symmetry

MDPI

Article

Machine Learning-Based Botnet Detection in Software-Defined Network: A Systematic Review

Khlood Shinan [1,*], Khalid Alsubhi [1,†], Ahmed Alzahrani [1,†] and Muhammad Usman Ashraf [2,†]

[1] Department of Computer Science Department, Faculty of Computing and Information Technology, King Abdulaziz University (KAU), Jeddah 21589, Saudi Arabia; kalsubhi@kau.edu.sa (K.A.); asalzahrani@kau.edu.sa (A.A.)
[2] Department of Computer Science, University of Management and Technology Sialkot, Lahore 54770, Pakistan; usman.ashraf@skt.umt.edu.pk
* Correspondence: khlood_alshehri@hotmail.com
† These authors contributed equally to this work.

Abstract: In recent decades, the internet has grown and changed the world tremendously, and this, in turn, has brought about many cyberattacks. Cybersecurity represents one of the most serious threats to society, and it costs millions of dollars each year. The most significant question remains: Where do these attacks come from? The answer is that botnets provide platforms for cyberattacks. For many organizations, a botnet-assisted attack is a terrifying threat that can cause financial losses and leave global victims in its wake. It is therefore imperative to defend organizations against botnet-assisted attacks. Software defined networking (SDN) has emerged as one of the most promising paradigms for this because it allows exponential increases in the complexity of network management and configuration. SDN has a substantial advantage over traditional approaches with regard to network management because it separates the control plane from network equipment. However, security challenges continue to arise, which raises the need for different types of implementation strategies to spread attack vectors, despite the significant benefits. The main objective of this survey is to assess botnet detection techniques by using systematic reviews and meta-analyses (PRISMA) guidelines. We evaluated various articles published since 2006 in the field of botnet detection, based on machine learning, and from 2015 in the field of SDN. Specifically, we used top-rated journals that featured the highest impact factors. In this paper, we aim to elaborate on several research areas regarding botnet attacks, detection techniques, machine learning, and SDN. We also address current research challenges and propose directions for future research.

Keywords: botnet; machine Learning; network security; SDN

Citation: Shinan, K.; Alsubhi, K.; Alzahrani, A.; Ashraf, M.U. Machine Learning-Based Botnet Detection in Software-Defined Network: A Systematic Review. *Symmetry* **2021**, *13*, 866. https://doi.org/10.3390/sym13050866

Academic Editor: José Carlos R. Alcantud

Received: 25 March 2021
Accepted: 7 May 2021
Published: 12 May 2021

1. Introduction

A network is a collection of devices that are connected over the Internet, and the total number of these devices increases every day. Undoubtedly, networks of financial and business institutions are continuously under security risk, which not only costs billions of dollars in damages and recovery, but also has a negative effect on their reputation. The increasing number of users affected by malicious software is becoming a critical problem. Botnets have become the main concern, since they are one of the biggest threats to security systems. Their popularity comes from their ability to control corporate mainframes by infiltrating any Internet-connected device that uses a digital video recorder (DVR) [1].

A botnet can be defined as a network of compromised host devices that are used to carry out malicious activities. Desktop computers, smartphones, notebooks, and tablets are examples of such host devices. A botnet consists of three components: an attacker called a botmaster, a command and control (C&C) server, and an infected machine called a bot. The botmaster requires a C&C channel to command the bots and coordinate malicious attacks. Examples of C&C channels are IRC, HTTP, and P2P. According to communication protocols, C&C channels

can be centralized or decentralized. Bots are used to send distributed denial of service (DDoS) attacks, phishing attacks, spam emails, and other forms of malicious attacks [1].

The WannaCry is one of the best-known examples of a ransomware attack. It harmed companies throughout the world in the spring of 2017. It is a type of malicious software, also known as malware, which is used by cybercriminals to extort money from the companies they target. During that time, they attacked over 200,000 computers in more than 150 countries. This included the United Kingdom National Health Service. All in all, it cost the UK £92 million and ran up global costs to a whopping £6 billion [2].

Over the last decade, botnet being a major persistent threat on the Internet has received greater research focus and a lot of attention, as evidenced by review papers and multiple surveys focusing on Botnet detection techniques. The survey done in 2015 [3], provides a comparison of existing research. it is the first survey present prevent and defense methods on Botnet. There are only three survey papers on Botnet Detection based on DNS traffic analysis presented by [4] in 2015, [5] in 2017, and [6] in 2019. They summarized the existing techniques in botnet detection methodologies that are based on DNS traffic analysis and techniques for mitigating the threat of botnets. While the survey in 2020 [7], focuses on the numerous approaches and compares the different methods used in the recent past in general botnet detection and IoT-Bot's detection. However, the survey in 2020 [8] discussed the concepts involved in defending a DDoS attack from traditional methods to IoT-specific ones.

Only one survey in botnet detection in Software Defined Networking (SDN) has been published in 2018 [9], and it is neither comprehensive nor takes into consideration various parameters vital for effective comparison. It even has a few papers as references that are not enough a comprehensive look.

This paper provides an up-to-date taxonomy, together with a review of the significant research works to analyze the current implementation of machine learning in botnet detection, both in traditional networks and SDN up to the present time, and classification of the proposed systems according to the taxonomy. To the best of our knowledge, this is the first survey to discuss Machine learning-based botnet detection techniques in SDN in which the problems, existing solutions, and the future research direction in the future are explored and clarified.

The main contributions are as follows:

- Describe basic features about botnet techniques.
- Analyze existing studies relating to the deployment of machine learning in botnet detection and summarize their findings.
- Analyze existing studies relating to the deployment of machine learning in botnet detection in SDN and summarize their findings.
- Identify new challenges and issues that have arisen as a result of existing solutions and suggest directions for future research.

The rest of this paper is organized as follows: The introduction and background to the botnet is briefly discussed in Section 1. The methods used in this review is presented in Section 2. Section 3 discusses the research finding and discussion and it is subdivided into four areas—botnet detection approaches; botnet detection based on machine Learning; botnet detection in SDN; and botnet detection in SDN based on machine Learning. In addition, Section 4 discusses the challenges and directions for future research. Section 5 conclude the paper.

1.1. Botnet Elements

It is very important to understand the basic elements of a botnet. A botnet has three main elements: bots, botmaster and command and control channel (C&C) [10].

1.1.1. Bot

A bot is a software program (malware) installed on a compromised host, and it can perform a series of activities, usually malicious. There are many ways to install a bot on

victims' machines, including when they access infected sites or viral mechanisms. Notably, a bots is not a vulnerability in applications or operating systems (OS). Rather, it is a program that is propagated by worms or used to install back doors on compromised machines. Bots are usually configured and initialized each time the infected device is booted by the victim. Actions are launched from particular commands sent through the C&C channel from the botmaster. The existence of the C&C channel is the main difference between a bot and other types of malware [11,12].

1.1.2. Botnet

A botnet is a group of infected hosts running bots and connected to a C&C channel waiting for instructions to execute malicious activities [11,12].

1.1.3. Botmaster

The botmaster controls the botnet from a remote location by sending instructions to the bots to conduct illegal acts, such as gaining financial benefits "by renting the network to send spam to other users". Botmasters try to compromise vulnerable computers (computers with weaker defense mechanism) by using a propagation mechanism. After they are infected, these machines become "slaves" or "zombies," and they are used to attack vulnerable hosts or conduct denial-of-service attacks (DoS) [11,12].

1.1.4. Command-and-Control

C&C infrastructure is the most critical element of a botnet. It is a control authority that can be either centralized or decentralized. One or more communication protocols are used by the botmaster to command slave hosts and coordinate their activities. Typically, the C&C infrastructure acts as the primary way to manage the bots in the botnet, and it is important for them to maintain a stable and secure connection in this infrastructure to work efficiently. Then, the architecture of the C&C infrastructure determines the reliability, robustness, and response time of a botnet. Generally, botnets are categorized as centralized or decentralized botnets [11,12].

1.2. Botnet Architecture

Botnets are categorized into three different structures according to the C&C channel: centralized architecture, decentralized architecture, and hybrid architecture [1,11–13]. According to the communication protocols used by botnets, botnets can be classified into several protocols discuss in Table 1.

Table 1. Botnet protocols.

Protocols	Def	Advantages	Examples
IRC	IRC is a protocol of real-time internet text messaging chat; Mainly used in centralized architecture.	1. Low-latency communication. 2. Simple commands. 3. Private (one-to-one) communication. 4. Capable of group (many to-many) communication. 5. simple to set up. 6. Flexibility in communication. 7. Anonymous real-time communication	Agobot, SDBot, Spybot, and GT Bot
HTTP	HTTP protocols attempt to blend botnet traffic into regular HTTP traffic. Mainly used in centralized architecture.	Difficult to detect and easily bypasses firewalls.	Bobax, ClickBot, Rustock and Blackenergy.
P2P	P2P is a communication protocol which is mainly used in decentralized architecture	hard to detect, very high resilience.	Slapper, Sinit, Phatbot, Nugache, Storm.

1.2.1. Centralized Architecture

In a centralized botnet architecture, the botmaster controls all the bots from a central hub known as a command and control server. In this structure, a single point (the C&C server) is used to exchange instructions between the botmaster and the bots. The major benefit of this architecture is that it provides reliable coordination of the bots for their botmaster. Moreover, it makes status monitoring easy for the botmaster, and it speeds up reaction time. In contrast, once the C&C server is identified, it is very easy for a defender to take down this type of botnet. The two protocols most often used in a centralized architecture are Internet relay chat (IRC) and hypertext transfer protocol (HTTP). A centralized architecture can suffer a single point of failure, because of a denial-of-service attack, and the botmaster is no longer able to communicate with the bots when an IRC or HTTP server is taken down [1,11–15]. Figure 1 shows a centralized model.

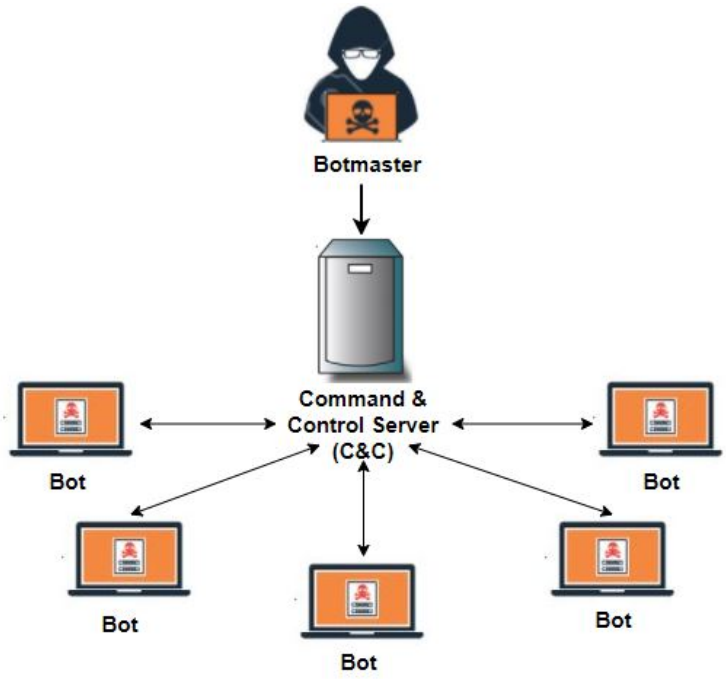

Figure 1. Centralized architecture.

- Botnet based on IRC: The internet relay chat (IRC) is a protocol of real-time internet text messaging or synchronous conferencing. This protocol is the most popular botnet C&C channel. The reasons for this popularity are: the botmaster has real-time communication with the bots, it supports a large number of clients, it works with different network topologies, it is an open-source and it has expandable design. Spybot, Agobot, SDBot, and GT Bot are the most famous IRC-based botnets.
- Botnet based on HTTP: The hypertext transfer protocol (HTTP) is another common protocol used by C&C servers. HTTP communication is widely used in many web-based applications. These web-based C&C bots attempt to blend into regular HTTP traffic, making them hard to detect. They can easily evade intrusion detection systems (IDSs) and bypass firewalls with port-based filtering techniques. Bobax, ClickBot, Rustock, and the most popular, Blackenergy, are well-known bots that use the HTTP protocol.

1.2.2. Decentralized Architecture

A decentralized form allows the bots to act autonomously. In this structure, the communication system is not based on servers for destroying and discovering a number of bots, and there is no centralized point for communication. In this kind of botnet, each bot can establish connections with other bots, and bots behave as both servers and clients.

Peer to Peer (P2P) is the most protocol used in a centralized architecture, Figure 2 shows a decentralized model [1,11–13].

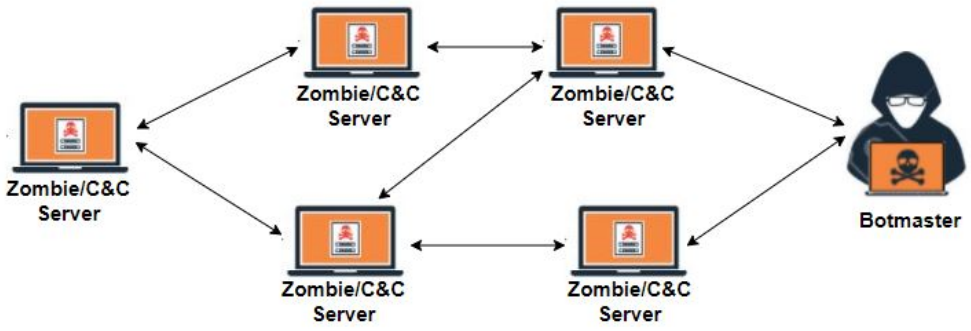

Figure 2. Decentralized architecture.

- Botnet based on Peer to Peer (P2P): This communication protocol is mainly used in a decentralized architecture.The goal of P2P botnets is to remove the failure point, which is the main vulnerability of a centralized structure. Therefore, detecting this type of botnet is very difficult. To send commands to all bots over the entire network, the botmaster needs to connect to only one of the bots (peers) [1,14,15].

1.2.3. Hybrid Architecture

The combination of centralized and decentralized architectures is a hybrid architecture. A hybrid botnet is divided into two types of bots: one is a servant and the other is a client bot. The servant bot receives the botmaster's commands and forwards them to the client bot. It is more difficult to detect and control botnets in a network with hybrid architecture than with centralized or decentralized architectures, yet the design of the botnet is not very complex [1,12,13].

1.3. Botnet Life Cycle

Learning the life cycle of botnets is an important factor in the successful analysis of botnet detection systems. Understanding each phase of this cycle can help to improve and develop an efficient botnet detection system. The host must go through five phases to become an active bot and part of a botnet as described in Refs, Figure 3 shows the life cycle of a botnet [11,13,16–18].

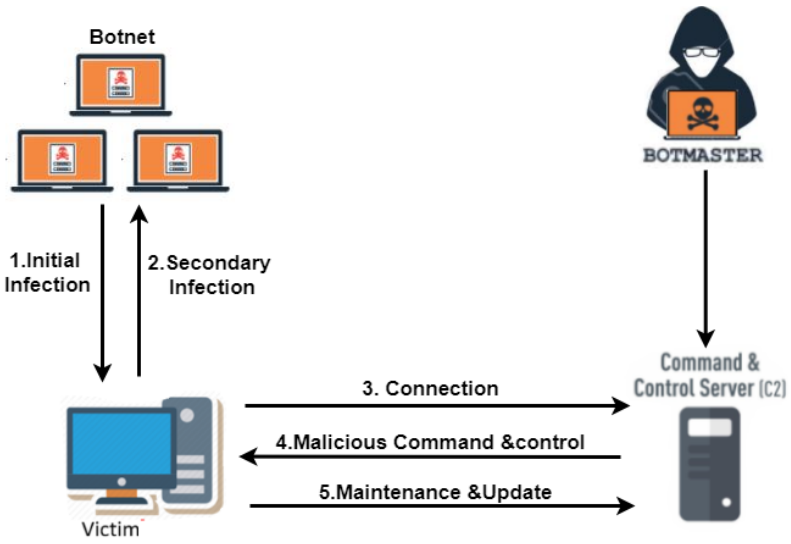

Figure 3. Botnet Life cycle.

The first phase is called the initial injection phase, where a host is compromised and becomes a potential bot. A host can be infected with different techniques like executing malicious software, downloading malware from an unauthorized website, clicking an email attachment, or using an infected removable disk such as thumb drive or flash drive. In the second phase, also called the secondary injection phase, the compromised host executes a program in a specified network database that searches for malware binaries. Once the bot program is installed, the infected host becomes as a real bot (or Zombie) and runs the malicious code. Bot binaries are usually downloaded using FTP or HTTP protocols. The third phase is also called the connection or rallying phase, where a C&C channel is installed by the bot program and then starts to communicate between the bot and the C&C server. Once it is connected, the bot can receive and respond to the commands from the botmaster, and the infected machine becomes part of the botnet army of the attacker. This phase occurs many times in the bot life cycle [11,13,16–18].

The fourth phase is also called the malicious phase, where the real command and control operations of the botnet are started. The bot attempts to execute a series of malicious operations depending on the botmaster's instructions. These include spamming, identity fraud, information leakage, distributed denial of services (DDoS), and click fraud. The final phase is also called the maintenance and upgrading phase, where updated bots are used to keep the other bots under the botmaster's control. The bots need this phase for several reasons. They may need to evade detection techniques by updating the bot binary, or they may want to add new features to their bot army [11,13,16–18].

1.4. Botnet Threats

A botnet is more harmful than conventional threats such as viruses and worms. The Honeynet project presented different types of botnet attacks, including click fraud, distributed denial-of-service (DDoS), spam, cyber warfare, exploiting resources, and stealing confidential information [19]. Other studies have shown that botnets can be manipulated to conduct a wide variety of illegitimate activities and several kinds of cybercrimes.

1.4.1. Distributed Denial of Service

Botnets can launch distributed denial of service (DDOS) attacks, in which the incoming traffic from an enormous number of sources floods the victim's system. The huge number of participants in a botnet gives the DDoS significant destructive power. This allows botmasters to use the botnet to take down the victim's control system by commanding bot members to send massive numbers of requests to the victim's system. Online gaming and gambling websites are examples of this attack [12,13,17]. Globally, the number of attacks has been grown up from 100 Gbps to 400 Gbps between 2018 and 2019, and the total number of DDoS attacks expected to double from 7.9 million in 2018 to 15.4 million by 2023 [20].

1.4.2. Spam

Spam is unwanted email messages that often contain malicious links or advertisements that are sent to a huge number of users. For an attacker, a botnet is the safest option to use as a platform for sending spam emails. The "Grum" botnet has sent approximately 40 billion malignant e-mails, using the 600,000 bots in its network. This attack began by sending botmaster commands to bots until they started sending spam emails to the address of the victim [13,17].

1.4.3. Stealing Information

A botmaster can command bots to obtain secret data from compromised hosts by using methods such as screen capture, reading log files, and keylogging. SDBot is an example of a botnet that employs advanced key-logging software to collect personal information, which can then be sold to others to perform illegitimate actions. Keylogging methods are the main tools used by Zeus Bots to steal private bank accounts and credit card information. This

permits the botmaster to extract usernames and passwords from a social network account, bank's website, and emails. Furthermore, the bot can retrieve private user information from the Windows application program interface (API) before it is encrypted by the web browser [12,13].

1.4.4. Exploiting Resources

Compromised hosts are recruited to execute illegal actions. For instance, bots were used on Twitter and Facebook to cast fake votes and to raise the number of followers. Moreover, a bot may use the victim's machine to access a website on a regular visit to increase the number of website users without permission from them.

2. Methods

The Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) method that provides an evidence-based minimum set of items for studies of a specific topic. For this review, it identified studies in the databases of multiple libraries then narrowed down the results, as shown in Figure 4. PRISMA comprises three phases: identification, scanning, and eligibility testing. In the identification phase, it recognized and gathered approximately 120 articles using the Google Scholar search engine. During the scanning phase, it removed duplicate and non-conforming papers, leaving about 105 papers in this research. During eligibility testing, whereby all the remaining articles were filtered to remove articles that were not about botnet-related detection. The result was approximately 96 articles for full review.

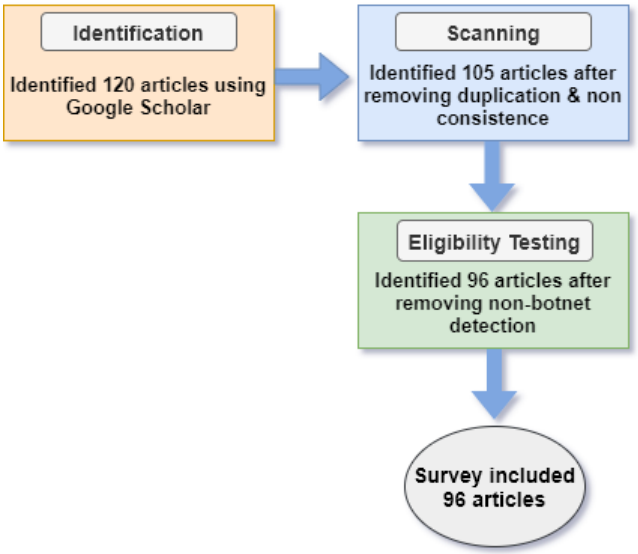

Figure 4. PRISMA method diagram, N represent the number of articles.

2.1. Search Strategy

This survey research used techniques based on keyword combinations and selecting data sources. The keyword combination used: (a) "botnet detection approaches"; (b) "botnet detection based on machine Learning"; (c) "botnet detection techniques in SDN"; and (d) "botnet detection based on machine learning in SDN". To select data sources, we searched for relevant publications in major research journals, including IEEE Xplore, MDPI, ACM Digital Library, ScienceDirect, SpringerLink, Google Scholar, Hindawi, and other computer engineering journals. After removing duplicate and irrelevant articles, the search yielded 96 articles. The numbers of articles extracted by search item and by year are shown in Figures 5 and 6, respectively.

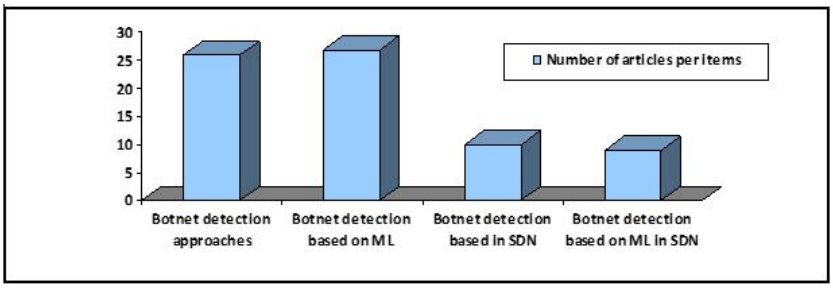

Figure 5. Number of articles by items.

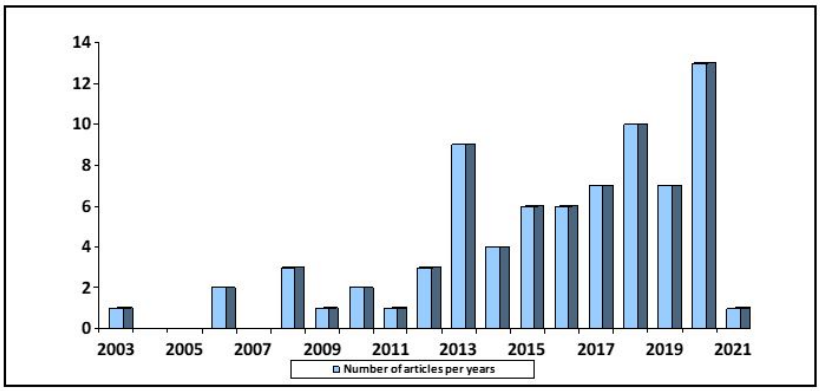

Figure 6. Number of articles per year.

2.2. Data Extraction

The data categories were collected from articles following:

a. Research contribution.
b. Detection techniques.
c. Botnet protocol.
d. Method.
e. Dataset.
f. Accuracy and other metrics.

3. Research Findings and Discussion

The topic of botnet detection, machine learning and SDN have individually received significant attention and interest both from academia and industry. Despite the fact that all technologies mentioned are still in their infancy phases, and have only existed for a few years, to the best of the author's knowledge, this paper is the first systematic analysis of previous efforts to apply machine learning to the SDN architecture for the purpose of botnet detection. The expected contributions of this work is to answer the research questions mentioned in Table 2 and to address the findings, based on observations made after reviewing the published articles. For answering RQs, the survey is divided into four areas: studies of approaches to botnet detection, studies of botnet detection based on machine learning, studies of botnet detection using a Software-Defined Network (SDN), and studies of botnet detection based on machine learning in a Software-Defined Network (SDN).

Table 2. Research questions.

Research Questions (RQs)	Discussion
RQ1: What are the latest studies of botnet detection approaches that have been implemented?	Discuss the approaches to detect botnets recently proposed in the literature and their limitations.
RQ2: What are the latest studies of botnet detection based on machine learning?	Present an overview of machine learning and summarize the studies in detecting botnets in network traffic that use machine learning techniques.
RQ3: What are the latest studies of botnet detection in Software-defined Networks (SDNs)?	Present an overview of SDN characteristics and summarize studies of botnet detection that use it.
RQ4: What are the latest studies of botnet detection based on machine learning (ML) in SDNs?	Discuss recent studies that detected botnets based on machine learning in SDN and their results.

3.1. RQ1: What Are the Latest Studies of Botnet Detection Approaches That Have Been Implemented?

There has been growing interest in approaches to bot detection and prevention. Identifying infected machines before they are exploited to launch malicious actions is the most important requirement in botnet detection. Various approaches to detect botnets have been proposed in the literature. Refs. [21,22] discussed two approaches that can be classified into honeynets and intrusion detection systems (IDS). Then, IDSs can be subdivided into two groups: signature-based systems and anomaly-based systems. Anomaly-based detection techniques can be further classified as host-based or network-based methods. Other studies [11,13,16,18] have divided botnet detection techniques into four general categories: honeynets, intrusion detection systems (IDS), data mining techniques and DNS-based techniques. However, botnet detection techniques were categorized by the authors in [1,15,23–25] based on the type of machine learning: supervised and unsupervised botnet detection. Figure 7 displays classification of botnet detection techniques.

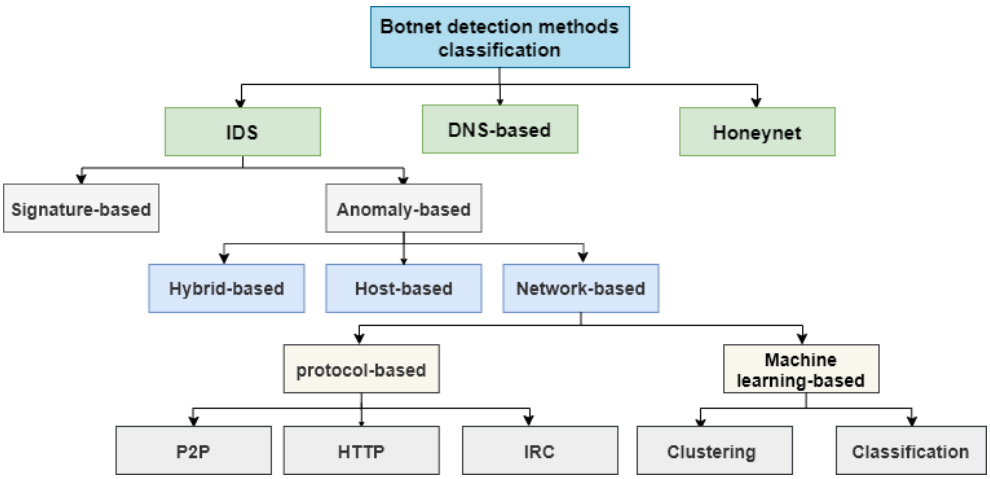

Figure 7. Botnet detection techniques classification.

3.1.1. Honeynet-Based Detection

A honeypot is a network set up with intentional vulnerabilities, it is used as a trap that aims to attract the attention of attackers to target this computer device without exposing the real network or its data to adversaries [13]. For more analysis, the honeypot uses the data from the infected host to measure the intensity of the attack and the characteristics of the botnet. The information extracted from the bots is also used to detect the attacker's

tools, C&C system, and motivations [13,22,26]. However, honeypots still have many challenges and problems. Their scalability is limited, since they demand intensive hardware equipment, and are often unable to provide definitive data [13,22]. Moreover, since the discovery of the infected system is one of the honeypot's plans, placing it as a trap is also challenging since attackers can take over honeypots to destroy other machines or systems outside the honeynet. Also, bots can evade honeypot detection by launching honeypot-aware attacks. Therefore, honeypots alone are not sufficient to identify botnets [22].

3.1.2. DNS-Based Detection

This method depends on the property of the botnet DNS query. It detects the C&C server bots by executing DNS queries to locate the C&C server, which is typically hosted by a Dynamic DNS (DDNS) provider. DNS is the most popular and easiest approach to detect the botnet. There are two main differences between legitimate DNS queries and botnet DNS queries. A legitimate query occurs continuously, whereas the query occurs simultaneously in the botnet DNS. In addition, only botnet members query the domain name of the C&C server. Thus, the number of IP addresses queried by the C&C domain is fixed. So, it is much easier to detect botnet DNS traffic by simply observing DNS activities and identifying any unexpected or unusual DNS querying [4,13].

The authors in [27] contributed to the discovery of the first botnet detection mechanism on DNS network-level behavior. It used a new technique specifically for Dynamic (DDNS) names that indicated unreasonably high or temporally concentrated query rates. Additional research was proposed by [18] to discuss botnet detection based on the monitoring of DNS traffic. The conclusion of that paper stipulated that DNS queries from bots can be effectively grouped by their points of similarity in the DNS request. Thereafter, they are used to detect bot activities.

However, The botnet can successfully avoid the DNS detection technique by using fast-flux. A botmaster DNS utilizes different IP addresses to avoid detection of the botnet servers' location. The distinguishing feature of botnets is the fast changes of IP addresses to domain names, that in turn, prevent host detection.

3.1.3. Intrusion Detection System (IDS)

An IDS is hardware or a software application used for monitoring system services. It allows the user to detect policy violations or malicious activities. These events and violations are then reported to the management site [28]. AIDS has several advantages, First, it has the ability to detect internal malicious activity. It raises an alarm if an attacker begins making transactions in a compromised account that are unidentified in normal user behavior. Second, since the system is built from personalized profiles, it is extremely difficult for a cybercriminal to know what normal user behavior is without triggering an alarm. Recently, There are quite a lot notable published articles that promise IDS represents another important area for cybersecurity [29]. In 2019 [30], proposed a novel statistical analysis and deep learning approach to provide intelligent intrusion detection (IDS) system. It introduces a deep autoencoder–based long short-term memory (LSTM) with statistical data analysis for extracting robust, optimal, and highly correlated characteristics. the authors validated the proposed IDS using the benchmark NSL-KDD database. Their experimental results show that the designed IDS achieves better efficiency than deep and conventional shallow networks. In 2020 [31] suggested a novel algorithm for a network intrusion detection system (NIDS) using an enhanced feature subset selected directly using a Genetic Algorithm (GA), Fuzzy C-mean clustering, and CNN extractor method. The results of this study might have been more interesting and achieved the highest accuracy of 98.2%. furthermore, The paper's main contribution In [32] proposed a novel approach to detecting adversarial attacks on artificial neural networks in the context of intrusion detection systems. Another recent survey by [33] conducted a comprehensive survey of intrusion detection approaches in the internet of things (IoT). The main observation of the authors is that classified IDSs based on the detection technique, IDS placement

technique, and security threat. Ascertaining IDS for botnet detection is currently an active research area, and it can be categorized into two main groups: signature-based systems and anomaly-based systems [12,13,16,18,21,22,33].

1. Signature-based IDS

This IDS (sometimes called a knowledge-based IDS) relies on detecting pre-computed hashes of existing malware binaries. It can be extremely powerful in monitoring inbound network traffic, and it can also process a high volume of network traffic efficiently. It is the preferred method for some industries, as it has a low false-positive rate. A signature-based IDS can be used as an agent, working on a gateway or an end-host to further examine binaries in transfer on-the-fly. Despite the fact that signature-based IDS relies on its database of known threats, which is in itself a clear advantage. Signature-based IDSs need frequent updates, by which they can be easily violated or modified by polymorphous attacks, zero-day attacks, and similar approaches. Thus, signature-based IDSs are not sufficient to detect botnest.

2. Anomaly-based IDS

These are widely used for botnet detection because they can detect unknown botnets. They flag the issue of identifying instances in a dataset that do not follow normal behavior. First they create a baseline of normal behavior for the protected system, and then they model a decision engine. This decision engine can determine any divergence or statistical deviation from the norm, and it can also alert thereon as a threat. Research has shown that anomaly-based IDS methods can also detect botnet based on high volumes of traffic, the number of network traffic anomalies, high network latency (traffic on unusual ports), and other unusual system behavior. The main disadvantages of anomaly detection are high levels of false alarms and the limitations of their training data [28,34]. Host-based, network-based, and hybrid anomaly detection are the different categories of anomaly detection [13]. Most of the research in this area is focused on rule-based techniques, and that may be enough to identify botnets. Consequently, generation botnets are effectively able to encrypt their commands and use state-of-the-art cloaking techniques that do not follow predefined rules. Nonetheless, we believe that the sophisticated nature of botnets is justification for saying that anomaly detection methods are not appropriate for botnet detection [35].

- Host-based anomaly detection In early botnet detection, host-based detection might be effective to identify known malware activities. This strategy both monitors and analyses the internal processes of a computer system and all network traffic received by a host computer. It has the advantage that it can discover if an attack is successful, and it also records what the attacker has performed on the host computer. By examining system traces such as event logs and system calls, this function is effectively accomplished [13,36,37]. The limitations of this system are high error rates and the need for extensive monitoring of all system activities, which consumes host system resources.
- Network-based anomaly detection This approach corresponds to the problem of discovering irregular patterns in network traffic that are not expected behavior. It analyzes and collects network traces, including flow statistics and network packets. Also, it allows a wide range of analysis, including botnet clustering, traffic classification, and network-wide anomaly detection while maintaining the performance of the monitored systems [13,38–40]. However, they focus primarily on offline learning, which is not suitable for botnet detection, and they require frequent retraining with new data. Furthermore, [41] has been focused on application detection as well as identifying significant techniques and issues in IP traffic analysis. Whereas [42] presented a review paper on network anomaly detection methods and [43] presented a review of intrusion detection system based on NetFlow. It also explained the principles of flow and classified attacks, and it reviewed detection strategies for botnets, scans, DDoS attacks, and worms in detail.

- Hybrid botnet detection Some authors have driven the development of botnet detection techniques further in conjunction with improving accuracy and precision. Some have used a hybrid intrusion detection system that incorporates the principles of both the network intrusion detection system(NIDS) and the host intrusion detection system (HIDS). Hybrid systems can detect new botnets in the early phase by collecting data and information for analysis from both the host side and the network side. This technique is more effective and efficient since it overcomes the limitations of both HIDS and NIDS [44].

After all, several studies [45–47] recently have been based on machine learning detection techniques, and they have given us the most promising methods for botnet detection, with a greater level accuracy.

3.2. RQ2: What Are the Latest Studies of Botnet Detection Based on Machine Learning?

Machine learning (ML) is a branch of artificial intelligence (AI) that aims to improve systems by learning from past experiences and to predict the future through data analysis. Data from past experiences is provided as input to the ML algorithm, which extracts patterns and build a model to represent the data. This model describes the existing patterns in the data so when it is given new unknown data, it should be able to make well-informed decisions. Based on the learning approaches and training data, ML methods are typically classified as supervised learning, unsupervised learning, and semi-supervised learning [1,15,23,48].

Supervised learning (SL) is a method in which training data are labeled. To construct the classifier, the computer "learns" from the labeled patterns and use them to predict labels for new data [15,24]. In unsupervised learning (UL), the training data have not been labeled. In this approach, the computer "learns" by analyzing data features to create the classifier. Among the many available ML algorithms Decision Tree (DT), BayesNet (BN), J48, Naive Bayes (NB) and Support Vector machine (SVM) are the most prominently used [15,24]. Semi-supervised learning is a mixture of SL and UL. In this approach, the input training dataset typically consists of both labeled and unlabeled data—usually a small amount of labeled data and a large amount of unlabeled data. Each approach has its benefits and drawbacks, as well as its own application domain [15,49].

Among these algorithms, the accuracy of botnet malware detection ranges from 95% to 99.9%. That means that no one algorithm can guarantee 100% detection in all the available architectures and diverse situations. Therefore, if only one algorithm is used for all situations, the results may not be as reliable as the predictions based on the dataset that was used to train the model. Hence, the best performing algorithms have been identified and combined to get better results under varied circumstances. Using this approach, the false-positive and false-negative rates are reduced significantly [23]. Figure 8 displays classification of Machine learning algorithms.

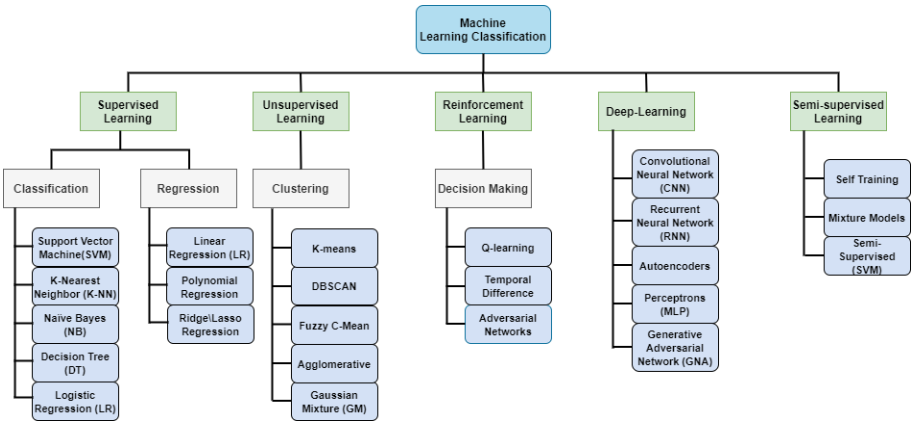

Figure 8. The classification of Machine Learning.

ML algorithm was first used for botnet detection in 2006 by [50]. That approach detects IRC chat traffic in flow-based botnet detection techniques by using ML algorithms to determine botnet and benign IRC traffic. That approach has two phases. In the first phase, several per-flow traffic features are extracted, including flow duration, maximum average byte count per packet, and initial congestion window. In the second phase, the algorithm is trained to detect bots by employing a Bayesian network classifier. Nonetheless, it yielded a very high false-positive rate: 15.04%. That means this system failed to capture botnet-specific network profiles effectively.

The BotMiner prototype was implemented in 2008 [38] to conduct network traffic clustering and correlations to distinguish bot hosts. That identification system groups similar communication and malicious traffic into clusters, then it performs cross-cluster correlation to identify hosts that have malicious activity patterns. It analyzed several real-world network traces that contained normal traffic and various real-world botnet traces. Those traces consisted of IRC, HTTP, and P2P-based botnet traffic. The results showed that BotMiner achieves an extremely high detection rate of 99.6% in conjunction with limited false positives in real-world network traces.

A P2P botnet detection system based on traffic analysis was proposed in 2011 [39]. Both host-based and flow-based features are considered in this system. The authors considered five supervised MLAs, namely: support vector machine (SVM), Gaussian-based classifier (GB), naive Bayesian (NB), nearest neighbors (NN) and artificial neural networks (ANN). The results compared the different algorithms according to the following matrix: detection performance, classification speed, and training. The results confirmed that the system achieved an accuracy level of 97% for SVM.

Disclosure, a large-scale wide-area botnet detection method, was introduced in 2012 [48]. This system employs C&C botnet detection models that use various supervised ML algorithms such as random forest and J48 decision tree classifier, as well as support vector machine algorithms. Furthermore, NetFlow data features are used and easily available. Only aggregate metadata regarding the number of packets transferred and flow duration are included, not full packet payloads. That research identified several groups of features that are extracted from the data, such as client-access patterns, flow size, and temporal behavior.The evaluation also showed that disclosure could detect botnet C&C channels in real time through datasets with billions of flows per day. It was also put to the test on two large real-world networks, with a True Positive identification rate of 65% and a False Positive rate of only 1%.

The effectiveness of flow-based features were discussed in 2014 [34], in combination with ML techniques for botnet detection. This study proposes the C4.5 decision tree supervised learning algorithm for detecting botnets and utilizes a publicly available real-world dataset and introduces a lot of diversity, through containing more than 40% unknown botnet samples in the test data. The features extracted are divided into four clusters and for feature selection a greedy algorithm is used. The research achieved lowest false positive of only 2.3% as well as moderate detection rate with 75%.

Furthermore, in 2015, a new approach was published in the paper Traffic-based IDS (T-IDS) [51]. It built a randomized data partitioned learning model (RDPLM). The features set, feature selection method, multiple randomized meta learning techniques and simplified sub spacing are all used in this model. The well-known Information Security and Object Technology (ISOT) dataset contains both malicious and normal traffic, mostly the traffic from two P2P botnets, namely Storm and Waledac from 2007 to 2009. The outcome showed that the proposed technique presented a significantly high detection precision rate (99.984%), and training time on the botnet dataset took about 21.38 s, a well-known fact.

The work done in 2015 [52], proposed three traffic classification methods that target TCP, UDP, and DNS traffic Random Forests classifier. Their experiment results indicate the possibility of obtaining high accuracy higher than 98% of botnet traffic classification for all three classification methods.

In 2017 [53], under a high-speed network environment, proposed a detection method for botnets using a supervised ML approach. PF-RING was used to build a detection framework for to solve the problem of high packet drop rates and also to extract the required fields from the dynamic traffic data. The author applied the random forest algorithm on the CTU dataset. High accuracy levels were acquired, including lower false-positive rates. However, the disadvantage of this study is that the researchers used only offline data; there was no online collection of other data.

Previous research in [54] suggested a method that utilizes supervised learning techniques to recognize P2P botnets. This research has been separated into two different phases. The first phase involves feature extraction from the traffic flows to characterize P2P botnets. The supervised learning algorithm was used to classify each flow and passed in the second phase of this mentioned research.

In 2018 [25], based on an artificial neural network classifier, a novel botnet-specific detection methodology was proposed. The SDN-specific dataset utilized and achieved a very high (up to 96%) traffic classification accuracy. That dataset was derived from the HogZilla dataset, which is a large collection of more than 990,000 samples based on both malicious and normal traffic each associated with 192 behavioural features.

Ref. [55] (2018) presented ML-based botnet detection, specifically using a botnet identification model called BotCap. Two main issues were discussed in that study: in-depth analysis of deep packet inspection (DPI) and collecting traffic information from different infected hosts in a network. Six botnet families were collected in the dataset for detailed analysis of several benign and botnet samples. This study employed SVM and J48 decision tree algorithms. These algorithms were trained to differentiate between benign and botnet network traffic. The results showed that the proposed method could identify infected hosts on a local network without requiring the collection of specific information from the infected machines. Moreover, the method obtained an accuracy level of 95% for IRC botnets and 80% for HTTP.

Ref. [56] (2018) proposed a deep learning-based botnet traffic analyzer named Botnet Traffic Shark (BoTShark). It adopted two deep learning methods, stacked autoencoders (SA) and convolutional neural networks (CNN) to detect malicious botnet traffic. The Installation Support Center of Expertise (US Army Corps of Engineers) (ISCX), a well-known research dataset in the field of botnet detection, was used to test BoTShark. The results indicated that SA achieved better results than CNN, with an accuracy of 91%. Because of minor false positives, it produced about 0.13% in detecting malicious traffic of botnets. The disadvantages of this study are that deep learning needs a huge amount of data for progressive learning and the system required high computational power.

To detect botnet behaviors in the network traffic [57] (2018), performed a study of different ML algorithms such as Naive Bayes, decision tree and neural network, as well as their ensembles. The results showed that the ensembling method was better able to detect botnet attacks in network traffic, compared to individual classifiers. The researchers used the CTU-13 botnet dataset for evaluation purposes. Most of the methods performed well, achieving an F1 score over 99%.

In 2019, Ref. [58] proposed a novel method to classify network traffic and also detect P2P botnets by way of ML algorithms. This two-stage traffic classification method used both ML techniques and the non-P2P traffic filtering mechanism on conversation features. During the first stage, the amount of network traffic was reduced by filtering non-P2P packages. Then, the extraction of conversation features, depending on flow similarity and data flow features, constituted the second stage. Experimental evaluations indicated that the proposed two-stage detection technique had a higher precision level than traditional P2P botnet detection approaches. The identification of P2P botnet traffic using a decision tree algorithm was found to have a high accuracy of 94.4%.

A multi-layer hybrid P2P botnet detection approach was suggested by [59] (2019). It applied ML classifiers to network traffic features. In this research, a framework structure focused on four theoretical layers. The first layer, non-P2P traffic, was filtered by a packet

to reduce processing overhead. The second layer, P2P and non-P2P traffic were identified by combining DNS queries, port filtering, and a fast heuristic P2P detectional approach. Then, in the third layer, feature reduction helped reduce the overfitting, which improved the precision rate of the classification model. In the final layer, a binary classifier was successful in classifying P2P traffic as normal or botnet. The experimental validation of this study showed that the decision tree algorithm achieved an average accuracy level of 98.7% when it was applied to the CTU and ISOT datasets. This was better than the results from other proposed models such as k-nearest neighbors, logistic regression, and artificial neural networks.

An experimental analysis in [60] (2019) was used to detect botnet DDoS attacks based on ML methods. It used DT, ANN, NB, SVM, and unsupervised machine learning (USML) on different datasets such as KDD99 and UNBS-NB 15. The results showed that the KDD99 dataset performed better than the UNBS-NB 15 dataset, with the maximum level of accuracy of 98.08%. In addition, regarding the performance metrics for the present study, the results suggested that USML was the best classifier for botnet detection of those used.

Ref. [61] (2019) proposed a novel deep learning framework to detect malicious domains generated by malicious Domain Generation Algorithms (DGA). in this study, Compared several deep-learning model architectures by training and testing individual models on the same subsets of data. A convolutional neural network was used, with a long short-term memory (CNN-LSTM) pipeline, was the model they proposed. They also evaluated the performance of the proposed method by using both publicly available and private datasets. The publicly available datasets included DGArchive and OSINT real-time DGA feeds, and the private datasets were from their internal network. The proposed framework results indicated that they could achieve higher accuracy, about 97.8%, and lower false positive rates.

Another study proposed a different deep learning-based botnet detection system that seems to be useful in recognizing P2P botnets [62] (2020). To automatically detect and learn policies for botnet, the research suggests an end-to-end data-driven method, using graph neural networks (GNNs). all traces collected were taken into consideration from the IP backbone CAIDA's monitors for background traffic. A high level of accuracy, approximately 99.5% was achieved in many different botnet scenarios.

In [63] (2020), supervised machine learning classification was used to classify P2P botnet traffic by using a flow-based approach. In total, the dataset consisted of 144,374 botnet traffic flows and 166,319 benign traffic flows, combined with the PeerRush dataset and botnet (2014) dataset (which is the combination of ISOT, ISCX 2012, and botnet traffic generation). This botnet detection approach uses three modules; feature extraction, feature selection or reduction, and at the last, classification. The results presented a comparison of the classifiers used for this research, and they specified the results achieved. A accuracy of 99.94% for detecting botnets was achieved using a J48 decision tree, along with a much lower false positive rate. The result for naïve Bayes was 99.46%. However the main disadvantages of the [64] study was the complexity of the model and a longer processing run-time.

A method based on deep learning was proposed by [65] (2020) to detect possible botnets by inspecting the behaviors of network traffic from network packets. The approach collects the packets for a period of time, and then extracts behavioral features from a range of packets. Then it analyzes these features with deep learning models. The principal contribution of that paper was a model that combined LSTM layers and RNN layers (RNN-LSTM) with two traditional models (LSTM and RNN). This model was trained using datasets from the malware capture facility project (MCFP), which was conducted by researchers at the Czech Technical University. However, due to the large size of the dataset, they selected four kinds of botnet: Dridex, Emotet, Sality, and Zbot. The results from the combination of LSTM and RNN proved to be very effective in botnet classification, with a higher level of accuracy of 99.36%. Moreover, deep learning methods were also shown to be effective in detecting DGA domains.

Ref. [66] (2020) proposed a sequential detection attack architecture that uses three ML models for Internet of Things (IoT) environments. The models it used were ANN, J48 decision tree, and NB. For feature reduction, the researchers used a correlated feature selection approach and made the system lightweight. They used the Network-based of IoT Botnet Attacks (N-BaIoT) dataset and achieved 99% accuracy. Furthermore in 2021 [67], employed RF for botnet attack classification in an IoT smart factory

A hybrid method for botnet detection called HANABot was introduced by [68] (2020). This method could detect new botnets in the early phase. The study used machine-learning decision mechanism such as the NB classifier and the DT classifier. The processed data were extracted from the sets of host data and from network traffic. The botnet communication traffic included HTTP, P2P, IRC, and DNS, by using IP fluxing. The researchers used two P2P-based botnets called Citadel and Zeus as the evaluation dataset, as well as the IRC-based botnets Rbot and Neris. They also used mixed botnet samples from the ISCX botnet dataset. The HANABot technique achieved an accuracy level of 99.6%, using 11 features.

Although many authors have studied and researched these features, the issue still needs to be explored, as these works leave room for improvement. Data packets are accessed in a distributed process, and every node of botnets must independently participate in the process of detection. This incurs high computation and communication costs. Thus, machine-learning techniques demand intensive resources to capture information of both outgoing and incoming traffic. For example, there might be a centralized traffic monitoring facility at the network gateway that starts the detection system once traffic features have been extracted. Some techniques might work for a network that has only low to medium traffic volume but be unsuccessful or ineffective in detecting larger networks such as cloud-based bots [66]. For this distributed design, it is impossible for to describe the entire network exactly, as well as the behavior of other nodes. Therefore, the author suggests that the efficiency and reliability of these detection methods should be improved. The fact that this network should house a centralized controller like an SDN simplifies this process and makes this a trouble-free task. Table 3 display the Summary of botnet detection based on ML.

Table 3. Summary of botnet detection based on ML.

Ref/Year	Contribution	Protocol	Methods	Dataset	Result	Comments
[50]/2006	First paper introduce botnet detection based on ML.	IRC	Bayesian networks	Dartmouth's wireless campus network traffic traces	FPR: 15.04%	The system failed to capture botnet-specific network profiles effectively
[38]/2008	Implement BotMiner prototype system to identify bot hosts.	IRC, HTTP, P2P	clustering algorithms	Real network traces	Accuracy: 99.6%	Achieves an extremely high detection rate with limited FPR in a real-world network trace
[39] 2011	System considers both flow-based features and host-based features.	P2P	Gaussian-based, KNN, SVM, NB, NN	Ericsson Research in Hungary	Accuracy: 97%	Their dataset is public but there is only one infected machine for each type of botnet, therefore no synchronization analysis can be done
[48]/2012	Proposed Disclosure, method for detecting botnets on a large scale.	IRC, HTTP, P2P	RF, J48 DT, SVM	university network dataset.	TPR: 65% FPR: 1%	There are no details of how many trees used on average in RF classifier

Table 3. *Cont.*

Ref/Year	Contribution	Protocol	Methods	Dataset	Result	Comments
[34]/2014	Indicate botnet detection based on flow based and ML.	IRC, HTTP, P2P	C4.5 DT	botnet (2014) (ISOT, ISCX 2012, and botnet traffic generating).	Accuracy: 75% FPR: 2.3%	Focuses on finding a mixture of packet-based, time-based, and behavior-based features that can be used to extract bot traffic
[51]/2016	Built an RDPLM that based on feature selection.	P2P	C4.5, DT, RT,	ISOT	Accuracy: 99.984% Training time: 21.38 s	The computational complexity and power increase at an exponential rate with this large dataset.
[53]/2017	Aimed to detect botnet under high-speed network environment.	IRC, HTTP, P2P	RF	CTU	Accuracy: 93.6% FAR: 0.3%	The researchers used only offline data; there was no online collection of other data.
[56]/2018	proposed botnet traffic analyzer based on a deep learning approach called BoTShark.	IRC, P2P	CNN, SA	ISCX	TPR: 0.91 FPR: 0.13	DL requires a huge amount of data and large amounts of processing power.
[55]/2018	botnet detection based on statistics features of network traffic using ML.	IRC, HTTP	J48, SVM	Real Botware sample collect it in their laboratory	Accuracy: HTTP (80%), IRC (95%). FPR: HTTP (0.05%), IRC (0.025%)	Able to identify infected hosts without requiring to collect information about them.
[57]/2018	ML ensembles flow-based for botnet detection	IRC, HTTP, P2P	GNB, NN, DT	CTU-13	F1 score over 0.99	DL needs a huge amount of data for progressive learning, as well as high computational power.
[59]/2019	Multi-layer approach to classify P2P traffic as normal or botnet based on ML classifier on network features.	P2P	DT, KNN, LR and ANN	CTU, ISOT	Accuracy: 98.7%	DT achieved better results than other proposed models KNN, LR and ANN
[61]/2019	Proposed DBD, a scalable deep learning DGA-based botnet detection system.	IRC, HTTP	CNN-LSTM	DGArchive, OSINT, internal network	Accuracy: 97.80%	Only DDoS attacks, one type of botnet attack, were included in this paper.
[58]/2019	Novel approach based in two-stage traffic classification.	P2P	DT, NB and ANN	CTU-13	Accuracy: 94.4%	NB was the lowest accuracy in this study, and they Prove that is used a two-stage technique is effective to detect P2P botnet traffic
[60]/2020	Analysis of the machine learning methods for botnet DDoS attack detection.	IRC, HTTP	DT, ANN, SVM, NB, USML	KDD99, UNBS-NB 15 dataset	Accuracy: 98.08%	Only DDoS attacks, one type of botnet attack, were included in this paper.
[66]/2020	botnet detection framework with sequential detection architecture based on ML in IoT network.	IRC, HTTP, P2P	ANN, J48 DT, NB	N-BaIoT dataset	Accuracy: 99%	Able to detect the known attacks and unknown attacks and their variances.
[62]/2020	P2P botnet detection based on Deep Learning.	P2P	GNN	CAIDA	Accuracy: 99.5%	The method is restricted to detecting attack nodes, especially in the sense of botnets, and does not detect individual attack flows.

Table 3. *Cont.*

Ref/Year	Contribution	Protocol	Methods	Dataset	Result	Comments
[63]/2020	Indicate P2P botnet detection based on ML using three modules feature extraction, feature selection or reduction, and classification	P2P	J48 DT	PeerRush,botnet (2014) (ISOT, ISCX 2012, and botnet traffic generating).	Accuracy: 99.94%	The main restriction is the complexity of the model and a longer processing run-time
[65]/2020	Aimed to detect botnets by inspecting the behaviors of network traffic from network packets using DL.	IRC, HTTP	LSTM-RNN, MCFP	real dataset collect it.	Accuracy: 99.36%	The proposed method able to detect different kinds of major botnets, as well as to adapt to the situation when those botnets alter how they interact or attack
[68]/2020	Hybrid botnet detection correlation between network traffic analysis and host traffic analysis.	IRC, HTTP, P2P	NB, DT	ISCX, CTU-13	Accuracy: 99.6%	The proposed technique implements the classification algorithm in an offline mode

3.3. RQ3: What Are the Latest Studies of Botnet Detection in Software-Defined Networks (SDNs)?

Software-defined networking (SDN) is an emerging technology which has been recently adopted to simplify the configuration of networks and network management. It has also gained incredible momentum from both academia and industry [69]. SDN refers to the principle of separating the data plane, (network devices) from the control plane (network intelligence). The control plane is responsible for making decisions on how network packets should be forwarded. It typically consists of one SDN controller. Network devices exist in the data plane. To communicate with the controller in the control plane, these network devices use specific protocols, such as OpenFlow. Furthermore, they handle only the packets in the flow tables managed by the controller; they do not handle packets on their own. In a conventional network, network administrators must access each network device. In contrast, SDN devices can be managed automatically through programming modules set up in the SDN controller. This allows developers to implement network functions and integrate them into the SDN controller simply by programming their modules to control packet flows in the data plane [25,69,70].

The conceptual architecture of SDN, generally consisting of three network layers is depicted in Figure 9. The layers are the infrastructure layer, the control layer, and the application layer [69,71]. The infrastructure layer, also known as the data plane, consists of numerous network devices with network functionality. This layer allows for managing and forwarding IP packets based on SDN controller decisions, that is, the forwarding rules for actions initiated by the applications running on the SDN controller, and in addition, installed on network devices by the so-called southbound interface [69,72–74]. Next is the the control plane, where the centralized and logical SDN controller is located. This is the most important and intelligent layer of any SDN architecture, and it acts as a network operating system (NOS). It hides underlying hardware and software infrastructure complications. It also provides a centralized view of the entire network, up to the application layer [69,73–75]. The management plane, also known as the Application layer, is situated at the top of the SDN architecture. It is collection of network applications that manage the control logic of a software-defined network. This is, where security services such as the firewall and network policies, such as quality of services (QoS), are specified. The communication protocol between the control layer and the management layer is referred to as the northbound interface. This is principally a set of open source application programming interfaces (APIs) [69,73].

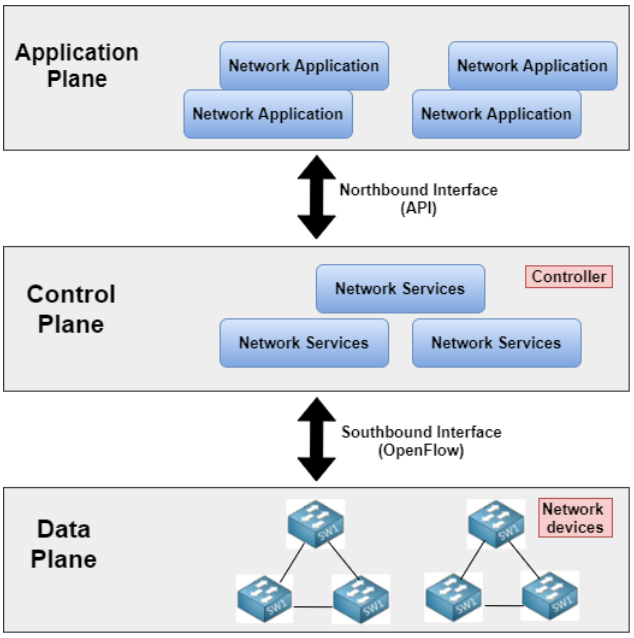

Figure 9. The architecture of software defined networks.

An SDN is more protected against certain threats, and it provides network management tools that are both extensible and scalable. However, it introduces vulnerabilities that do not exist in conventional networks. Seven main threat vector were defined by the authors in [76], (1) threats on control plane communication, (2) forged or fake traffic flows, trust between the controller and the applications, (4) trusted resources for forensics and remediation, (5) threats on vulnerabilities in switches, (6) threats on vulnerabilities in controllers, and (7) vulnerabilities in admin station. While the integration of the entire network management and configuration in a centralized SDN controller is considered a single point of failure.

Up until now, only a few related studies have been done regarding this issue. The research done in 2019 [77] simulated and discussed that using blockchain technology and SDN to detect botnets and prevent attacks.That study proposed a distributed botnet detection scheme for IoT that uses blockchain and SDN to detect botnets. The detection is done automatically, and the administrator does not need to be involved manually. The work using blockchain has the advantage of keeping IoT devices out of botnet bondage.

In 2013, FRESCO [78] performed one of the earliest attempts to provide SDN network systems security services. That framework allows the rapid development of security applications using OpenFlow on the control plane. the application layer uses APIs and interpreters to support modular applications, and the security enforcement kernel (SEK) performs all policy-related actions. Those two parts are built into the NOX OpenFlow controller. The concept behind FRESCO is to allow security-specific modules to be easily designed so they can be combined as an OpenFlow application.

However, there are quite a few methods for botnet detection in SDNs. While botnet detection techniques in SDNs were discussed by [77,78], they can not detect in time and achieve better performance with the advantage of SDN controller's global view. Despite all the features mentioned in the research above, there was no solution that used the computational power of the SDN data plane to perform more fine-grained and accurate detection.

3.4. RQ4: What Are the Latest Studies of Botnet Detection Based on Machine Learning (ML) in SDNs?

Recently, researchers have suggested combining ML and SDN to construct a scalable and accurate run-time bot-detection framework. The framework uses centralized learning with distributed detection to achieve detection scalability. To achieve high accuracy, the system not only conducts ML-based bot activity detection but it also performs parallel

detection of C&C channels. Studies have shown that, in the past five years, integrating ML with SDN plays an important role in the identification of botnets, which we discuss below. Table 4 displays Summary of botnet detection based on ML in SDN.

During 2015, [79] investigated the detection of botnets (IRC, HTTP, and P2P botnets) based on SDN controller. Decision Tree classifier is found to be effective for detecting peer-to-peer botnets, while SVM and BNs were more successful in detecting command and control (C&C) botnets like HTTP and IRC botnets.

Several authors in 2016 [46] have documented botnet detection in SDNs by gathering centralized network flow statistics in the procedure of OpenFlow counters. On collected counters, this study applies C4.5, a decision tree that depends on a supervised machine learning classification algorithm. Also, it employs a publicly available real-world botnet dataset. During the testing process, they tested unknown botnet samples, and the detection model did not train and achieve 50% diversity. The experimental results show an 80% detection rate.

Table 4. Summary of botnet detection based on ML in SDN.

Ref/Year	Contribution	Controller	Layer	Classifier	Dataset	Results	Comments
[79]/2015	Using IPFIX template for botnet detection in SDN.	POX Controller	Controller plane	DT, SVM, Bayesian networks	Private dataset	DT high Accuracy (ACC) detect P2P	The approach does not share evaluation results.
[46]/2016	Propose centralized collection of flow statistics for botnet detection using OpenFlow counters.	Opendaylight Controller.	Application plane	C4.5, DT	CTU-13, ISOT	ACC: 80%	This study aims to detect the attacks on the data plane to reduce the controller overhead
[80]/2016	first paper that leverages ML approach for defining security rules on the SDN controller.	Not mention.	Controller plane	C4.5 DT, Bayesian Network, DT, NB	LongTail	ACC: 91.68%	The results show that BayesNet performs better than the other three algorithms.
[81]/2017	flow-based approach to detect botnet by ML to SDN.	Opendaylight Controller	Application plane	C4.5 DT	CTU-13, ISOT	ACC: 97% known botnets, 90% unknown botnets.	the method of this study requires extensive computation
[82]/2018	Propose framework integrated SDN and ML to P2P botnet.	Ryu Controller	Controller plane	KNN, SVM	PeerRush	ACC: 99.88%	the approach in this paper using binary classifiers for each P2P traffic in SDN.
[83]/ 2018	Propose SDN framework to detect DDoS based on ML for the Campus network.	Ryu Controller	Controller plane	SVM	KDD99	ACC: 99.8%.	The dataset used is old and did not include recent DDoS attacks types
[84]/2018	Proposed a framework to mitigate botnet by SDN/NFV and ML.	Floodlight Controller	Controller plane	RF	CTU-13	ACC: 100% TPR: 92% FPR: 10%	This work needs to cover more protocols to defend against botnet attacks
[85]/2020	prposed hybrid CNN-LSTM model to detect slow DDoS attacks in SDN.	ONOS controller	Application Plane	CNN-LSTM	Synthetic Genera-tion of Traffic flow	>99%	The detection framework is performed by using offline datasets
[86]/2021	Proposed DDoS detection based on hybrid ML and the statistical method on SDN	Floodlight Controller	Controller plane	RT, LR, J48, BN and REPTree	UNB-ISCX, CTU-13, and ISOT datasets	In UNB-ISCX dataset: ACC: 99.85% FPR: 0.1%, In CTU-13 dataset: ACC: 99.12%	This method can be improved in networks by involving more than one controller

In 2016, Ref. [80] four widely known ML algorithms to predict potential vulnerable hosts and malicious code on historical data were used. These are DT, NB, C4.5 and Bayesian network (BN). The authors suggested to defy security rules on the SDN controller to secure potentially compromised hosts and restrict possible attackers' access by blocking the entire subnet. The results of this study found that the average estimate precision rate

of 91.68% was achieved by using BNs. The BNs outperformed all three of the previously mentioned algorithms.

In 2017, a study done by [81], suggested the utilization of the flow-based approach which would detect botnet in software-defined networks by further using ML algorithms without the need for reading packet payload. This paper's objective is to combine real-time flow trace data, paired with the historical context, thus allowing the author to extract an enriched feature set that further enables classification and to improve detection and thereafter, decreasing the false-positive rate. Furthermore, a C4.5 decision tree-based supervised ML algorithm approach is applied for the process of botnet classification. Moreover, botnet traces are derived from publicly available CTU-13 botnet datasets and ISOT botnet datasets. The preliminary results show that the system has ability of detecting unknown botnet, with results achieved of 97% for identify known botnets and 90% for unknown botnets, respectively.

In 2018, Ref. [82] presented an effective framework to detect P2P botnets. This framework is integrated SDN and ML to detect and categorize P2P network traffics. It has been divided P2P botnet detection into 5 main components as Traffic Flow and Feature Extractor Module, P2P Application Detection Module, Report to OpenFlow Controller, Flow Rule Modify on OpenFlow Switch and then Drop, Forward, or Redirect Packet. The experimental evidence showed that their framework proposed able to minimize the overhead of network administrators by automatically analyze network traffic and flexibly adjust flow entries in OpenFlow switches through the SDN controller. The data collection like Zeus and Storm botnet trace files used as botnet network traffic, and eMule, uTorrent, and Skype network trace files as benign network traffic for P2P applications. To assess the efficiency of classification accuracy and traffic management, they did an experiment of the system proposed in a testbed. The findings showed that with a high accuracy rate, the system cable detects all forms of P2P network traffic and automatically manage flow traffic entries through the SDN controller. For the classification module, the authors evaluated the output of k-NN, SVM, and RF. Where the experimental results showed that the SVM outperformed the KNN, the training period of the SVM is significantly longer than that of the KNN. As a result, the authors used KNN for the primary classification since SVM takes much longer to train than KNN. In addition, by analyzing flow intervals, this study demonstrated that the Decision Tree classifier is useful for detecting peer-to-peer (P2P) botnets. SVM and Bayesian networks, on the other hand, are good at detecting C&C communication in centralized botnets like HTTP and IRC botnets [46].

In 2018, Ref. [83] suggested an SDN framework for the Campus network ML -based to detect and defend DDoS attacks. Suggested framework consists of three modules, which are traffic collection module, DDoS attack identification module, and flow table delivery module. The tools used in this study are Ryu to build controller, The DDoS attack detection module was built using the SVM model, and Mininet is used to construct an experimental topology and simulate an SDN network. This experiment was conducted using the KDD99 dataset which has been widely used for testing and training network intrusion detection datasets. The results of the experiment demonstrate the efficiency of the DDoS attack detection system with a high accuracy rate of 99.8%.

in 2018, Ref. [54] proposed a framework that uses Traffic flow classification methods based on ML to reduce detection complexity and determines high-level SDN policies depend on the derived flow classes. Both supervised and unsupervised learning methods were used in this research. The supervised learning used pre-trained models for several types of traffic and C4.5 decision tree classifiers with features such as NetFlow Features includes flow tuple, packet count, packet size and inter-packet arrival time. For unsupervised learning on the same set of features, they clustered together different traffic flows by using the k-means algorithm. After collecting the traffic flow data by ML, they explored how it could be integrated into an SDN controller and provide an outline of the required software architecture and hardware. However, they did not implement new types of traffic. The results showed the normal traffic performs well, with an F-score of 80%. The F-scores

for each application profile decreased by 10% to 15% when attack traffic was introduced, but they remained elevated enough for the system to be effective.

In 2018, Ref. [84] proposed a framework for detecting botnet attacks and mitigating them by using SDN/NFV with a ML method. The network functions detect known attacks locally through various networking protocols while they continue to collect traffic feature set data from real-time traffic in the data plane. In that case, only feature-set information must be sent to the SDN controller for additional detection of distributed network attacks. The study had an accuracy level of 100%, with a 92% true positive rate (TPR) and only a 10% false-positive rate (FPR).

In 2020, Ref. [85] recommended the use of a hybrid CNN-LSTM model to detect slow DDoS attacks in SDN-based networks. Their experiment achieves more than 99% in all considered performance metrics.

in 2021, Ref. [86] presented DDoS attack detection based on hybrid machine learning algorithms and the statistical method implemented on SDN platform. This method contains of the collector, entropy-based, and classification section. The UNB-ISCX, CTU-13, and ISOT datasets were used in the experiments, and the results show that this system outperforms its counterparts in terms of accuracy in detecting DDoS attacks in SDN.

4. Challenges and Directions for Future Research

Today, the most significant threat on the Internet is the botnet. Despite collaborative efforts by law enforcement agencies and governments, as well as botnet detection approaches widely discussed and developed by numerous researchers, botnets continue to grab top news headlines all over the world. Numerous issues have been brought to light, but they have not been fully addressed, while new challenges keep emerging, stemming from new botnet attacks. In this section, we summarize some of the challenging research issues in botnet detection.

4.1. Quantity and Quality of Datasets Used

The existing methods particularly those based on machine learning techniques mostly rely on the quality and quantity of datasets used to train and test the algorithm. As a result, ensuring that there is enough amount, and the high quality of the data would help to increase the accuracy of the results for anomaly detection and reduce its influence on the proposed solution's performance. The popular botnet dataset like ISOT, ISCX, CTU-13, and KDD 99 are mentioned in this survey and summarize in Table 5.

Table 5. Summary of botnet detection based on ML in SDN.

Dataset	Year	Host Data	Network Data	Labelled	Classes	Protocols	Avg. Sample	Sample Units	Scenario	Comments
KDD99	1999	Yes	Yes	Yes	4	__	5.2 M	TCP packet	1	41 features of samples that represent both legitimate and attack traffic. It is old and did not include recent attacks types.
ISOT	2010	No	Yes	Yes	2	HTTP and P2P	1.7 M	Flows	1	It combines several existing malicious and non-malicious datasets. It includes lack normal data
Ctu-13	2011	No	Yes	Yes	7	IRC, HTTP and P2P	80 M	Flows	13	Well-known public benchmark includes both botnet and normal traffic. The first submitted in 2013, but dataset's website states 2011.
ISCX IDS	2012	No	Yes	No	4	IRC	2.5 M	Flows	7	It uses in several IDS studies

4.2. Fast-Flux Evasion Techniques

The botnet can hide its identity by using the Fast-Flux method and then engaged in illegal activities, which is a DNS technique that masks botnets by rapidly switching among a network of compromised hosts that act as proxies. This allows cybercriminals to

delay or evade detection [8]. Machine learning techniques were proposed to develop and incorporate a real-time malicious fast-flux domain detection solution in [87].

4.3. Deep Packet Inspection (DPI)

In DPI, the system is assumed to have access to the payload of every packet. When the payload is not encrypted, this method can be extremely effective. However, the majority of new malware uses evasion techniques like payload encryption, protocol encapsulation, and obfuscation. Moreover, inspecting all packets on a high-speed network is a costly task due to the speed of networks and the amount of packets exchanged across networks is increasing daily. The study in 2018 [55], suggested that network-flow information be used to eliminate the need for DPI, which has several advantages. First, there is no DPI processing overhead, which improves system performance. Second, no examining packet contents mean implies that no privacy has been violated. Finally, the system becomes more resistant to encrypted traffic.

4.4. Offline Mode Classification

Existing botnet detection executes the classification algorithm in offline mode, so that, before being processed, internet traces are saved in a PCAP file. For this reason, botnet detection should be enhanced to achieve real-time classification with instantaneous levels of high accuracy. This can be accomplished by reducing the export time of the Netflow trace collector and move the processing, (which includes the supply upon immediate arrival of the exported Netflow traces) to the classification technique. Botnet activities can then be distinguished, and the necessary action can be taken. There are variations in the speed of real-time classification processing. This process depends on a number of factors, such as the export time intervals and the amount of the exported NetFlow traces.

4.5. High Intensive Resources Required

In some networks, even contemporaneous botnet detection mechanisms, such as ML and SDN, fail to meet the requirements of accurate and expeditious detection. This is because they often demand intensive resources to support traffic monitoring and collection. Furthermore, they provide limited understanding of the different stages of bot activities, especially in the early stages. This is critical for detecting bots' malicious intent.

4.6. Host Acting as Normal

A botnet can make changes in the registry of the infected host without indicating the date of the modification. The botnet can erase some of the important information that forensic investigation is trying to find on the victim's machine. It is also likely that the traffic size between bots will be similar to that of legitimate users in a given period. Therefore, botnet detection techniques must be redeveloped and more complex methods discovered.

4.7. Overhead in Host-Based Bot Detection

Detection techniques like processing and storage may cause overhead in the hosts because they must keep running to inspect network traffic and collect data. To avoid this challenge, we must identify general behavioral patterns of bots in system call levels and extract system call profiles. Then we can analyze these profiles, along with network traffic profiles. We must also take advantage of ML techniques to ascertain the best threshold values. There is increasing concern that in all the studies mentioned above, data packets are accessed in a distributed way, and each botnet node must participate in the detection process independently. This is very likely to create high computational and communication costs.

4.8. Computational Power of ML

Previous studies detected both known and unknown attacks with low error rates by exploited ML-driven anomaly detection with traffic-based statistical features. To identify C&C traffic, techniques like these demand comparing all flows against each other, which

incurs a significant computational overhead. Furthermore, the earlier studies are inefficient and unreliable, as they can be easily evaded by tweaking flow characteristics and encryption [17].

4.9. Unreliable Extraction of NetFlow Features

Flow-based features such as source and destination IPs, protocols, and the number of packets sent or received, are the most used features in the field of bot detection. However, these characteristics do not fully capture the communication patterns that might expose additional aspects of malicious hosts. Moreover, flow-level models incur high computational overhead and they can be avoided by tweaking behavioral characteristics such as modifying packet structure. They also can be easily evaded by tweaking flow characteristics and encryption [17].

To overcome these limitations, Graph-based features exploring those approaches is the most promising avenue for future study, in which graphs are extracted from network flow host-to-host communication patterns [35,88–90]. Because graph-based features are derived from flow-level information that reflects the true behavior of hosts, graphs are an alternative that overcomes these limitations. The authors believe that incorporating graph-based features into ML generates robustness against complex patterns of communication and the onslaught of unknown attacks.

5. Conclusions

The number of network security incidents has increased significantly in recent years. Despite the fact that there have been numerous botnet detection studies, the development of further research on the SDN security system is extremely low, as can be clearly seen by the lack of current research. This specifically pertains to the aspects of defending networks and identifying the variations in the attacks with regard to SDN network security. Moreover, to the best of the author's knowledge, there seems to be a significant lack of thorough and methodological analysis of the state of the art regarding various approaches to botnet detection. More specifically, this pertains to botnet detection based on ML and botnet detection based on ML in SDN. Furthermore, it can be clearly seen that many issues that need more research regarding the SDN security system, explicitly in botnet attacks, taking this survey paper into consideration. This survey has reviewed various techniques for detecting botnets in traditional networks as well as SDN. To overcomes all limitations mentioned in this survey, we believe graph-based features based on ML for bot detection in SDN is the most promising avenue for future study. While, Graph-based features, derived from flow-level information, which reflect the true structure of communications, interactions, and behavior of hosts is an alternate that overcomes these constraints.

Author Contributions: Conceptualization, K.S., K.A. and A.A.; methodology, K.S.; software, K.S.; validation, K.S., K.A., A.A. and M.U.A.; formal analysis, K.S.; investigation, K.S., K.A. and A.A.; resources, K.S.; data curation, K.S.; writing—original draft preparation, K.S.; writing—review and editing, K.A., A.A. and M.U.A. ; visualization, K.S., K.A. and A.A.; supervision, K.A. and A.A.; project administration, K.A. and A.A.; funding acquisition, K.S., K.A., A.A. and M.U.A. All authors have read and agreed to the published version of the manuscript.

Funding: The Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia has funded this project, under grant no. (KEP-10-611-42).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The Deanship of Scientific Research (DSR) at King Abdulaziz University, Jeddah, Saudi Arabia has funded this project, under grant no. (KEP-10-611-42). Therefore, the authors gratefully acknowledge the DSR for technical and financial support.

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

References

1. Alothman, B. Similarity based instance transfer learning for botnet detection. *Int. J. Intell. Comput. Res. (IJICR)* **2018**, *9*, 880–889. [CrossRef]
2. Morse, A. *Investigation: WannaCry Cyber Attack and the NHS*; Report by the National Audit Office. Accessed; National Audit Office: London, UK, 2018; Volume 1.
3. Amini, P.; Araghizadeh, M.A.; Azmi, R. A survey on Botnet: Classification, detection and defense. In Proceedings of the 2015 International Electronics Symposium (IES), Surabaya, Indonesia, 29–30 September 2015; pp. 233–238.
4. Alieyan, K.; ALmomani, A.; Manasrah, A.; Kadhum, M.M. A survey of botnet detection based on DNS. *Neural Comput. Appl.* **2017**, *28*, 1541–1558. [CrossRef]
5. Li, X.; Wang, J.; Zhang, X. Botnet detection technology based on DNS. *Future Internet* **2017**, *9*, 55. [CrossRef]
6. Singh, M.; Singh, M.; Kaur, S. Issues and challenges in DNS based botnet detection: A survey. *Comput. Secur.* **2019**, *86*, 28–52. [CrossRef]
7. Gaonkar, S.; Dessai, N.F.; Costa, J.; Borkar, A.; Aswale, S.; Shetgaonkar, P. A survey on botnet detection techniques. In Proceedings of the2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 24–25 February 2020; pp. 1–6.
8. Vishwakarma, R.; Jain, A.K. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommun. Syst.* **2020**, *73*, 3–25. [CrossRef]
9. Hadianto, R.; Purboyo, T.W. A survey paper on botnet attacks and defenses in software defined networking. *Int. J. Appl. Eng. Res.* **2018**, *13*, 483–489.
10. Ali, I.; Ahmed, A.I.A.; Almogren, A.; Raza, M.A.; Shah, S.A.; Khan, A.; Gani, A. Systematic Literature Review on IoT-Based Botnet Attack. *IEEE Access* **2020**, *8*, 212220–212232. [CrossRef]
11. Silva, S.S.; Silva, R.M.; Pinto, R.C.; Salles, R.M. Botnets: A survey. *Comput. Netw.* **2013**, *57*, 378–403. [CrossRef]
12. Limarunothai, R.; Munlin, M.A. Trends and challenges of botnet architectures and detection techniques. *J. Inf. Sci. Technol.* **2015**, *5*, 51–57.
13. Ghafir, I.; Svoboda, J.; Prenosil, V. A survey on botnet command and control traffic detection. *Int. J. Adv. Comput. Netw. Secur.* **2015**, *5*, 7580.
14. Haddadi, F.; Morgan, J.; Gomes Filho, E.; Zincir-Heywood, A.N. Botnet behaviour analysis using ip flows: With http filters using classifiers. In Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014; pp. 7–12.
15. Miller, S.; Busby-Earle, C. The role of machine learning in botnet detection. In Proceedings of the 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), Barcelona, Spain, 5–7 December 2016; pp. 359–364.
16. Vania, J.; Meniya, A.; Jethva, H. A review on botnet and detection technique. *Int. J. Comput. Trends Technol.* **2013**, *4*, 23–29.
17. Panimalar, P.; Rameshkumar, K. A review on taxonomy of botnet detection. In Proceedings of the 2014 International Conference on Advances in Engineering and Technology (ICAET), Singapore, 29–30 March 2014; pp. 1–4.
18. Asha, S.; Harsha, T.; Soniya, B. Analysis on botnet detection techniques. In Proceedings of the 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS), Karnataka, India, 6–7 May 2016; pp. 1–4.
19. Alauthman, M. An Efficient Approach to Online Bot Detection Based on a Reinforcement Learning Technique. Ph.D. Thesis, Northumbria University, Newcastle upon Tyne, UK, 2016.
20. Cisco Umbrella. *Cisco Annual Internet Report (2018–2023) White Paper*. 2020. Available online: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed on 25 April 2021).
21. Zeidanloo, H.R.; Shooshtari, M.J.Z.; Amoli, P.V.; Safari, M.; Zamani, M. A taxonomy of botnet detection techniques. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; Volume 2, pp. 158–162.
22. Karim, A.; Salleh, R.B.; Shiraz, M.; Shah, S.A.A.; Awan, I.; Anuar, N.B. Botnet detection techniques: Review, future trends, and issues. *J. Zhejiang Univ. Sci. C* **2014**, *15*, 943–983. [CrossRef]
23. Samson, F.; Vaidehi, V. Hybrid botnet detection using ensemble approach. *J. Theor. Appl. Inf. Technol* **2017**, *95*, 1646–1654.
24. Hyslip, T.S.; Pittman, J.M. A survey of botnet detection techniques by command and control infrastructure. *J. Digit. Forensics Secur. Law* **2015**, *10*, 2. [CrossRef]
25. Krishnan, P.; Duttagupta, S.; Achuthan, K. VARMAN: Multi-plane security framework for software defined networks. *Comput. Commun.* **2019**, *148*, 215–239. [CrossRef]
26. Amit, D.; Prashant, G. Botnet Detection through DNS based approach. *Int. J. Appl. Innov. Eng. Manag. (IJAIEM)* **2013**, *2*, 497–501.
27. Dagon, D.; Zou, C.C.; Lee, W. Modeling Botnet Propagation Using Time Zones. In Proceedings of the Network and Distributed System Security Symposium, NDSS, San Diego, CA, USA, 23–26 February 2006; Volume 6, pp. 2–13.
28. Hassan, M.M.M. Network Intrusion Detection System Using Genetic Algorithm and Fuzzy Logic. Ph.D. Thesis, Gauhati University, Guwahati, India, 2013.
29. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 1–22. [CrossRef]
30. Ieracitano, C.; Adeel, A.; Morabito, F.C.; Hussain, A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing* **2020**, *387*, 51–62. [CrossRef]

31. Nguyen, M.T.; Kim, K. Genetic convolutional neural network for intrusion detection systems. *Future Gener. Comput. Syst.* **2020**, *113*, 418–427. [CrossRef]
32. Pawlicki, M.; Choraś, M.; Kozik, R. Defending network intrusion detection systems against adversarial evasion attacks. *Future Gener. Comput. Syst.* **2020**, *110*, 148–154. [CrossRef]
33. Zarpelão, B.B.; Miani, R.S.; Kawakani, C.T.; de Alvarenga, S.C. A survey of intrusion detection in Internet of Things. *J. Netw. Comput. Appl.* **2017**, *84*, 25–37. [CrossRef]
34. Beigi, E.B.; Jazi, H.H.; Stakhanova, N.; Ghorbani, A.A. Towards effective feature selection in machine learning-based botnet detection approaches. In Proceedings of the 2014 IEEE Conference on Communications and Network Security, San Francisco, CA, USA, 29–31 October 2014; pp. 247–255.
35. Chowdhury, S.; Khanzadeh, M.; Akula, R.; Zhang, F.; Zhang, S.; Medal, H.; Marufuzzaman, M.; Bian, L. Botnet detection using graph-based feature clustering. *J. Big Data* **2017**, *4*, 1–23. [CrossRef]
36. Creech, G.; Hu, J. A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns. *IEEE Trans. Comput.* **2013**, *63*, 807–819. [CrossRef]
37. Garfinkel, T.; Rosenblum, M. A virtual machine introspection based architecture for intrusion detection. In Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, CA, USA, 5–8 May 2003; Volume 3, pp. 191–206.
38. Gu, G.; Perdisci, R.; Zhang, J.; Lee, W. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection. In Proceedings of the 17th USENIX Security Symposium, San Jose, CA, USA, 28 July–1 August 2008.
39. Saad, S.; Traore, I.; Ghorbani, A.; Sayed, B.; Zhao, D.; Lu, W.; Felix, J.; Hakimian, P. Detecting P2P botnets through network behavior analysis and machine learning. In Proceedings of the 2011 Ninth Annual International Conference on privacy, Security and Trust, Montreal, QC, Canada, 19–21 July 2011; pp. 174–180.
40. Zhao, D.; Traore, I.; Sayed, B.; Lu, W.; Saad, S.; Ghorbani, A.; Garant, D. Botnet detection based on traffic behavior analysis and flow intervals. *Comput. Secur.* **2013**, *39*, 2–16. [CrossRef]
41. Bhuyan, M.H.; Bhattacharyya, D.K.; Kalita, J.K. Network anomaly detection: Methods, systems and tools. *IEEE Commun. Surv. Tutor.* **2013**, *16*, 303–336. [CrossRef]
42. Zhang, W.; Yang, Q.; Geng, Y. A survey of anomaly detection methods in networks. In Proceedings of the 2009 International Symposium on Computer Network and Multimedia Technology, Wuhan, China, 18–20 December 2009; pp. 1–3.
43. Sperotto, A.; Schaffrath, G.; Sadre, R.; Morariu, C.; Pras, A.; Stiller, B. An overview of IP flow-based intrusion detection. *IEEE Commun. Surv. Tutor.* **2010**, *12*, 343–356. [CrossRef]
44. Bridges, R.A.; Glass-Vanderlan, T.R.; Iannacone, M.D.; Vincent, M.S.; Chen, Q. A survey of intrusion detection systems leveraging host data. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–35. [CrossRef]
45. Brezo, F.; de la Puerta, J.G.; Ugarte-Pedrero, X.; Santos, I.; Bringas, P.G. A supervised classification approach for detecting packets originated in a HTTP-based botnet. *CLEI Electron. J.* **2013**, *16*, 2. [CrossRef]
46. Tariq, F.; Baig, S. Botnet classification using centralized collection of network flow counters in software defined networks. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 1075.
47. Barthakur, P.; Dahal, M.; Ghose, M.K. A framework for P2P botnet detection using SVM. In Proceedings of the 2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Sanya, China, 10–12 October 2012; pp. 195–200.
48. Bilge, L.; Balzarotti, D.; Robertson, W.; Kirda, E.; Kruegel, C. Disclosure: Detecting botnet command and control servers through large-scale netflow analysis. In Proceedings of the 28th Annual Computer Security Applications Conference, Orlando, FL, USA, 3–7 December 2012; pp. 129–138.
49. Wang, X.; Xu, Y.; Chen, C.; Yang, X.; Chen, J.; Ruan, L.; Xu, Y.; Chen, R. Machine Learning Empowered Spectrum Sharing in Intelligent Unmanned Swarm Communication Systems: Challenges, Requirements and Solutions. *IEEE Access* **2020**, *8*, 89839–89849. [CrossRef]
50. Carl, L.; Walsh, R.; Lapsley, D.; Strayer, W.T. Using machine learning technliques to identify botnet traffic. In Proceedings of the 2006 31st IEEE Conference on Local Computer Networks, Tampa, FL, USA, 14–16 November 2006.
51. Al-Jarrah, O.Y.; Alhussein, O.; Yoo, P.D.; Muhaidat, S.; Taha, K.; Kim, K. Data randomization and cluster-based partitioning for botnet intrusion detection. *IEEE Trans. Cybern.* **2015**, *46*, 1796–1806. [CrossRef]
52. Stevanovic, M.; Pedersen, J.M. An analysis of network traffic classification for botnet detection. In Proceedings of the 2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), London, UK, 8–9 June 2015; pp. 1–8.
53. Chen, R.; Niu, W.; Zhang, X.; Zhuo, Z.; Lv, F. An effective conversation-based botnet detection method. *Math. Probl. Eng.* **2017**, *2017*. [CrossRef]
54. Comaneci, D.; Dobre, C. Securing networks using SDN and machine learning. In Proceedings of the 2018 IEEE International Conference on Computational Science and Engineering (CSE), Bucharest, Romania, 29–31 October 2018; pp. 194–200.
55. Gadelrab, M.S.; ElSheikh, M.; Ghoneim, M.A.; Rashwan, M. BotCap: Machine learning approach for botnet detection based on statistical features. *Int. J. Commun. Netw. Inf. Secur* **2018**, *10*, 563.
56. Homayoun, S.; Ahmadzadeh, M.; Hashemi, S.; Dehghantanha, A.; Khayami, R. BoTShark: A deep learning approach for botnet traffic detection. In *Cyber Threat Intelligence*; Springer: Cham, Switzerland, 2018; pp. 137–153.
57. Ryu, S.; Yang, B. A comparative study of machine learning algorithms and their ensembles for botnet detection. *J. Comput. Commun.* **2018**, *6*, 119. [CrossRef]

58. Khan, R.U.; Kumar, R.; Alazab, M.; Zhang, X. A Hybrid Technique to Detect Botnets, Based on P2P Traffic Similarity. Technical Report; In Proceedings of the 2019 Cybersecurity and Cyberforensics Conference (CCC), Melbourne, VIC, Australia, 8–9 May 2019.
59. Khan, R.U.; Zhang, X.; Kumar, R.; Sharif, A.; Golilarz, N.A.; Alazab, M. An adaptive multi-layer botnet detection technique using machine learning classifiers. *Appl. Sci.* **2019**, *9*, 2375. [CrossRef]
60. Tuan, T.A.; Long, H.V.; Son, L.H.; Kumar, R.; Priyadarshini, I.; Son, N.T.K. Performance evaluation of Botnet DDoS attack detection using machine learning. *Evol. Intell.* **2019**, *13*, 283–294. [CrossRef]
61. Vinayakumar, R.; Soman, K.; Poornachandran, P.; Alazab, M.; Jolfaei, A. DBD: Deep learning DGA-based botnet detection. In *Deep Learning Applications for Cyber Security*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 127–149.
62. Zhou, J.; Xu, Z.; Rush, A.M.; Yu, M. Automating Botnet Detection with Graph Neural Networks. *arXiv* **2020**, arXiv:2003.06344.
63. Gahelot, P.; Dayal, N. Flow based botnet traffic detection using machine learning. In *Proceedings of ICETIT 2019*; Springer: Cham, Switzerland, 2020; pp. 418–426.
64. Asadi, M.; Jamali, M.A.J.; Parsa, S.; Majidnezhad, V. Detecting botnet by using particle swarm optimization algorithm based on voting system. *Future Gener. Comput. Syst.* **2020**, *107*, 95–111. [CrossRef]
65. Shi, W.C.; Sun, H.M. DeepBot: A time-based botnet detection with deep learning. *Soft Comput.* **2020**, *24*, 16605–16616. [CrossRef]
66. Soe, Y.N.; Feng, Y.; Santosa, P.I.; Hartanto, R.; Sakurai, K. Machine Learning-Based IoT-Botnet Attack Detection with Sequential Architecture. *Sensors* **2020**, *20*, 4372. [CrossRef]
67. Lee, S.; Abdullah, A.; Jhanjhi, N.; Kok, S. Classification of botnet attacks in IoT smart factory using honeypot combined with machine learning. *PeerJ Comput. Sci.* **2021**, *7*, e350. [CrossRef] [PubMed]
68. Almutairi, S.; Mahfoudh, S.; Almutairi, S.; Alowibdi, J.S. Hybrid Botnet Detection Based on Host and Network Analysis. *J. Comput. Netw. Commun.* **2020**, *2020*. [CrossRef]
69. Alharbi, T. Deployment of blockchain technology in software defined networks: A survey. *IEEE Access* **2020**, *8*, 9146–9156. [CrossRef]
70. Elsayed, M.S.; Le-Khac, N.A.; Jurcut, A.D. InSDN: A Novel SDN Intrusion Dataset. *IEEE Access* **2020**, *8*, 165263–165284. [CrossRef]
71. Tank, G.P.; Dixit, A.; Vellanki, A.; Annapurna, D. *Software-Defined Networking: The New Norm for Networks*; Open Networking Foundation: Menlo Park, CA, USA, 2012.
72. McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; Turner, J. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 69–74. [CrossRef]
73. Blial, O.; Ben Mamoun, M.; Benaini, R. An overview on SDN architectures with multiple controllers. *J. Comput. Netw. Commun.* **2016**, *2016*. [CrossRef]
74. Adnan, M.; Iqbal, J.; Waheed, A.; Amin, N.U.; Zareei, M.; Goudarzi, S.; Umer, A. On the Design of Efficient Hierarchic Architecture for Software Defined Vehicular Networks. *Sensors* **2021**, *21*, 1400. [CrossRef]
75. Gude, N.; Koponen, T.; Pettit, J.; Pfaff, B.; Casado, M.; McKeown, N.; Shenker, S. NOX: Towards an operating system for networks. *ACM SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 105–110. [CrossRef]
76. Ahmed, U.; Raza, I.; Hussain, S.A.; Ali, A.; Iqbal, M.; Wang, X. Modelling cyber security for software-defined networks those grow strong when exposed to threats. *J. Reliab. Intell. Environ.* **2015**, *1*, 123–146. [CrossRef]
77. Shafi, Q.; Basit, A. DDoS Botnet prevention using blockchain in software defined Internet of Things. In Proceedings of the 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), Islamabad, Pakistan, 8–12 January 2019; pp. 624–628.
78. Shin, S.W.; Porras, P.; Yegneswara, V.; Fong, M.; Gu, G.; Tyson, M. Fresco: Modular composable security services for software-defined networks. In Proceedings of the 20th Annual Network & Distributed System Security Symposium, NDSS, San Diego, CA, USA, 24–27 February 2013.
79. Wijesinghe, U.; Tupakula, U.; Varadharajan, V. Botnet detection using software defined networking. In Proceedings of the 2015 22nd International Conference on Telecommunications (ICT), Sydney, Australia, 27–29 April 2015; pp. 219–224.
80. Nanda, S.; Zafari, F.; DeCusatis, C.; Wedaa, E.; Yang, B. Predicting network attack patterns in SDN using machine learning approach. In Proceedings of the 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Palo Alto, CA, USA, 7–10 November 2016; pp. 167–172.
81. Tariq, F.; Baig, S. Machine learning based botnet detection in software defined networks. *Int. J. Secur. Its Appl.* **2017**, *11*, 1–11. [CrossRef]
82. Su, S.C.; Chen, Y.R.; Tsai, S.C.; Lin, Y.B. Detecting p2p botnet in software defined networks. *Secur. Commun. Netw.* **2018**, *2018*. [CrossRef]
83. Yang, L.; Zhao, H. DDoS attack identification and defense using SDN based on machine learning method. In Proceedings of the 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), Yichang, China, 16–18 October 2018; pp. 174–178.
84. Park, Y.; Kengalahalli, N.V.; Chang, S.Y. Distributed security network functions against botnet attacks in software-defined networks. In Proceedings of the 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Verona, Italy, 27–29 November 2018; pp. 1–7.
85. Nugraha, B.; Murthy, R.N. Deep Learning-based Slow DDoS Attack Detection in SDN-based Networks. In Proceedings of the 2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), online, 9–12 November 2020; pp. 51–56.

86. Dehkordi, A.B.; Soltanaghaei, M.; Boroujeni, F.Z. The DDoS attacks detection through machine learning and statistical methods in SDN. *J. Supercomput.* **2021**, *77*, 2383–2415. [CrossRef]
87. Kumar, S.A.; Xu, B. A Machine Learning Based Approach to Detect Malicious Fast Flux Networks. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1676–1683.
88. Daya, A.A.; Salahuddin, M.A.; Limam, N.; Boutaba, R. BotChase: Graph-Based Bot Detection Using Machine Learning. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 15–29. [CrossRef]
89. Lagraa, S.; François, J.; Lahmadi, A.; Miner, M.; Hammerschmidt, C.; State, R. BotGM: Unsupervised graph mining to detect botnets in traffic flows. In Proceedings of the 2017 1st Cyber Security in Networking Conference (CSNet), Rio de Janeiro, Brazil, 18–20 October 2017; pp. 1–8.
90. Blaise, A.; Bouet, M.; Conan, V.; Secci, S. BotFP: Fingerprints clustering for bot detection. In Proceedings of the NOMS 2020—2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–7.