
208. Galadima, H., Seeam, A., Ramsurrun, V.: Cyber deception against DDoS attack using moving target defence framework in SDN IOT-EDGE networks. In: 2022 3rd International Conference on Next Generation Computing Applications (NextComp), pp. 1–6 (2022). https://doi.org/10.1109/NextComp55567.2022.9932172
209. Shi, Y., et al.: CHAOS: an SDN-based moving target defense system. Secur. Commun. Netw. **2017**, e3659167 (2017). https://doi.org/10.1155/2017/3659167
210. Steinberger, J., et al.: DDoS defense using MTD and SDN. In: NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, pp. 1–9 (2018). https://doi.org/10.1109/NOMS.2018.8406221
211. Ghourab, E.M., Azab, M.: Software-defined moving-target defense for resilient trustworthy VANETs. TechRxiv (2022). https://doi.org/10.36227/techrxiv.21779921.v1
212. Yang, G., Ge, M., Gao, S., Lu, X., Zhang, L.Y., Doss, R.: A differential privacy mechanism for deceiving cyber attacks in IoT networks. In: Yuan, X., Bai, G., Alcaraz, C., Majumdar, S. (eds.) Network and System Security, Lecture Notes in Computer Science, pp. 406–425. Springer Nature Switzerland, Cham (2022)
213. Xing, J., Yang, M., Zhou, H., Wu, C., Ruan, W.: Hiding and Trapping: a deceptive approach for defending against network reconnaissance with software-defined network. In: 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), pp. 1–8 (2019). https://doi.org/10.1109/IPCCC47392.2019.8958776
214. Lin, H.: SDN-based in-network honeypot: preemptively disrupt and mislead attacks in IoT networks. arXiv preprint arXiv:1905.13254 (2019)
215. Tan, Y., Liu, J., Wang, J.: How to protect key drones in unmanned aerial vehicle networks? An SDN-based topology deception scheme. IEEE Trans. Veh. Technol. **200**, 1–13 (2022). https://doi.org/10.1109/TVT.2022.3200339
216. Kyung, S., et al.: HoneyProxy: design and implementation of next-generation honeynet via SDN, pp. 1–9 (2017). IEEE, https://doi.org/10.1109/cns.2017.8228653
217. Bernieri, G., Conti, M., Pascucci, F.: MimePot: a model-based honeypot for industrial control networks. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp. 433–438 (2019). https://doi.org/10.1109/SMC.2019.8913891
218. Anjum, I., Zhu, M., Polinsky, I., Enck, W., Reiter, M.K., Singh, M.P.: Role-based deception in enterprise networks. In: Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, Virtual Event USA: ACM, pp. 65–76 (2021). https://doi.org/10.1145/3422337.3447824
219. Li, R., Zheng, M., Bai, D., Chen, Z.: SDN based intelligent honeynet network model design and verification. In: 2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE), pp. 59–64 (2021). https://doi.org/10.1109/MLISE54096.2021.00019

220. Kim, J., Nam, J., Lee, S., Yegneswaran, V., Porras, P., Shin, S.: BottleNet: hiding network bottlenecks using SDN-based topology deception. IEEE Trans. Inf. Forensics Secur. **16**, 3138–3153 (2021). https://doi.org/10.1109/TIFS.2021.3075845
221. Shimanaka, T., Masuoka, R., Hay, B., Center, H., Tech, V.: Cyber Deception Architecture: Covert Attack Reconnaissance Using a Safe SDN Approach, p. 10 (2019)
222. Chiang, C.Y.J., et al.: On defensive cyber deception: a case study using SDN. In: MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), pp. 110–115 (2018). https://doi.org/10.1109/MILCOM.2018.8599755
223. Chiang, C.Y.J, Poylisher, A., Chadha, R., Labs, V.: Enhancing Cyber Defense with Autonomous Agents Managing Dynamic Cyber Deception (Position Paper), p. 6 (2017).
224. Gao, C., Wang, Y., Xiong, X., Zhao, W.: MTDCD: an MTD enhanced cyber deception defense system. In: 2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 1412–1417 (2021). https://doi.org/10.1109/IMCEC51613.2021.9482133
225. Achleitner, S., La Porta, T.F., McDaniel, P., Sugrim, S., Krishnamurthy, S.V., Chadha, R.: Deceiving network reconnaissance using SDN-based virtual topologies. IEEE Trans. Netw. Serv. Manag. **14**(4), 1098–1112 (2017). https://doi.org/10.1109/TNSM.2017.2724239
226. Belalis, I., Kavallieratos, G., Gkioulos, V., Spathoulas, G.: Enabling Defensive Deception by Leveraging Software Defined Networks, p. 10 (2020)
227. Ge, M., Cho, J.-H., Kim, D., Dixit, G., Chen, I.-R.: Proactive defense for internet-of-things: moving target defense with cyberdeception. ACM Trans. Internet Technol. **22**, 1–31 (2021). https://doi.org/10.1145/3467021
228. Luo, X., Yan, Q., Wang, M., Huang, W.: Using MTD and SDN-based Honeypots to Defend DDoS Attacks in IoT. In: 2019 Computing, Communications and IoT Applications (ComComAp), pp. 392–395 (2019). https://doi.org/10.1109/ComComAp46287.2019.9018775
229. Islam, M.M., Al-Shaer, E.: Active deception framework: an extensible development environment for adaptive cyber deception. In: 2020 IEEE Secure Development (SecDev), pp. 41–48 (2020). https://doi.org/10.1109/SecDev45635.2020.00023
230. Maleh, Y., Qasmaoui, Y., El Gholami, K., Sadqi, Y., Mounir, S.: A comprehensive survey on SDN security: threats, mitigations, and future directions. J. Reliab. Intell. Environ. (2022). https://doi.org/10.1007/s40860-022-00171-8
231. Cui, Y., et al.: Towards DDoS detection mechanisms in Software-Defined Networking. J. Netw. Comput. Appl. **190**, 103156 (2021). https://doi.org/10.1016/j.jnca.2021.103156
232. Jimenez, M.B., Fernández, D., Rivadeneira, J.E., Rivadeneira, J.E., Bellido, L., Cardenas, A.: A survey of the main security issues and solutions for the SDN architecture. IEEE Access (2021). https://doi.org/10.1109/access.2021.3109564
233. Yurekten, O., Demirci, M.: SDN-based cyber defense: a survey. Future Gener. Comput. Syst. **115**, 126–149 (2021). https://doi.org/10.1016/j.future.2020.09.006
234. Dantas Silva, F.S., Silva, E., Neto, E.P., Lemos, M., Venancio Neto, A.J., Esposito, F.: A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios. Sensors (2020). https://doi.org/10.3390/s20113078
235. Babiker Mohamed, M., Matthew Alofe, O., Ajmal Azad, M., Singh Lallie, H., Fatema, K., Sharif, T.: A comprehensive survey on secure software-defined network for the Internet of Things (2021). https://doi.org/10.1002/ett.4391
236. Bawany, N.Z., Shamsi, J.A., Salah, K.: DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arab. J. Sci. Eng. **42**(2), 425–441 (2017). https://doi.org/10.1007/s13369-017-2414-5
237. Beslin Pajila, P.J., Golden Julie, E.: Detection of DDoS attack using SDN in IoT: a survey. In: Balaji, S., Rocha, Á., Chung, Y.-N. (eds.) Intelligent Communication Technologies and Virtual Mobile Networks, Lecture Notes on Data Engineering and Communications Technologies, pp. 438–452. Springer International Publishing, Cham (2020)
238. Rawat, D.B., Reddy, S.R.: Software defined networking architecture, security and energy efficiency: a survey. IEEE Commun. Surv. Tutor. **19**(1), 325–346 (2017). https://doi.org/10.1109/COMST.2016.2618874
239. Taipalus, T.: Systematic mapping study in information systems research. J. Midwest Assoc. Inf. Syst. JMWAIS (2023). https://doi.org/10.17705/3jmwa.000079
240. IEEE. IEEE Xplore (2023). https://ieeexplore.ieee.org/Xplore/home.jsp. Accessed 18 Sep 2023
241. Google. Google Scholar (2023). https://scholar.google.com/. Accessed 18 Sep 2023
242. RefSeek. RefSeek—Academic Search Engine (2023). https://www.refseek.com/. Accessed 18 Sep 2023
243. Cheswick, W.R., Bellovin, S.M., Rubin, A.D.: Firewalls and Internet Security (Repelling the Willy Hacker), 2nd edn. Pearson Education Inc., USA (2003)
244. Abie, H.: An Overview of Firewall Technologies (2000). Available: https://www.researchgate.net/publication/2371491_An_Overview_of_Firewall_Technologies. Accessed 16 May 16 2022
245. Alsaqour, R., Motmi, A., Abdelhaq, M.: A systematic study of network firewall and its implementation. Int. J. Comput. Sci. Netw. Secur. **21**(4), 199–208 (2021). https://doi.org/10.22937/IJCSNS.2021.21.4.24
246. Kaplesh, P., Goel, A.: Firewalls: A Study on Techniques, Security and Threats, p. 12 (2019)
247. Katwal, G., Sood, M.: A comparative study of traditional network firewalls & SDN firewalls. Int. J. Latest Trends Eng. Technol. (2016). Available: https://www.ijltet.org/journal/146865056311.pdf. Accessed 16 May 2022
248. Satasiya, D., Rupal, and R.D.: Analysis of software defined network firewall (SDF). In: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), pp. 228–231 (2016). https://doi.org/10.1109/WiSPNET.2016.7566125
249. Oscarson, P.: 'Information Security Fundamentals. In: Irvine, C., Armstrong, H. (eds.) Security Education and Critical Infrastructures, IFIP Advances in Information and Communication Technology, pp. 95–107. Springer US, New York (2003)
250. Wu, J., Bi, J., Li, X., Ren, G., Williams, M., Xu, K.: a source address validation architecture (SAVA) testbed and deployment experience. Internet Engineering Task Force, Request for Comments RFC 5210 (2008). https://doi.org/10.17487/RFC5210
251. Wu, J., Bi, J., Bagnulo, M., Baker, F., Vogt, C.: Source address validation improvement (SAVI) framework. Internet Engineering Task Force, Request for Comments RFC 7039 (2013). https://doi.org/10.17487/RFC7039
252. Li, Y., Li, D., Cui, W., Zhang, R.: Research based on OSI model. In: 2011 IEEE 3rd International Conference on Communication Software and Networks, pp. 554–557 (2011). https://doi.org/10.1109/ICCSN.2011.6014631
253. INCIBE. Security in the GOOSE protocol. In: INCIBE-CERT (2020). https://www.incibe-cert.es/en/blog/security-goose-protocol. Accessed 24 Dec 2022
254. Liu, W., Ren, P., Liu, K., Duan, H.: Behavior-based malware analysis and detection. In: 2011 First International Workshop on Complexity and Data Mining, pp. 39–42 (2011). https://doi.org/10.1109/IWCDM.2011.17
255. Burguera, I., Zurutuza, U., Nadjm-Tehrani, S.: Crowdroid: behavior-based malware detection system for Android. In Proceedings of the 1st ACM workshop on Security and privacy in

smartphones and mobile devices, in SPSM '11, pp. 15–26 (2011). Association for Computing Machinery, New York. https://doi.org/10.1145/2046614.2046619
256. Sethi, K., Kumar, R., Sethi, L., Bera, P., Patra, P.K.: A novel machine learning based malware detection and classification framework. In: 2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security), pp. 1–4 (2019). https://doi.org/10.1109/CyberSecPODS.2019.8885196
257. Singh, J., Singh, J.: A survey on machine learning-based malware detection in executable files. J. Syst. Archit. **112**, 101861 (2021). https://doi.org/10.1016/j.sysarc.2020.101861
258. Santos, I., Devesa, J., Brezo, F., Nieves, J., Bringas, P.G.: OPEM: a static-dynamic approach for machine-learning-based malware detection. In: Herrero, Á., Snášel, V., Abraham, A., Zelinka, I., Baruque, B., Quintián, H., Calvo, J.L., Sedano, J., Corchado, E. (eds.) International Joint Conference CISIS'12-ICEUTE´12-SOCO´12 Special Sessions, Advances in Intelligent Systems and Computing, pp. 271–280 (2013). Springer, Berlin. https://doi.org/10.1007/978-3-642-33018-6_28
259. Bazrafshan, Z., Hashemi, H., Fard, S.M.H., Hamzeh, A.: A survey on heuristic malware detection techniques. In: The 5th Conference on Information and Knowledge Technology, pp. 113–120 (2013). https://doi.org/10.1109/IKT.2013.6620049
260. Rehman, Z.-U., et al.: Machine learning-assisted signature and heuristic-based detection of malwares in Android devices. Comput. Electr. Eng. **69**, 828–841 (2018). https://doi.org/10.1016/j.compeleceng.2017.11.028
261. Khodamoradi, P., Fazlali, M., Mardukhi, F., Nosrati, M.: Heuristic metamorphic malware detection based on statistics of assembly instructions using classification algorithms. In: 2015 18th CSI International Symposium on Computer Architecture and Digital Systems (CADS), pp. 1–6 (2015). https://doi.org/10.1109/CADS.2015.7377792
262. Masdari, M., Khezri, H.: A survey and taxonomy of the fuzzy signature-based intrusion detection systems. Appl. Soft Comput. **92**, 106301 (2020). https://doi.org/10.1016/j.asoc.2020.106301
263. Ioulianou, P., Vasilakis, V., Moscholios, I., Logothetis, M.: A signature-based intrusion detection system for the internet of things. In Information and Communication Technology Form, AUT: York (2018). Available: https://eprints.whiterose.ac.uk/133312/. Accessed 2 Nov 2022
264. Kumar, V., Sangwan, D.O.P.: Signature based intrusion detection system using SNORT. Int. J. Comput. Appl. **1**, 35–41 (2012)
265. Wang, C., Lu, Z.: Cyber deception: overview and the road ahead. IEEE Secur. Priv. **16**(2), 80–85 (2018). https://doi.org/10.1109/MSP.2018.1870866
266. Lu, Z., Wang, C., Zhao, and S.: Cyber deception for computer and network security: survey and challenges. arXiv http://arxiv.org/abs/2007.14497 (2020). Accessed 19 Nov 2022
267. Kyung, S., et al.: HoneyProxy: design and implementation of next-generation honeynet via SDN. In: 2017 IEEE Conference on Communications and Network Security (CNS), pp. 1–9 (2017). https://doi.org/10.1109/CNS.2017.8228653
268. Fu, X., Yu, W., Cheng, D., Tan, X., Streff, K., Graham, S.: On recognizing virtual honeypots and countermeasures. In: 2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, pp. 211–218 (2006). https://doi.org/10.1109/DASC.2006.36
269. Zamiri-Gourabi, M.-R., Qalaei, A.R., Azad, B.A.: Gas what? I can see your GasPots. Studying the fingerprintability of ICS honeypots in the wild. In: Proceedings of the Fifth Annual Industrial Control System Security (ICSS) Workshop, in ICSS, pp. 30–37. Association for Computing Machinery New York (2019). https://doi.org/10.1145/3372318.3372322
270. Holz, T., Raynal, F.: Detecting honeypots and other suspicious environments. In: Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop, pp. 29–36 (2005). https://doi.org/10.1109/IAW.2005.1495930
271. Ericsson. 6G—Follow the journey to the next generation (2023). https://www.ericsson.com/en/6g. Accessed 4 May 2023
272. Abdulqadder, I.H., Zhou, S.: SliceBlock: context-aware authentication handover and secure network slicing using DAG-blockchain in edge-assisted SDN/NFV-6G environment. IEEE Internet Things J. **9**(18), 18079–18097 (2022). https://doi.org/10.1109/JIOT.2022.3161838
273. Abdel Hakeem, S.A., Hussein, H.H., Kim, H.: Security requirements and challenges of 6G technologies and applications. Sensors **22**, 1969 (2022). https://doi.org/10.3390/s22051969
274. Singh, S., Mehla, V., Nikolovski, S.: LSSDNF: a lightweight secure software defined network framework for future internet in 5G–6G. Future Internet (2022). https://doi.org/10.3390/fi14120369
275. Chiti, F., Degl'Innocenti, A., Pierucci, L.: Secure networking with software-defined reconfigurable intelligent surfaces. Sensors **23**, 2726 (2023). https://doi.org/10.3390/s23052726
276. Paloalto. What is Microsegmentation?. Palo Alto Networks (2023). https://www.paloaltonetworks.com/cyberpedia/what-is-microsegmentation. Accessed 21 June 2023
277. Walker, G.: SDN-based Micro-Segmentation for OT networks. Allied Telesis (2023). https://www.alliedtelesis.com/mt/en/blog/sdn-based-micro-segmentation-ot-networks. Accessed 21 June 2023
278. Beshley, M., Klymash, M., Scherm, I., Beshley, H., Shkoropad, Y.: Emerging network technologies for digital transformation: 5G/6G, IoT, SDN/IBN, cloud computing, and blockchain. In: Klymash, M., Luntovskyy, A., Beshley, M., Melnyk, I., Schill, A. (eds.) Emerging Networking in the Digital Transformation Age, Lecture Notes in Electrical Engineering, pp. 1–20. Springer Nature Switzerland, Cham (2023)
279. Beshley, M., Pryslupskyi, A., Panchenko, O., Beshley, and H.: SDN/cloud solutions for intent-based networking. In: 2019 3rd International Conference on Advanced Information and Communications Technologies (AICT), pp. 22–25 (2019). https://doi.org/10.1109/AIACT.2019.8847731
280. Wei, Y., Peng, M., Liu, Y.: Intent-based networks for 6G: insights and challenges. Digit. Commun. Netw. **6**(3), 270–280 (2020). https://doi.org/10.1016/j.dcan.2020.07.001
281. Suarez, T., Rowan, B.: D2.5: NGIoT Roadmap and Policy Recommendations, ICT-56-2020 (2022). Available: https://www.ngiot.eu/wp-content/uploads/sites/73/2022/05/EU-IoT_D2.5_NGIoT-Roadmap-v01.0.pdf
282. EU (2022) A Roadmap for the Next-Generation IoT in Europe | Shaping Europe's digital future. https://digital-strategy.ec.europa.eu/en/news/roadmap-next-generation-iot-europe. Accessed 7 May 2023

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.